Computers, Materials & Continua
DOI:10.32604/cmc.2021.015761
Article

Tech Science Press

Deep Learning Multimodal for Unstructured and Semi-Structured Textual Documents Classification

Nany Katamesh, Osama Abu-Elnasr* and Samir Elmougy

Faculty of Computers and Information, Department of Computer Science, Mansoura University, 35516, Egypt
*Corresponding Author: Osama Abu-Elnasr. Email: mr_abuelnasr@mans.edu.eg
Received: 05 December 2020; Accepted: 23 January 2021

Abstract: Due to the availability of a huge number of electronic text documents from a variety of sources representing unstructured and semi-structured information, the document classification task becomes an interesting area for controlling data behavior. This paper presents a document classification multimodal for categorizing textual semi-structured and unstructured documents. The multimodal implements several individual deep learning models such as Deep Neural Networks (DNN), Recurrent Convolutional Neural Networks (RCNN) and Bidirectional-LSTM (Bi-LSTM). The Stacked Ensemble based meta-model technique is used to combine the results of the individual classifiers to produce better results, compared to those reached by any of the above mentioned models individually. A series of textual preprocessing steps are executed to normalize the input corpus followed by text vectorization techniques. These techniques include using Term Frequency Inverse Term Frequency (TFIDF) or Continuous Bag of Word (CBOW) to convert text data into the corresponding suitable numeric form acceptable to be manipulated by deep learning models. Moreover, this proposed model is validated using a dataset collected from several spaces with a huge number of documents in every class. In addition, the experimental results prove that the proposed model has achieved effective performance. Besides, upon investigating the PDF Documents classification, the proposed model has achieved accuracy up to 0.9045 and 0.959 for the TFIDF and CBOW features, respectively. Moreover, concerning the JSON Documents classification, the proposed model has achieved accuracy up to 0.914 and 0.956 for the TFIDF and CBOW features, respectively. Furthermore, as for the XML Documents classification, the proposed model has achieved accuracy values up to 0.92 and 0.959 for the TFIDF and CBOW features, respectively.

Keywords: Document classification; deep learning; text vectorization; convolutional neural network; bi-directional neural network; stacked ensemble

1 Introduction

Due to the wide variety of the types of the documents circulating over the internet used in large scale of different applications, identifying the type of document is a critical task for the classification models in order to simplify further operations. Textual semi-structured and unstructured documents have many differences related to their nature which include the structure of the textual representation, degree of ambiguity, degree of redundancy, degree of using punctuation symbols and use of idioms and metaphors [1]. Therefore, intensive preprocessing steps are required to get acceptable classification results through using textual representation techniques.

In addition, document classification is a process of effectively managing large volumes of documents through assigning one or more documents to a specific class from a set of predefined classes. Formally, let $D = \{d_1, d_2, \ldots, d_n\}$ the set of all documents of a size n documents and $C = \{c_1, c_2, \ldots, c_m\}$ the set of predefined classes of *m* classes [2].The document classification task can be also modeled as $f : D \rightarrow C$ that assigns one document d_i to a specific class, c_i. Furthermore, it engages various fields including Natural Language Processing (NLP), machine learning and information retrieval to work altogether to conduct the classification of the textual resources [3].

Moreover, machine learning algorithms, such as Deep Neural Network (DNN) [4,5], Recurrent Neural Network (RNN) [4,5], Convolutional Neural Network (CNN) [4,5], Recurrent CNN (RCNN) [6,7], Long short-Term Memory (LSMT) model [4,8] and Bidirectional LSTM (Bi-LSTM) [9,10], are used to train the document classification models based on the word embedding feature vectors extracted from the textual documents. Besides, term Frequency Inverse Term frequency (TF-IDF) [11–15] and Continuous Bag-of-Words (CBOW) [16–19] are popular text vectorization techniques that generate hand-crafted feature vectors.

The main issue with the classification of text documents relates to the great diversity in the nature of documents that require special kinds of manipulations. Although there have been an increasing body of efforts using DL approaches for handling such issue, most of these approaches are designed for dealing with a certain type of data, while others have ignored the relationships between data that affect the expressive power of the extracted features. Thus, there is a need to develop a generic approach for textual documents classification across a wide range of data types with a variety of complex structures.

Therefore, this paper aims to develop an automatic document classification model for categorizing semi-structured and un-structured textual resources using the Deep Learning (DL) techniques based on various text vectorization techniques. Tokenization and various text normalization techniques are used at the preprocessing level. Furthermore, TF-IDF and CBOW are used at the feature level. Additionally, DNN, LSTM and Bi-LSTM are used at the classification level.

