Advances in Science, Technology and Engineering Systems Journal
Vol. 5, No. 4, 700-709 (2020)
www.astesj.com
Special Issue on Multidisciplinary Sciences and Engineering

ASTES Journal
ISSN: 2415-6698

Deep Learning Approach for Automatic Topic Classification in an Online Submission System

Tran Thanh Dien, Nguyen Thanh-Hai, Nguyen Thai-Nghe[*]

College of Information and Communication Technology, Can Tho University, Can Tho city, 900000, Vietnam

ARTICLE INFO

Article history:
Received: 8 July, 2020
Accepted: 11 August, 2020
Online: 28 August, 2020

Keywords:
Article classification
Deep learning
Knowledge search
Online submission system

ABSTRACT

Topic classification is a crucial task where knowledge categories exist within hierarchical information systems designed to facilitate knowledge search and discovery. An application of topic classification is article (e.g., journal/conference paper) classification which is very useful for online submission systems. In fact, numerous online journals/magazine submission systems usually receive thousands of article submissions or even more for each month. This leads to a huge amount of time-consumption of editors to process and categorize the submissions aiming to look for and assign appropriate reviewers to the submitted articles. In this study, we propose an approach based on natural language processing techniques and machine learning algorithms (both classic machine learning and deep learning) to automatic classify the topics of articles in an online submission system. We show by promising performance collected from prediction tasks to present that the proposed method is a potential approach for applying to the real system.

1 Introduction

With the rapid development of data sources and computational algorithms, the classification tasks, especially in text classification, have revealed an important role in numerous fields [1]. Text classification is a supervised learning technique that has been deploying popularly in practical cases [2]. More specific, text classification tasks are classical text processing problems attempting to categorize a unseen text into a group of known texts based on its similarities to the considered group [3]. The authors in [4] stated that text classification tasks are the assignment or categorization of labels on a new text based on the similarities of the considered text to the labeled texts in the training set. The framework using text classification automatically allows information to be processed and searched easier. Furthermore, sorting each of them takes a lot of time and effort with a large number of texts, not to mention the possibility of inaccurate categorization due to the subjectivity of the people. From the proposed studies, there are numerous real applications of text classification tasks including news classification by topics in online newspapers, knowledge management, spam email filtering, and supportable tools for search engines on the Internet, etc. [1].

The text classification tasks are attracting numerous scientists with a vast of the studies proposed with different algorithms including machine learning as well as mathematical-statistical model. In recent years, machine learning has been widely implemented and investigated with numerous advancements. Many learning algorithms include k-nearest neighbors (kNN), Naïve Bayes, support vector machines (SVM), decision tree, and artificial neural network, etc. which are leveraged to solve text classification problem with the published studies in [5–12].

An important application of text classification is article/journal classification where not only the authors but also the editorial boards of the magazines/journals would like to know how to classify a document into a relevant topic of those magazine/journal to search or/and submit. More specifically, the submission system enables us to disseminate texts and extract relevant information automatically when a manuscript is submitted to the system. From the predicted category, editors can find appropriate reviewers for the submission faster and help to speed up the review process.

This paper is an extension of work originally presented in the International Conference on Advanced COMPuting and Applications (ACOMP) 2019 in Nha Trang, Vietnam [13]. We present an approach using machine learning algorithms to classify automatically the articles which were submitted via an online submission system. More specifically, when we submit an article (the extension can be doc(x) or pdf, etc.) to the online submission system, the system not only extract automatically the information on the author, title, and abstract but stratify and assign the topic to the submission. For this procedure, we can use the natural language processing techniques to pre-process the data before fetching them into a machine learning

[*] Corresponding Author: Nguyen Thai-Nghe, Can Tho city, Vietnam, Contact: +84918028402 & email: ntnghe@cit.ctu.edu.vn

https://dx.doi.org/10.25046/aj050483

algorithm to do the classification tasks. Comparing to the work in [13], this study provides contributions as follows:

- After pre-processing the data, we proposed to use Deep Learning (MLP) for the classification tasks. Experimental results show that by using this approach, the results even get better than using the SVM in our previous work.

- For experiments, we have collected 5 data sets (instead of 2 Vietnamese data sets as in [13]). These data sets represent for 3 languages (English, Turkey, and Vietnamese) and have multi-classes (2, 4, 6, 9, and 10 classes as presented in Table 1)

- In this work, since the topic classes are imbalanced, we have used AUC (Area Under the ROC Curve) as a metric for comparison. Previous works show that when the data sets are imbalanced, the AUC metric is a better measurement for evaluation [14, 15]. We will analyze this point in the result section.

- For faster experiments, the pre-processing data steps are combined to a single procedure as presented in Algorithm 1.

- We have reviewed more and up-to-date related works.

In the next sections of the paper, we present a literature review on robust machine learning algorithms used in text classification tasks in Section 2. In Section 3, we introduce the proposed framework including classification models and steps for pre-processing. Section 4 describes the empirical results and Section 5 provides insightful remarks of the study.

2 Related works

In the context of the enormous development of digital information in recent years, text mining techniques hold a crucial role in information and knowledge management and mining, attracting the attention of scientists [1]. Text classification through modern techniques is the division of a dataset including documents into two or more topics. The text classification purposes to assign a predefined label or category to a document. For example, a new article published on an online newswire system can be assigned to one of the given topics while each submission sent to an online journal system can be automatically stratified into its topics, etc.

2.1 Related works on text classification

Numerous papers have proposed methods and learning architectures on text classification to enhance performance and apply in practical cases. For example, authors in [16] employed the Chi-square feature selection (referred to, hereafter, as ImpCHI) to enhance the classification performance for Arabic documents classification. They also compared this improved chi-square with three traditional features selection metrics namely mutual information, information gain and Chi-square. Another study in [17] introduced a new firefly algorithm based feature selection method which achieved a precision value equals to 0.994 on an Open Source Arabic Corpora (OSAC) dataset.