Furthermore, the remainder of this paper is organized as follows: The researchers highlight and summarize the related literature review in Section 2. Then, Section 3 discusses the proposed approach in details. Next, Section 4 presents the experimentation results. Finally, the conclusions are demonstrated in Section 5.

2 Literature Review

2.1 Document Classification Approaches

Document classification has two main different approaches: Manual and automatic classification. The first approach is both expensive and time consuming. However, it provides the user with a great control over the process. The user identifies the relationships between documents

and handles the classification issues. On the other hand, the second approach ends up in faster and more objective classification. It applies content-based matching of one or more predefined categories to documents. In addition, automatic document classification can be accomplished through using one of the following three classification models: Supervised, unsupervised and rule-based classification.

First, in the supervised learning classification, the training model is based on using a small training set of predefined input–output sample documents. This is in an attempt to generalize the categorization task and deduce the classification rules to precisely classify new emergency documents.

Second, in the unsupervised learning classification, patterns are discovered and documents are categorized based on similar words and phrases. The most similar documents are the ones that have more attributes in common.

Third, in the rule-based classification, a set of linguistic rules that define the relationships between the input dataset and their associated categories are formulated and parsed. It is most suitable for predicting data containing a mixture of numerical and qualitative features. Moreover, it is very accurate for small document sets, where the classification results are always based on the predefined rules. However, the task of defining rules can be tedious for large document sets with many categories.

2.2 Related Work

In this sub-section, the researchers highlight the previous literature studies that covered the contributions of the researchers in various areas of research related to the classification process, including feature representation and vectorization and individual and multimodal classification.

2.2.1 Feature Representation and Vectorization

Huang et al. [20] have presented a statistical feature representation method that extracts the most descriptive terms in a document. It also assesses the importance of the word through counting the number of times it occurs in each document and assigning it to the feature space. This method ignores the semantic values of the words and word relationships in each sentence. Therefore, it leads to poor similarity results.

In addition, Melamud et al. [21] have presented context2vec neural architecture which uses word2vec's CBOW architecture with a major enhancement achieved through implementing bidirectional LSTM instead of its native context modeling. This model is an unsupervised approach that handles embedding procedures based on large corpora and produces high quality word representation to learn a generic embedding function for variable length contexts.

Yang et al. [22] have also improved feature representation through getting the semantic and syntactic relations among words and providing rich dictionary resources that can cover all aspects of the NLP tasks. This model generates both definitions and example sentences of target words. The experimental results prove that the model has achieved high performance with regard to both definition modeling and usage modeling tasks. Nevertheless, it still needs more enhancements to generate more meaningful example sentences.

2.2.2 Individual Deep Learning Classifiers

Yao et al. [23] have proposed a Graph Convolution Neural Network (GCN) method for text classification. It is used to achieve strong classification performances with a small proportion of labeled documents, interpretable words and document node embedding. This model consists

of a knowledge graph, where each node refers to an object category and input represented as word embedding of nodes for predicting class. It also uses a single GCN layer with a larger neighborhood which includes both one-hop and multi-hops nodes in the graph to overcome over-smoothing. However, this method is weak with regard to learning representation on a large scale of unlabeled text data.

Moreover, Naqvi et al. [24] have developed a roman Urdu news headline classifier based on different individual machine learning techniques, Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Long short term memory (LSTM) and Convolutional Neural Network (CNN), to classify news into relevant categories on which further analysis and modeling can be done. Firstly, the news dataset is collected using scraping tools. Then, a phonetic algorithm is used to control lexical variation and test news from different websites. The experimental results prove that the MNB classifier has achieved the best accuracy among the other mentioned classifiers.

Yoon [25] has proposed a convolutional neural network model for sentence classification. This model uses a single convolution layer after extracting word embedding for tokens in the input sequence. It has achieved acceptable results on multiple benchmarks using several variants of hyperparameter tuning and static vectors, compared to other DL models that utilize complex pooling schemes.

Furthermore, Zhang et al. [26] have implemented character-level convolutional networks (ConvNets) for text classification. This model encodes characters using one-hot encoding scheme to convert each numerical categorical entry in the dataset into columns of either zeros or ones based on the number of categories. These encoded characters have been fed as inputs to the deep learning architecture with multiple convolution layers. This model proves that character-level convolutional networks achieve competitive results with regard to large scale datasets.

2.2.3 Multimodal Deep Learning Classifiers

Zulqarnain et al. [27] have proposed a classification model based on a combination of Gated Recurrent Unit (GRU) and Support Vector Machine (SVM). They have replaced Softmax activation function in the output layer with GRU. This model has achieved remarkable results particularly when the size of the storage is limited. It has also overcome the issues of vanishing and explosion of gradient.