Some techniques for data pre-processing phase have been introduced in numerous studies. Authors in [18] presented the maximum matching segmentation (MMSEG) algorithm to segment words. When the segmentation completed, the text was fetched into a vector form, with the vectorized TF*IDF. Next, data were classified using decision trees and Support Vector Machines with the Weka package. The experimental results of [18] were evaluated on the dataset with 7,842 texts categorized in 10 different topics. About 500 texts of each topic were selected randomly for training phase and the remaining were used to verify independence. As reported in the paper, the performance with Support Vector Machines algorithm was greater than the algorithm of decision tree. Besides, the authors deployed singular value decomposition to analyze and reduce the characteristic space dimension and hence, it can improve classification performance of Support Vector Machines algorithm.

Another work [19] studied the semantic relation extraction and classification in scientific paper abstracts. The authors presented the steps of setup procedures and experimental results of semantic relation extraction and classification on scientific papers datasets. The task included three sub-tasks: the first classification was performed on the clean data while the second one was on noisy data, and the final task combined extraction and classification scenario. Some datasets which were used in the challenges such as a subset of abstracts of published papers in the ACL Anthology Reference Corpus, annotated for domain specificentities and semantic relations were introduced in this study. Root Cause Analysis of Incidents using Text Clustering was proposed in [20]. The authors studied the use of two machine learning (ML) algorithms, namely random forest (RF), and support vector machine (SVM) and found that SVM performed best in classifying the accident narratives. Multinomial Naive Bayes (MNB), Logistic Regression (LR), Support Vector Machines were also deployed in [21] to prediction task on Twitter Data.

Approaches based on deep learning techniques are also carried out in numerous studies. The authors in [22] presented A Comprehensive Review on deep learning techniques for text classification. The authors in [23] proposed three fundamental architectures of deep learning models for the tasks of text classification: Deep Belief Neural (DBN), “Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). The work also introduced basic guidance about the deep learning models that which models is best for the task of text classification. The authors [24] presented a novel angle to further improve this representation learning, i.e., feature projection, and projected existing features into the orthogonal space of the common features. Another paper is [25] which illustrated a model of statistics like TF-IDF, to exploit pre-trained SOTA DL models (such as the Universal Sentence Encoder) without any need for traditional transfer learning or any other expensive training procedure on Text Classification tasks of UNGA Resolutions. The authors in [26] introduced a model with Gated recurrent unit (GRU) and support vector machine. The method implemented a linear support vector machine (SVM) as the replacement of Softmax in the final output layer of a GRU model to obtain comparative, remarkable results. The work in [27] introduced a new improved algorithm of the original Sine Cosine Algorithm for feature selection with the software written in Matlab2013 to achieve high performance on the datasets of Reuters-21578 collection [1]. Some studies [28, 29] have

[1] D.D., 2004. Reuters-21578. Retrieved November 6, 2013, http://www.daviddlewis.com/0Aresources/testcollections/reuters21578/

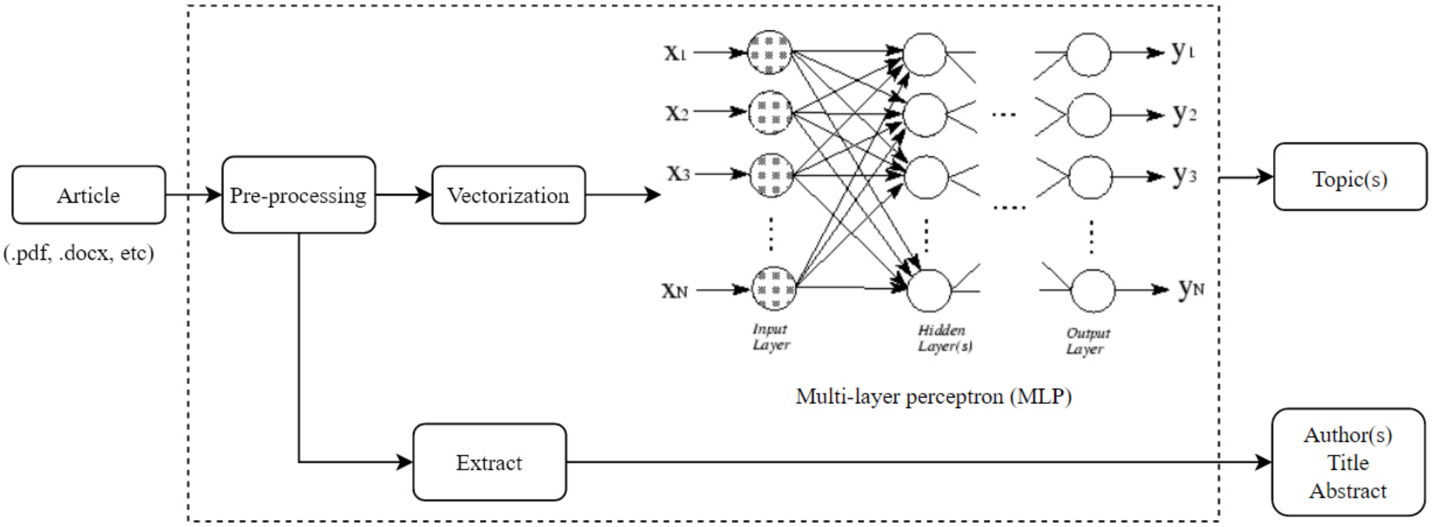

Figure 1: The proposed architecture for pre-processing and classification of articles

attempted to use Reinforcement Learning for text classification.

2.2 Text vectorization

In real word, some text representation models have been proposed including vector space model, bag of words model, and graph-based model. The vector space model [30] is deployed in this work because it can represent unformatted documents as simple and formulaic notation. Due to its advantages, many researches based on vector space model are implemented [31]. According to the details of this method, each document or article is represented as a vector while each component of such vector is a separated term that is assigned to a value namely "weight" of that term.

TF measures the frequency of a word in a document. It depends on the document length and the generality of word. For measuring the weight of a word, the number of occurrences of the word is divided by the length of the document (the number of words) as equation 1:

$$TF(t,d) = \frac{\text{number of occurrences of term } t \text{ in d}}{\text{Total number of terms in d}} \quad (1)$$

There is difference between TF and IDF. TF counts frequency of a term t in document d, where as DF counts occurrences of term t in the document set N. For calculating the TF, all the terms reveal the same importance. However, it is found that not all terms in a dataset are important, for instance, connecting terms, determiners; and prepositions. It is necessary to reduce the importance of such terms with computing IDF by the formula as Equation 2:

$$IDF(t,D) = log\frac{\text{Total document in D}}{\text{Number of document including } t} \quad (2)$$

We compute TF*IDF which integrate between TF and IDF. The method calculates the TF*IDF value of a term via its importance in a document belonging to a document set. Using such method, we are able to filter out common words and retain high value words as equation 3.

$$TF*IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

In this study, we propose an approach of automated classification of articles submitted via an online submission system. When the submission (with extension such as *.doc(x), *.pdf, etc.) is uploaded, the system, then, extracts the author's information and abstract to stratify the submission.

3 Proposed method

The proposed overall system including extracting information, pre-processing data methods and categorizing articles is exhibited in Fig. 1.

In proposed model, when a new article under formatting of .docx, .pdf, etc. is submitted to the system, its information consisting of abstract, author(s), title can be automatically extracted, and especially categorized into an appropriate topic based on the previous data trained by machine learning models. Due to article's pre-formatted template, article's information extraction is easy to get. Therefore, this work only focuses on how to classify article's topics when it is submitted to the system.

In the following sections, we will present how to pre-process the data and setting up the classification models.

3.1 Data pre-processing

Data pre-processing described as Algorithm 1. This algorithm receives a document as input, then the document is converted and normalized with techniques such as changing to lower cases, removing blanks, etc. The algorithm also separates documents into words and eliminate noises then transform to the vector. The details are in Algorithm 1.

File format conversion and word standardization: If the documents of datasets have the extension of .doc(x), they will be converted to plain text (.txt) for easy use in most of classification models. After converting file format, word standardization performs to convert all text characters into lowercase while spaces are also eliminated.

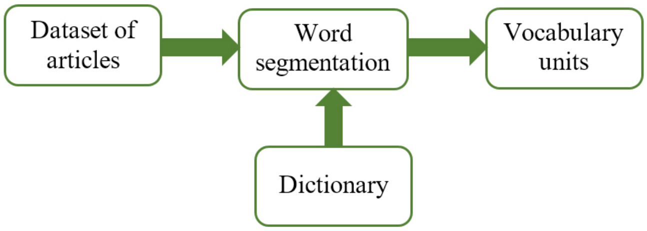

Figure 2: Process of word segmentation

Algorithm 1: Pre-processing for documents

```
**Data:** InputDocument $d$
**Result:** Vectorized-Documents D
$Convertion(d)$: convert the input document (word/pdf) to text
$WordNormalization(d)$: changed to lower cases, removing
  blanks
$WordSegmention(d)$: separate document to words
$RemovingStopWords(d)$: remove noise words
$Vectorization(d)$: convert documents to respectively vectors
**Return** sets of Vectorized-Documents D
```

Word segmentation: In some languages including Vietnamese, spaces do not segment words but separate syllables so the segmentation is an important phase in NLP. There are a lot of tools that have been successfully developed to segment words with relatively high accuracy. For instance, the *VnTokenizer* segmentation tool [2] was used for Vietnamese word segmentation. It was based on the integrated methods of maximum matching, weighted finite-state transducer and regular expression parsing, running on the dataset of Vietnamese syllabary and Vietnamese vocabulary dictionary. This tool segments Vietnamese documents into vocabulary units (words, names, numbers, dates and other regular expressions) with over 95% accuracy described in Fig. 2

Removing stop words: As mentioned in [32] that stop words are the words that widely appear in all documents of the considered dataset , or the words that appear only in one and several documents. Therefore, they do not contain useful information or make sense. For tasks of text classification, the appearance of such types of words not only do not help examine the classification but also cause noises leading to a decrease in performance of the classification process.

Text vectorization In this study, we use TF-IDF (term frequency-inverse document frequency) as a statistical measure that evaluates how relevant a term is to a document in a collection of documents.

Text classification algorithms From a set of documents $\{d_1 ... d_n\}$ called a training set, in which document d_i is labeled under c_j belonging to set of categories C = $\{c_1 ... c_m\}$, classification model is determined for classifying any document d_k into an appropriate category of set C.

In our experiments, some text classification algorithms which are implemented for comparison include SVM algorithms, tree decision and deep learning techniques.

3.2 Deep learning model

The proposed model is presented in Fig. 1 where the input attributes are selected from Table 1 and the output (prediction) of the model including classes depending the selected dataset. The proposed MultiLayer Perceptron (MLP) architecture includes one hidden layer with 16 neurons (see an illustration in Figure 3) which is conducted from fine-tune hyperparameters experiments (see Section 4.2) running n hidden layer(s) with various (m) numbers of neurons.