Haralabopoulos et al. [28] have proposed an automated sentiment classification model used to categorize human-generated content. This model consists of several multi-label DNN classification architectures and two ensembles. The first architecture is a simple CNN with fully connected layers. The second architecture integrates a Gated Recurrent Unit (GRU) with a convolution layer. The third architecture implements TFIDF and a DNN with three fully connected layers. This model has made the best use of these articulated architectures to improve classification results without hyper-parameters tuning or data over-fitting.

Kowsari et al. [29] have also proposed a classification model called Random Multimodal Deep Learning (RMDL) that concatenates standard DL architectures in order to develop robust and accurate architectures for classification tasks. Their constructive model is based on three architectures: CNN, RNN and DNN. The output is generated using majority vote on output of these architectures. The results prove the effectiveness of this model.

Moreover, Ding et al. [30] have proposed a model with multi-layer RNN called Densely Connected Bidirectional LSTM (DC-Bi-LSTM) for text classification. It has used LSTM to encode a sequence of input. In each layer, the hidden states have been represented as a reading memory.

This model has made improvements over the traditional Bi-LSTM, achieved high performance and improved information flow in large tasks. Besides, the researchers expect that the performance may be improved in case of including the implementation of dense Bi-LSTM module instead of the Bi-LSTM encoder.

Furthermore, Wang et al. [31] have proposed a classification model based on a combination of the Dynamic Semantic Representation model and the Deep Neural Network model (DSRM-DNN). Firstly, it generates a model to capture the context of words and selects semantic words dynamically where each word's attribute has been assigned a weight to be quantified. Secondly, it has fed these features as elements to the text classifier that is composed of deep belief network and back-propagation neural network. This model improves the speed and accuracy of text classification, taking into consideration the value of the low-frequency words and new words.

In addition, Cireşan et al. [32] have proposed a multi-model neural networks classifier that is composed of multi-column deep neural networks as combination architectures of DNN and Convolutional Neural Networks (CNN). Moreover, CNN empowers the DNN max-pooling layer by using feed-forward networks with convolutional layers to include local and global pooling layers and, hence, improve the classification results.

3 The Proposed Model

The proposed supervised automatic document classification model is adopted to categorize semi-structured and un-structured textual documents using DL techniques. It is decomposed of three subsequence stages: The textual data preprocessing, text vectorization and document classification. Fig. 1 shows this proposed framework.

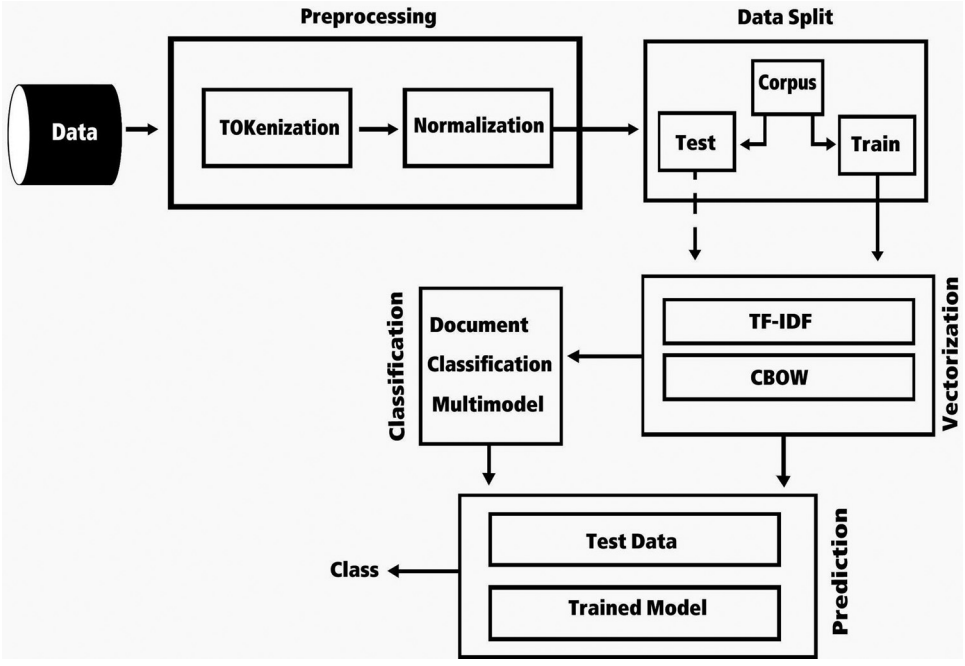

Figure 1: The proposed document classification framework

3.1 Textual Data Preprocessing

Once the data is imported from the corpus, it is automatically preprocessed to be suitable as an input to the classification model. Textual data preprocessing involves two basic steps: text tokenization and text normalization. Algorithm 1 illustrates the tasks required to be completed during the preprocessing process.

Algorithm 1: Textual data preprocessing

```
Function DataPreprocessing (Doc) return FilteredList
Input:
    Doc, a full text documents in form of raw sentences
Output:
    FilteredList, a list of preprocessed documents, initially empty
Variables:
    W, every word in tokenized list
    TokenizedList, list of tokens
    LowercasedList, list of lower case words
    RootWord, word after lemmatization method
    FilteredList, list of preprocessed documents, initially empty
Begin
    TokenizedList = Tokenize(Doc)
    LowercasedList = ToLowerCase(TokenizedList)
    Foreach W in TokenizedList
        Check if (W is not Stop_Word)
            W = Keep-Slang-Abbreviation(W)
            RootWord = Lemmatize(W)
            FilteredList.Append(RootWord)
        Endif
    Endforeach
    return FilteredList
End
```

3.2 Text Vectorization

In order to convert the text data into the corresponding suitable numeric form acceptable to be processed by DL techniques, TFIDF and CBOW models are used to convert the raw text data into their corresponding numbers.

3.2.1 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a numerical statistic approach that aims to measure the importance of a word to a textual document in a corpus (i.e., dataset) [15]. It also acts as a weighting factor in information retrieval and text mining issues. The higher the TF-IDF value is, the more the words will be in the document.

The TF-IDF weight assigns a weight to each term in a document depending on both its Term Frequency (TF) and its Inverse Document Frequency (IDF). It can be obtained through multiplying the values of the both terms, as given in Eq. (1).

$$w_{i,j} = tf_{i,j} \cdot idf_{i,D} \tag{1}$$

where $w_{i,j}$ is TF-IDF value of word i in document j. TF refers to the ratio of the number of times a word occurred in a document to the total number of words in the document, which can be obtained by Eq. (2).

$$tf_{i,j} = \frac{f_{i,j}}{n_j} \tag{2}$$

where $f_{i,j}$ is the frequency of word i in document j. n_j is the total number of words in document j.

IDF acts as a measure of how much information the word provides, it is calculated via Eq. (3).

$$idf_{i,D} = \log \frac{|D|}{|\{d \in D : i \in d\}|} \tag{3}$$

where $|D|$ is the total number of documents, $|\{d \in D : i \in d\}|$: is the number of documents containing the word *i*; if a number of this term is zero, it becomes $1 + |\{d \in D : i \in d\}|$

3.2.2 Continuous Bag-of-Words (CBOW) Model

CBOW is a predictive DL model to map words to vectors and find out the word embedding. This is in order to capture contextual and semantic similarities [18]. Let $W = \{w_{i-n}, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_{i+n}\}$, CBOW tries to predict the target given its surrounding context words. It can be modeled as $f : X \rightarrow Y$, where $Y = w_i$ represents the target word while $X = W - w_i$ represents the context surrounding words.

3.3 Textual Documents Categorization

This paper builds an effective document classification multimodal to categorize big corpus textual documents. This multimodal is a stacked ensemble combination of several individual DL techniques: DNN, RCNN and Bi-LSTM. Fig. 2 shows the structure of the proposed classification multimodal.

3.3.1 Deep Neural Network (DNN)

The DNN architectures feed-forward multilayer architectures. The researchers' implementation of the DNN is basically as a discriminatively trained model that uses ReLU as an activation function. The input is a chain of word embedding features. Furthermore, the output layer houses neurons equal to the number of classes and uses Softmax function.

In addition, the data input (500×50) is generated from an embedding vectorization layer that has passed to five consequent levels of hidden layers; and there are 512 nodes in each hidden layer. Each hidden level is decomposed of both a dropout layer and a dense layer. A dense layer represents a matrix vector multiplication of trainable parameters that implements the ReLU activation function, as given in Eq. (4). Moreover, a dropout layer has been used for setting the trainable parameters to be zero with probability. Next, the output layer of size 3 has been used, where the generative output is multi-class classification that uses softmax as an activation function, as stated in Eq. (5).

$$f(z) = \max(0, x) \tag{4}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \forall j \in \{1, \ldots, k\} \tag{5}$$