In order to investigate the difference in the performance, starting at one hidden layer, we increase the number of neurons from 2 to 128. Each multiplies two times the preceding one (2^n with $n = 1..7$). When we obtain the best number of neurons, let say k, for example, we begin to increase the number of hidden layers from 2 to 5 with k neurons for each hidden layer to observe the changes in the prediction results. The experiments for hyperparameters search are done on Scientific_Articles dataset. As exhibited from Figure 3, the network receives 3431 attributes of Scientific_Articles dataset as the input following by a hidden layer including 16 neurons and produces 9 outputs corresponding to the predicted probabilities of 9 topics for the classification. After selecting hyperparameters from the experiments, we keep the number of neurons and one hidden layer for prediction tasks on five datasets while the number of nodes of the input layer and output layer can be vary depending on the considered dataset.

The MLP models which perform the binary classification are implemented Sigmoid function to do prediction tasks. The Sigmoid function [33] usually appears in the output layers of Deep learning architectures. It transforms the input values which lie in the domain $\mathbb{R}$ to outputs have the domain in [0,1]. The Sigmoid function is also called “squashing" because this function squashes any input in the range of (-inf,-inf) to the range of [0,1]. When we shifted to gradient-based learning, the Sigmoid was considered as a natural selection due to its smooth and differentiable approximation to a thresholding unit. The Sigmoid function is given by the formula:

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{4}$$

where, x denotes data after being computed by the preceded neural layer.

For multi-classification problems, we use softmax function (Equation 5) with k classes. The Softmax function normalizes an input value into a vector of values that follows a probability distribution whose total sums up to 1.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j^k e^{x_j}} \tag{5}$$

The activation function namely, ReLU [34], is also implemented in our architecture. ReLU follows the formula:

$$f(x) = max(0,x) \tag{6}$$

where, x denotes data after being processed by the preceded neural layer.

[2] N. T. M. Huyen, V. X. Luong and L. H. Phuong, “VnTokenizer”, 2010. http://vntokenizer.sourceforge.net/

Table 1: Five considered datasets descriptions

Data set	#Instances	#Attributes	#Classes	Language
Reuters_Corn	2,158	1,503	2	English
School_Text_Books	1,786	2,566	4	English
Turkish_News_Articles	3,600	5,693	6	Turkish
Scientific_Articles	650	3,431	9	Vietnamese
VnExpress_Newsletters	10,000	3,266	10	Vietnamese

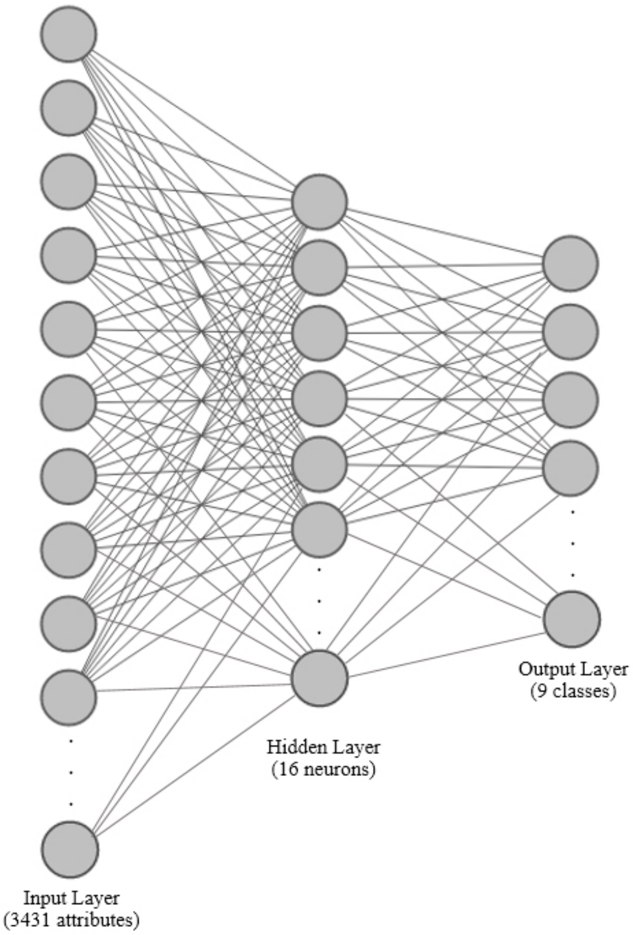

Figure 3: The proposed MLP architecture conducted from tune parameters experiments on Scientific_Articles dataset

ReLU is the most widely used activation function for deep learning architectures with state-of-the-art results to date. ReLU helps models to produce better performance and generalization in deep learning compared to the Sigmoid and Tanh activation functions. It represents a nearly linear function, so this activation function preserves the properties of linear models that made them easy to optimize, with the gradient-descent method [35, 36].

ReLu holds a role as an activation function to transform the output of the preceded hidden neural layer. ReLU helps to improve neural networks by speeding up training. Gradients of logistic and hyperbolic tangent models are lower than the positive portion of the ReLU. That means the positive portion is updated more rapidly as training progresses.

In order to reduce overfitting issues, we consider implementing Early Stopping technique with a patience epoch of 5. This means that the loss cannot be improved after 5 consecutive epochs, the learning will be stopped. Otherwise, the learning will be continued to run to 10 epochs. The network is implemented with Adam optimizer function, use a batch size of 100 and a default learning rate of 0.001.

4 Experimental results

4.1 Data Description

This study used five experimental datasets in three various languages (including English, Turkish, and Vietnamese) as described in Table 1. The reuters corn is available at UCI repository[3] for binary classification tasks. The School text books of 11th and 12th grade which is available at Kaggle Website[4] with four topics. A collection of Turkish news and articles dataset can be downloadable at UCI repository [5] including 3600 samples on 6 categories. The Scientific articles of a university and VnExpress Newsletters in Vietnam were used in our previous work at ACOMP 2019 [13] include 650 samples, 3431 features and 10000 samples, 3266 features, respectively. The considered numbers of classes also vary from binary classification to 10-class classification. The largest dataset is VnExpress_Newsletters with 10000 samples on 10 different topics.