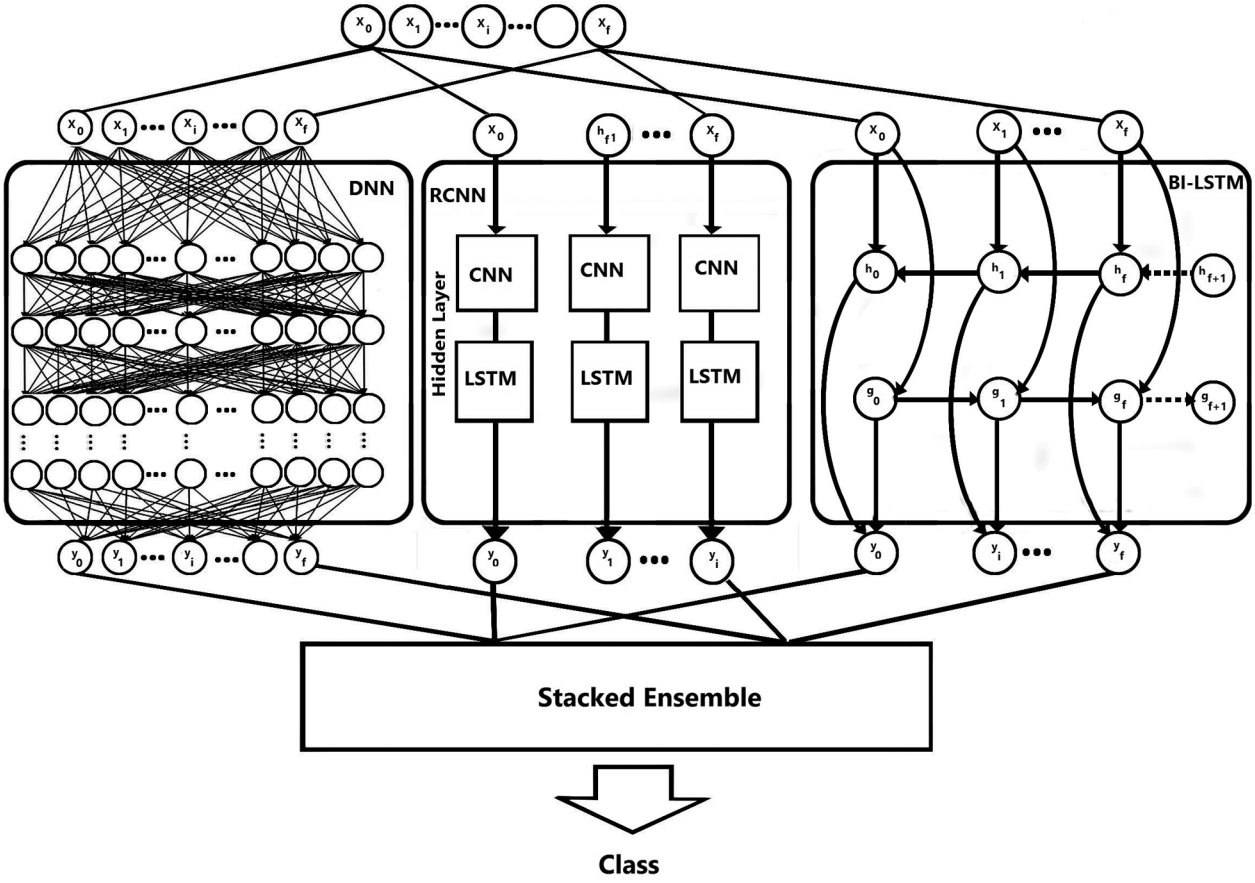

Figure 2: The proposed classification multimodal

3.3.2 Recurrent Convolutional Neural Network (RCNN)

This technique is a combination of RNN and CNN in order to capture the contextual information with the recurrent structure and to construct the representation of the text using the CNN technique.

The data input (500×50) is generated from an embedding vectorization layer that has passed to the hidden combination layer ofthe CNN and RNN techniques. The CNN consists of four consequent levels of convolution layers (4-Conv1D), with 256 filters with a kernel size = 2. Besides, the ReLU activation function is followed by four consequent levels max-pooling (4-MaxPooling1D). The RNN consists of four consequent levels of LSTM (4-LSTM) with 256 number of nodes passed to the two levels of the dense layer using the ReLU activation function. After that, the output is generated using Eq. (5).

3.3.3 Bidirectional-LSTM

Bidirectional LSTMs (Bi-LSTMs) are an extension of typical LSTMs that are intended to enhance the performance of the classification model. Bi-LSTMs train two LSTMs instead of one LSTM on the input sequence. The first provides feed-forward from the input sequence to the output, while the other provides feed-backward in a reverse order. The idea behind this technique

is to allocate the forward state part to be responsible for the positive time direction and the backward state part to keep track of the opposite direction.

The data input (500×50) is generated from an embedding vectorization layer that has passed to the bidirectional layer. The bidirectional layer uses 100 memory cells in parallel in the both LSTMs to generate an output with a shape of 30 data points wide and 256 data points' height. Next, the time distributed layer is used to generate an output shape with 30 data points wide and 256 data points' height. The generated shape is passed to the flatten layer that produces an output shape of 7680 points; and that is finally fed as an input to the dense layer to find the closest output class.