We also face imbalanced datasets issues where the number of samples of some classes is much more than other classes. For example, a class of Reuters_corn dataset occupies to 97% while only 3% is for the other.

The performances of classifiers are examined by average AUC on 3-fold cross-validation. The folds are the same for all classifiers, i.e. training and test sets were identical for each classifier. AUC is a reliable metric for evaluating classifiers where data are not balanced. AUC is widely used in numerous studies to examine the performance of prediction tasks and it is reliable metric to measure the performance of prediction.

4.2 Hyper-parameter turning

In order to select appropriate parameters for the MLP models, we run the experiments with various configurations of MLP architec-

[3] https://storm.cis.fordham.edu/ gweiss/data-mining/datasets.html

[4] https://www.kaggle.com/deepak711/4-subject-data-text-classification

[5] https://archive.ics.uci.edu/ml/datasets/TTC-3600:%20Benchmark%20dataset%20for%20Turkish %20text%20categorization

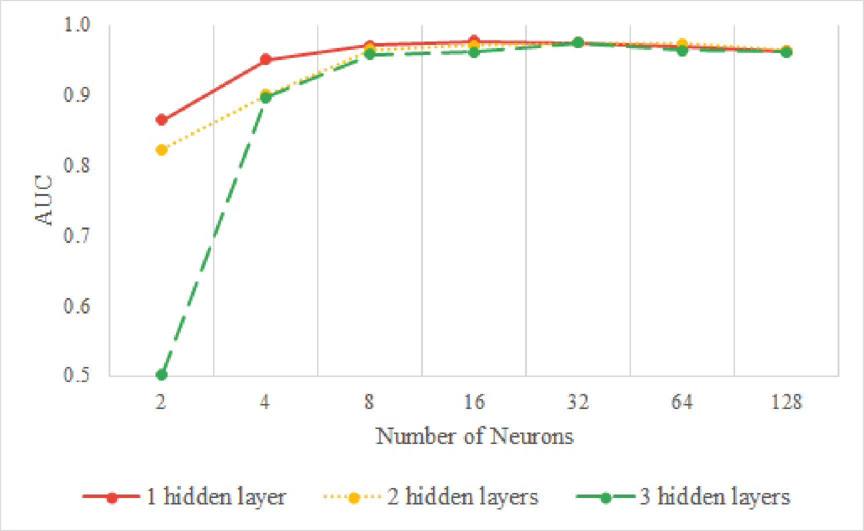

Figure 4: Hyper-parameter search for the Number of Hidden Layers

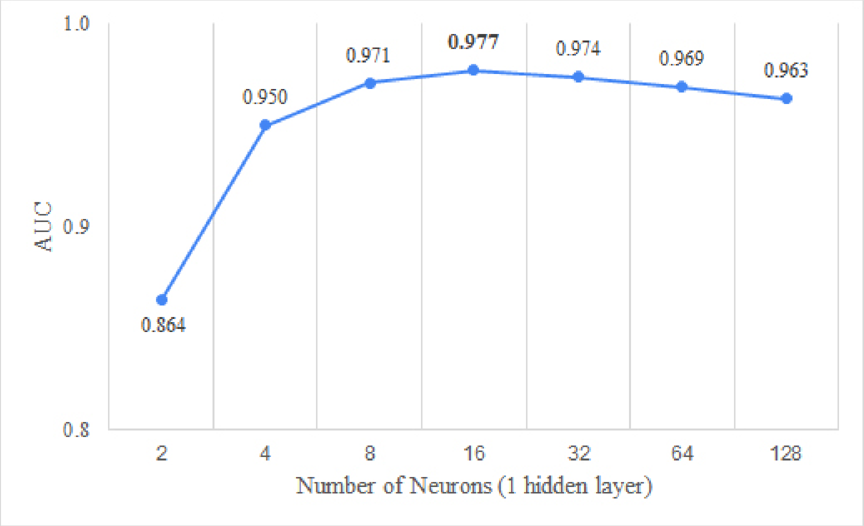

Figure 5: Hyper-parameter search for the Number of Neurons

tures.

In Figure 4 and Figure 5, we reveal the performance of different configurations of MLP on Scientific_Articles dataset. The results show the performance is rising according to the width of the MLP. However, the performances reach peaks which are vary depending on the number of hidden layers used. After reaching the peak, the performance tends to go down when we keep increasing the number of neurons. As seen from Figure 4, with one hidden layer, we obtain the best with 16 neurons while using two or three hidden layers achieve the highest performance with 32 neurons per hidden layer. We noted that the performance decreases when we add more layers due to overfitting, but these differences are trivial when the number of neurons per hidden layer is high.

In most of the cases, the architectures of MLP with one neural layer obtain greater performance than the architectures with more hidden layers (see details performance of the number of neurons with one hidden layer in Figure 5). However, with a large number of neurons, we can see the performance of various numbers of neural layers are nearly the same. Our results exhibit that the shallow architecture of MLP including one hidden layer with 16 neurons reaches the best performance. The number of hidden layers and the number of neurons conducted from the experiments in hyperparameter tuning, we implement an MLP architecture with one hidden layer including 16 neurons to run the learning and validation on the other 4 datasets.

The experiments of five considered datasets are presented and compared with various machine learning methods in Section 4.3.