3.3.4 Stacked Ensemble Technique

This technique is intended to combine a set of previously trained models (DNN, RCNN and Bi-LSTM) and merge them with the concatenation function to generate the final classification outcome [33].

4 Experimental Results

4.1 Dataset Description

The training set consists of three textual classes: XML, JSON and PDF documents that are collected by web-crawling different websites. A total of 50.000 documents are randomly picked and allocated for JSON and XML classes, taken from the following websites: https://catalog.data.gov/dataset?res_format=JSON and https://www.sba.gov/sites/default/files/data.json. For XML and JSON requests, an internal logger is used that collects 100.000 of such requests. Additionally, regarding the PDF class, the dataset consists of 11,228 newswires from Reuters labeled over 46 topics.

4.2 Evaluation Metrics

Multiple performance and evaluation criteria are used to ensure the improvement of the proposed model, in comparison to the other existing models. Precision [34] act as Positive Predictive Value (PPV), as stated in Eq. (6).

$$\text{PPV (Precision)} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{6}$$

Recall [34] act as True Positive Rate (TPR), as given in Eq. (7).

$$\text{Recall (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

F-measure [34] is calculated by the harmonic means between precision and recall as illustrated in Eq. (8).

$$\text{F-measure} = \frac{2 \times \text{Precision (PPV)} \times \text{Recall (TPR)}}{\text{Precision (PPV)} + \text{Recall (TPR)}} \tag{8}$$

4.3 Experiments

In this section, a series of experiments are done to evaluate the performance of the researchers' revised individual classifiers and the results of the proposed combined document classification multimodal.

4.3.1 Experimental Results of DNN Model

Tabs. 1–3 illustrate the precision, recall and f-measure of the experimentation results of the individual DNN model for predicting PDF, JSON and XML documents, respectively. These results are based on the researchers' suggested hyper parameters that include the following values: the numbers of epochs, the learning rate values, the batch size values and the numbers of hidden layers. First, Tab. 1 illustrates the classification results for predicting PDF documents in the case of using the TFIDF and CBOW text vectorization techniques. Second, Tab. 2 demonstrates the classification results for predicting JSON documents in the case of using the TFIDF and CBOW text vectorization techniques. Finally, Tab. 3 shows the classification results for predicting XML documents in the case of using the TFIDF and CBOW text vectorization techniques.

Table 1: Classification results of DNN for predicting PDF documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Learning rate	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
50	0.001	2	100	0.509	0.519	0.5007	0.549	0.599	0.607
		2	150	0.593	0.609	0.617	0.669	0.69	0.69
		4	100	0.558	0.593	0.605	0.618	0.6243	0.605
		4	150	0.69	0.695	0.68	0.709	0.724	0.71
	0.00146	2	100	0.75	0.55	0.59	0.6287	0.6365	0.629
		2	150	0.67	0.66	0.68	0.738	0.754	0.72
		4	100	0.83	0.841	0.839	0.846	0.85	0.84
		4	150	0.796	0.81	0.825	0.836	0.84	0.825
75	0.001	2	100	0.836	0.827	0.816	0.850	0.86	0.85
		2	150	0.788	0.820	0.808	0.8087	0.802	0.819
		4	100	0.71	0.73	0.729	0.74	0.754	0.76
		4	150	0.83	0.825	0.84	0.856	0.860	0.855
	0.00146	2	100	0.745	0.725	0.714	0.7635	0.765	0.774
		2	150	0.806	0.816	0.82	0.8466	0.846	0.858
		4	100	0.737	0.727	0.73	0.7767	0.786	0.753
		4	150	0.858	0.846	0.852	0.857	0.880	0.8712

4.3.2 Experimental Results of the RCNN Model

Tabs. 4–6 illustrate the precision, recall and f-measure of the experimentation results of the individual RCNN model for predicting PDF, JSON and XML documents, respectively. These results are based on the researchers' suggested hyper parameters that include the following values: The numbers of epochs, the learning rate values, batch size values and the numbers of hidden layers. Tab. 4 illustrates the classification results for predicting PDF documents in the case of using the TFIDF and CBOW text vectorization techniques. Moreover, Tab. 5 clarifies the classification results for predicting JSON documents in the case of using the TFIDF and CBOW text vectorization techniques. Finally, Tab. 6 displays the classification results for predicting XML documents in the case of using the TFIDF and CBOW text vectorization techniques.