4.3 Topic classification Results of various machine learning algorithms on five considered datasets

Previous work [13] we showed that SVM works very well for automatic topic classification in an online submission system, however, in this work we have continued to improve and showed that using Deep Learning approach the results even better. Since the data sets are imbalanced, we report the AUC instead of the accuracy metric as

Table 2: Performance Comparison in AUC (Area under the ROC Curve) of various machine learning approaches on five considered datasets

Data set	**Classifier**	**AUC**
Reuters_Corn	MLP	**0.991**
	SVM	0.811
	Decision Tree	0.813
School_Text_Books	MLP	**0.999**
	SVM	0.991
	Decision Tree	0.928
Turkish_News_Articles	MLP	**0.962**
	SVM	0.949
	Decision Tree	0.871
Scientific_Articles	MLP	**0.977**
	SVM	0.965
	Decision Tree	0.819
VnExpress_Newsletters	MLP	**0.990**
	SVM	0.985
	Decision Tree	0.876

reported in Table 2. We have also used Decision Tree as a baseline for comparison.

Table 2 reveals the topic classification performances in AUC of three different machine learning algorithms on five considered datasets. It is easy to see that MLP outperforms other algorithms. In most cases, Decision Tree algorithm gives the worst result while SVM holds the second place. The classification performances of MLP are promising results which all achieve over 0.960 in AUC. Three datasets of them reach over 0.990 while the article classification in Turkish reveals the lowest performance but this result is still high with an AUC of 0.962. An example of visualization for the AUC and precision-recall are presented in Figures 6 and 7, other datasets are quite similar.

We also present the results in confusion matrices of School_Text_Books dataset to observe how different in the performances between MLP and SVM are in Table 3.

As shown from the results above, MLP outperforms SVM and we might like to know how different between them. We can see the difference is that SVM performs worse than MLP on the class where owns the minimum number of samples among the considered classes (these numbers are formatted with blue and bold, revealed in Table 3). With 98 samples for the class of “geography", this number is compared to other classes to see that we face imbalanced issues in data. However, MLP achieves a promising classification result comparing to SVM on the class with much fewer samples. Similar results are also revealed in other datasets. This is expected to bring to a reliable result in practical cases where we usually meet imbalanced dataset issues.

Binary classification tasks on Reuters_Corn dataset reveal the same results where Deep learning approach also reaches a better performance on the class with fewer samples (see the results in Table 4). The Class indicates whether the entry is related to corn ($b = 1$) or not ($a = 0$). As observed from the results, we collected fewer samples which related to “corn" so we also face imbalanced issue in the dataset. In this case, the class of 1 only occupies 3% compared to 97% samples belonging to the class of 0. The same result with multi-classes classification tasks, we obtain better performance for the binary classification tasks with MLP on the class with fewer samples (these numbers are formatted with blue and bold, revealed in Table 4) when we compare to SVM algorithm.

Experimental results in this work showed that the MLP is more suitable than the SVM in case of imbalanced data where the minority class is more interesting to predict. This is also the reason why we have selected the AUC as a measure instead of the Accuracy [14,15].

For the training times, the MLP was completed the training stage in a couple of minutes for the datasets using in this study, so it does not a matter for an online system where we can automatically set-up a training schedule after a time interval (e.g., the model can be automatically updated after one day or other intervals depending on the real number of submissions).

5 Conclusion

Leveraging techniques of natural language processing and machine learning algorithms, we presented a solution to the automated classification of articles to support authors/editors saving their efforts and time for processing articles on the system. Data pre-processing techniques with steps introduced in this work are significantly improved to make the dataset in a standardized format for learning with the three considered algorithms of Multilayer Perceptron, Support Vector Machines and Decision Tree. The experiments are done on five various datasets. The data used vary in the number of features, attributes as well as the number of classes. All datasets reach the performance of over 0.960 in AUC with deep learning models.

As shown from experimental results, deep learning algorithm

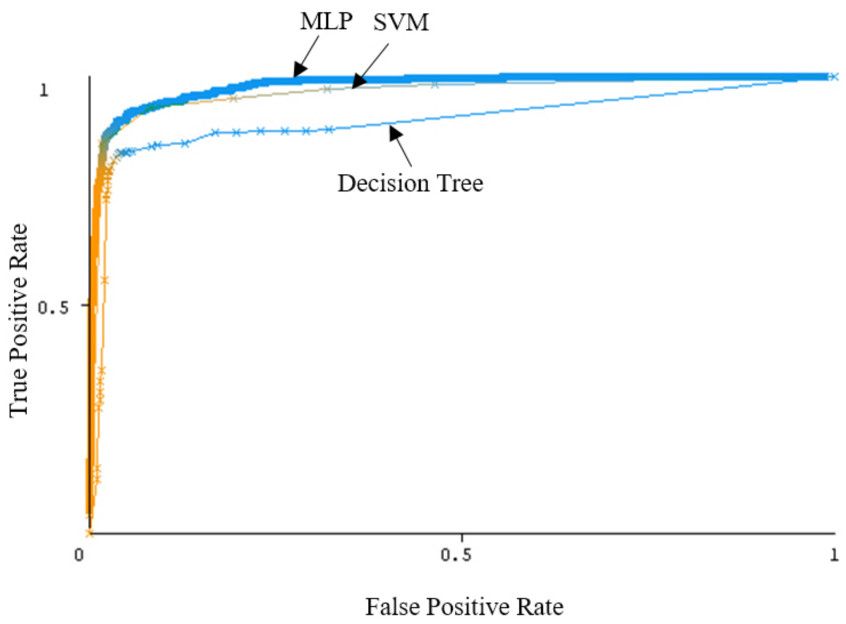

Figure 6: The Area under the ROC (AUC) for Turkish_News_Articles dataset

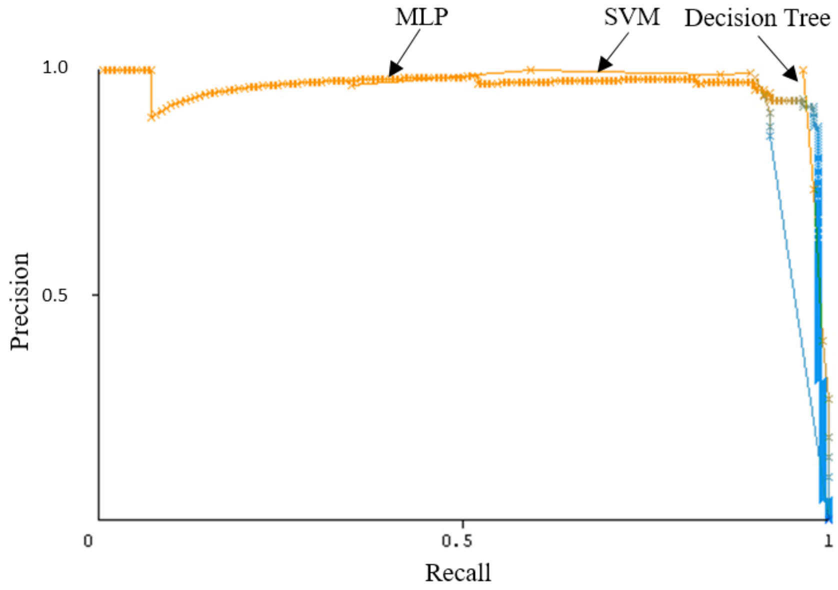

Figure 7: The Precision-Recall for Turkish_News_Articles dataset

Table 3: Confusion matrix comparison of predictions between Multilayer Perceptron and Support Vector Machines on School_Text_Books dataset

	Deep learning					**SVM**			
	Predicted classes					Predicted classes			
Actual classes	a	b	c	d		a	b	c	d
	283	0	0	1	a = accounts	281	1	0	2
	0	626	7	2	b = biology	0	632	3	0
	0	8	**87**	3	c = geography	0	16	**78**	4
	0	4	3	762	d = physics	0	4	1	764

with Multilayer perceptron exhibits better classification performance than the classic machine learning such as Support Vector Machines.

Table 4: Confusion matrix comparison of predictions between Multilayer Perceptron and Support Vector Machines on Reuters_Corn dataset with binary classification

	Deep learning			**SVM**	
	Predicted classes			Predicted classes	
Actual classes	a	b		a	b
	2064	25	a = 0	2088	1
	13	**56**	b = 1	26	**43**

Some parameters are also evaluated to reach promising results in classification tasks. The results show that the proposed model is feasible to extract information and stratify articles automatically whenever a document is submitted to the system. We continue to find a solution for larger datasets in further research.

The proposed architecture of Multilayer perceptron is rather shallow with one neural layer. A vast of hyper-parameters of MLP are evaluated and we see that the performance tends to be saturated when we increase both the number of hidden layers as well as the number of neurons. Further studies should take into sophisticated architectures to improve performance in document categorization tasks.

Conflict of Interest The authors declare no conflict of interest.

References

[1] K. Thaoroijam, “A Study on Document Classification using Machine Learning Techniques,” International Journal of Computer Science Issues, **11**(1), 217–222, 2014.

[2] Y. Li, L. Zhang, Y. Xu, Y. Yao, R. Y. K. Lau, Y. Wu, “Enhancing Binary Classification by Modeling Uncertain Boundary in Three-Way Decisions,” IEEE Transactions on Knowledge and Data Engineering, **29**(7), 1438–1451, 2017, doi:https://doi.org/10.1109/TKDE.2017.2681671.

[3] F. Sebastiani, “Machine Learning in Automated Text Categorization,” **34**(1), 1–47, 2002, doi:https://doi.org/10.1145/505282.505283.

[4] Y. Yang, J. O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization,” in Proceedings of the Fourteenth International Conference on Machine Learning, ICML ’97, 412–420, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, doi:https://dl.acm.org/doi/10.5555/645526.657137.

[5] C. C. Aggarwal, C. Zhai, A Survey of Text Classification Algorithms, 163–222, Springer US, Boston, MA, 2012, doi:10.1007/978-1-4614-3223-4_6.

[6] M. A. Bijaksana, Y. Li, A. Algarni, “A Pattern Based Two-Stage Text Classifier,” in P. Perner, editor, Machine Learning and Data Mining in Pattern Recognition, 169–182, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, doi:https://doi.org/10.1007/978-3-642-39712-7_13.

[7] J. Chen, H. Huang, S. Tian, Y. Qu, “Feature selection for text classification with Naïve Bayes,” Expert Systems with Applications, **36**(3, Part 1), 5432 – 5435, 2009, doi:https://doi.org/10.1016/j.eswa.2008.06.054.

[8] M. Haddoud, A. Mokhtari, T. Lecroq, S. Abdeddaïm, “Combining supervised term-weighting metrics for SVM text classification with extended term representation,” Knowledge and Information Systems, **49**, 909–931, 2016, doi:https://doi.org/10.1007/s10115-016-0924-1.

[9] A. Chouchoulas, Q. Shen, “A Rough Set-Based Approach to Text Classification,” in N. Zhong, A. Skowron, S. Ohsuga, editors, New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, 118–127, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, doi:https://doi.org/10.1007/978-3-540-48061-7_16.

[10] A. P. Gopi, R. N. S. Jyothi, V. L. Narayana, K. S. Sandeep, “Classification of tweets data based on polarity using improved RBF kernel of SVM,” International Journal of Information Technology, 2020, doi:https://doi.org/10.1007/s41870-019-00409-4.

[11] N. Thai-Nghe, Q. D. Truong, “An Approach for Building a Semi-automatic Online Consultancy System,” in 2015 International Conference on Advanced Computing and Applications (ACOMP), 51–58, 2015, doi:https://doi.org/10.1109/ACOMP.2015.11.