Table 2: Classification results of DNN for predicting JSON documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Learning rate	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
50	0.001	2	100	0.539	0.54	0.53	0.60	0.619	0.627
		2	150	0.56	0.59	0.57	0.69	0.67	0.69
		4	100	0.60	0.63	0.615	0.637	0.629	0.635
		4	150	0.59	0.612	0.609	0.71	0.721	0.71
	0.00146	2	100	0.75	0.55	0.59	0.6287	0.6365	0.629
		2	150	0.69	0.71	0.698	0.748	0.763	0.756
		4	100	0.84	0.85	0.849	0.854	0.84	0.86
		4	150	0.80	0.81	0.827	0.826	0.85	0.83
75	0.001	2	100	0.84	0.838	0.82	0.864	0.856	0.86
		2	150	0.80	0.830	0.828	0.817	0.831	0.829
		4	100	0.73	0.753	0.748	0.75	0.764	0.749
		4	150	0.86	0.845	0.861	0.879	0.881	0.867
	0.00146	2	100	0.765	0.758	0.732	0.758	0.763	0.771
		2	150	0.84	0.839	0.85	0.829	0.834	0.881
		4	100	0.773	0.75	0.76	0.757	0.746	0.763
		4	150	0.865	0.859	0.842	0.864	0.870	0.882

Table 3: Classification results of DNN for predicting XML documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Learning rate	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
50	0.001	2	100	0.559	0.56	0.566	0.62	0.63	0.631
		2	150	0.62	0.61	0.60	0.687	0.69	0.70
		4	100	0.62	0.63	0.641	0.657	0.669	0.65
		4	150	0.608	0.619	0.62	0.709	0.71	0.70
	0.00146	2	100	0.78	0.767	0.75	0.637	0.645	0.639
		2	150	0.687	0.69	0.628	0.768	0.759	0.761
		4	100	0.83	0.82	0.83	0.864	0.87	0.873
		4	150	0.81	0.819	0.83	0.848	0.86	0.84
75	0.001	2	100	0.85	0.847	0.75	0.859	0.873	0.854
		2	150	0.829	0.837	0.849	0.847	0.851	0.848
		4	100	0.75	0.763	0.788	0.753	0.774	0.759
		4	150	0.87	0.85	0.84	0.891	0.873	0.89
	0.00146	2	100	0.75	0.77	0.776	0.748	0.753	0.752
		2	150	0.87	0.857	0.86	0.819	0.82	0.879
		4	100	0.783	0.77	0.89	0.77	0.76	0.753
		4	150	0.867	0.859	0.849	0.861	0.887	0.872

Table 4: Classification results of RCNN for predicting PDF documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Learning rate	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
15	0.001	3	100	0.499	0.50	0.51	0.547	0.612	0.608
		3	150	0.602	0.615	0.627	0.68	0.696	0.701
		5	100	0.59	0.613	0.599	0.608	0.613	0.625
		5	150	0.661	0.663	0.657	0.739	0.745	0.732
	0.00146	3	100	0.73	0.65	0.74	0.637	0.645	0.639
		3	150	0.68	0.69	0.70	0.743	0.761	0.74
		5	100	0.840	0.853	0.852	0.7651	0.862	0.85
		5	150	0.82	0.82	0.835	0.846	0.851	0.835
50	0.001	3	100	0.856	0.837	0.826	0.809	0.853	0.847
		3	150	0.78	0.810	0.828	0.827	0.83	0.82
		5	100	0.72	0.73	0.74	0.75	0.764	0.79
		5	150	0.846	0.835	0.82	0.867	0.87	0.863
	0.00146	3	100	0.745	0.725	0.794	0.7651	0.776	0.781
		3	150	0.806	0.816	0.82	0.866	0.846	0.858
		5	100	0.737	0.727	0.73	0.7767	0.7867	0.753
		5	150	0.835	0.840	0.852	0.856	0.866	0.8712

Table 5: Classification results of RCNN for predicting JSON documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Learning rate	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
15	0.001	3	100	0.518	0.53	0.509	0.559	0.607	0.614
		3	150	0.619	0.62	0.618	0.698	0.71	0.796
		5	100	0.61	0.62	0.60	0.619	0.62	0.631
		5	150	0.664	0.690	0.681	0.712	0.739	0.740
	0.00146	3	100	0.738	0.722	0.73	0.847	0.85	0.890
		3	150	0.731	0.718	0.71	0.72	0.71	0.73
		5	100	0.891	0.829	0.879	0.89	0.862	0.887
		5	150	0.859	0.863	0.855	0.873	0.863	0.873
50	0.001	3	100	0.831	0.869	0.873	0.889	0.851	0.878
		3	150	0.78	0.810	0.872	0.875	0.849	0.83
		5	100	0.72	0.73	0.761	0.749	0.781	0.78
		5	150	0.846	0.835	0.834	0.857	0.863	0.83
	0.00146	3	100	0.745	0.725	0.756	0.773	0.781	0.891
		3	150	0.806	0.816	0.834	0.865	0.858	0.88
		5	100	0.737	0.727	0.78	0.778	0.771	0.83
		5	150	0.835	0.840	0.862	0.891	0.865	0.882