[12] B. E. Boser, I. M. Guyon, V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” in Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, 144–152, ACM Press, 1992, doi:https://doi.org/10.1145/130385.130401.

[13] T. T. Dien, B. H. Loc, N. Thai-Nghe, “Article Classification using Natural Language Processing and Machine Learning,” in 2019 International Conference on Advanced Computing and Applications (ACOMP), 78–84, 2019, doi:https://doi.org/10.1109/ACOMP.2019.00019.

[14] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, “Learning from class-imbalanced data: Review of methods and applications,” Expert Systems with Applications, **73**, 220 – 239, 2017, doi:https://doi.org/10.1016/j.eswa.2016.12.035.

[15] N. Thai-Nghe, Z. Gantner, L. Schmidt-Thieme, “A new evaluation measure for learning from imbalanced data,” in The 2011 International Joint Conference on Neural Networks, 537–542, 2011, doi:https://doi.org/10.1109/IJCNN.2011.6033267.

[16] S. Bahassine, A. Madani, M. Al-Sarem, M. Kissi, “Feature selection using an improved Chi-square for Arabic text classification,” Journal of King Saud University - Computer and Information Sciences, **32**(2), 225 – 231, 2020, doi:https://doi.org/10.1016/j.jksuci.2018.05.010.

[17] S. Larabi Marie-Sainte, N. Alalyani, “Firefly Algorithm based Feature Selection for Arabic Text Classification,” Journal of King Saud University - Computer and Information Sciences, **32**(3), 320–328, 2020, doi:https://doi.org/10.1016/j.jksuci.2018.06.004.

[18] C.-H. Tsai, “MMSEG: A word identification system for Mandarin Chinese text based on two variants of the maximum matching algorithm,” Avaible on internet at http://www. geocities. com/hao510/mmseg, 2000.

[19] K. Gábor, D. Buscaldi, A.-K. Schumann, B. QasemiZadeh, H. Zargayouna, T. Charnois, “SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers,” in Proceedings of The 12th International Workshop on Semantic Evaluation, 679–688, Association for Computational Linguistics, New Orleans, Louisiana, 2018, doi:http://dx.doi.org/10.18653/v1/S18-1111.

[20] S. Sarkar, N. Ejaz, M. Kumar, J. Maiti, “Root Cause Analysis of Incidents Using Text Clustering and Classification Algorithms,” in P. K. Singh, B. K. Panigrahi, N. K. Suryadevara, S. K. Sharma, A. P. Singh, editors, Proceedings of ICETIT 2019, 707–718, Springer International Publishing, Cham, 2020, doi:https://doi.org/10.1007/978-3-030-30577-263.

[21] P. Harjule, A. Gurjar, H. Seth, P. Thakur, “Text Classification on Twitter Data,” in 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), 160–164, 2020, doi:https://doi.org/10.1109/ICETCE48199.2020.9091774.

[22] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. A. Chenaghlu, J. Gao, “Deep Learning Based Text Classification: A Comprehensive Review,” ArXiv, **abs/2004.03705**, 2020.

[23] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, M. Rehan, “A comparative review on deep learning models for text classification,” Indonesian Journal of Electrical Engineering and Computer Science, **19**(1), 325–335, 2020, doi:https://doi.org/10.11591/ijeecs.v19.i1.pp325-335.

[24] Q. Qin, W. Hu, B. Liu, “Feature Projection for Improved Text Classification,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 8161–8171, Association for Computational Linguistics, Online, 2020, doi:http://dx.doi.org/10.18653/v1/2020.acl-main.726.

[25] F. Sovrano, M. Palmirani, F. Vitali, "Deep Learning Based Multi-Label Text Classification of UNGA Resolutions," ArXiv, **abs/2004.03455**, 2020.

[26] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, M. Rehan, "Text classification based on gated recurrent unit combines with support vector machine," International Journal of Electrical and Computer Engineering, **10**(4), 3734–3742, 2020, doi:http://doi.org/10.11591/ijece.v10i4.pp3734-3742.

[27] M. Belazzoug, M. Touahria, F. Nouioua, M. Brahimi, "An improved sine cosine algorithm to select features for text categorization," Journal of King Saud University - Computer and Information Sciences, **32**(4), 454 – 464, 2020, doi:https://doi.org/10.1016/j.jksuci.2019.07.003.

[28] E. Lin, Q. Chen, X. Qi, "Deep reinforcement learning for imbalanced classification," Applied Intelligence, **50**(8), 2488–2502, 2020, doi:10.1007/s10489-020-01637-z.

[29] D. Chai, W. Wu, Q. Han, F. Wu, J. Li, "Description Based Text Classification with Reinforcement Learning," ArXiv, **abs/2002.03067**, 2020.

[30] C. S. Perone, "Machine learning: Cosine similarity for vector space models (Part III)," URL:http://blog.christianperone.com, 2019.

[31] J.-Y. Chang, I.-M. Kim, "Analysis and Evaluation of Current Graph-Based Text Mining Researches," Advanced Science and Technology Letter, **42**, 100–103, 2013, doi:http://dx.doi.org/10.14257/astl.2013.42.2.

[32] H. Saif, M. Fernandez, Y. He, H. Alani, "On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter," in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 810–817, European Language Resources Association (ELRA), Reykjavik, Iceland, 2014.

[33] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," ArXiv, **abs/1811.03378**, 2018.

[34] V. Nair, G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in J. Fürnkranz, T. Joachims, editors, Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, 807–814, Omnipress, 2010.

[35] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, G. E. Hinton, "On rectified linear units for speech processing," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 3517–3521, 2013, doi:https://doi.org/10.1109/ICASSP.2013.6638312.

[36] G. E. Dahl, T. N. Sainath, G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 8609–8613, 2013, doi:https://doi.org/10.1109/ICASSP.2013.6639346.