Table 6: Classification results of RCNN for predicting XML documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Learning rate	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
15	0.001	3	100	0.51	0.52	0.507	0.539	0.591	0.617
		3	150	0.601	0.615	0.623	0.689	0.692	0.684
		5	100	0.562	0.591	0.625	0.628	0.63	0.645
		5	150	0.673	0.65	0.78	0.718	0.725	0.702
	0.00146	3	100	0.78	0.63	0.602	0.637	0.645	0.639
		3	150	0.691	0.681	0.692	0.781	0.764	0.74
		5	100	0.809	0.82	0.834	0.856	0.84	0.863
		5	150	0.803	0.809	0.832	0.840	0.839	0.851
50	0.001	3	100	0.841	0.843	0.826	0.849	0.838	0.819
		3	150	0.798	0.819	0.8091	0.807	0.812	0.807
		5	100	0.720	0.731	0.7287	0.753	0.751	0.771
		5	150	0.865	0.865	0.859	0.868	0.859	0.865
	0.00146	3	100	0.791	0.791	0.749	0.758	0.775	0.782
		3	150	0.840	0.839	0.828	0.879	0.846	0.870
		5	100	0.791	0.787	0.79	0.775	0.781	0.769
		5	150	0.878	0.875	0.865	0.880	0.871	0.89

Table 7: Classification results of Bi-LSTM for predicting PDF documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Element vector	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
50	50	50	100	0.85	0.80	0.83	0.85	0.83	0.87
		100	150	0.82	0.84	0.81	0.82	0.85	0.87
		50	100	0.85	0.83	0.80	0.85	0.89	0.89
		100	150	0.69	0.70	0.69	0.71	0.73	0.70
	100	50	100	0.82	0.84	0.81	0.82	0.82	0.89383
		100	150	0.79	0.78	0.79	0.80	0.85	0.907
		50	100	0.60	0.63	0.62	0.83	0.82	0.856
		100	150	0.86	0.88	0.87	0.90	0.89	0.9025
100	50	50	100	0.70	0.68	0.67	0.91	0.902	0.908
		100	150	0.863	0.81	0.80	0.94	0.90	0.92
		50	100	0.82	0.80	0.82	0.92	0.91	0.90
		100	150	0.75	0.78	0.79	0.91	0.92	0.905
	100	50	100	0.80	0.79	0.785	0.80	0.82	0.815
		100	150	0.80	0.88	0.87	0.88	0.90	0.919
		50	100	0.81	0.80	0.82	0.91	0.90	0.89
		100	150	0.87	0.89	0.89	0.95	0.93	0.93

4.3.3 Experimental Results of Bi-LSTM Model

Tabs. 7–9 demonstrate the precision, recall and f-measure of the experimentation results of the individual Bi-LSTM model for predicting PDF, JSON and XML documents, respectively. These results are based on the researchers' suggested hyper parameters that include different numbers of epochs, element vectors, batch size values and numbers of hidden layers. Tab. 7 illustrates the classification results for predicting PDF documents in the case of using the TFIDF and CBOW text vectorization. Furthermore, Tab. 8 shows the classification results for predicting JSON documents in the case of using the TFIDF and CBOW text vectorization techniques. Finally, Tab. 9 clarifies the classification results for predicting XML documents in the case of using the TFIDF and CBOW text vectorization techniques.

Table 8: Classification results of Bi-LSTM for predicting JSON documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Element vector	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
50	50	50	100	0.861	0.87	0.859	0.849	0.838	0.88
		100	150	0.825	0.834	0.82	0.85	0.867	0.89
		50	100	0.84	0.85	0.83	0.881	0.88	0.87
		100	150	0.698	0.72	0.68	0.70	0.71	0.72
	100	50	100	0.83	0.829	0.83	0.819	0.828	0.838
		100	150	0.76	0.79	0.88	0.91	0.906	0.91
		50	100	0.69	0.65	0.67	0.85	0.83	0.86
		100	150	0.89	0.90	0.89	0.91	0.903	0.92
100	50	50	100	0.723	0.80	0.71	0.859	0.91	0.907
		100	150	0.836	0.808	0.83	0.915	0.926	0.919
		50	100	0.825	0.816	0.827	0.908	0.918	0.903
		100	150	0.59	0.87	0.88	0.908	0.905	0.906
	100	50	100	0.86	0.90	0.88	0.83	0.819	0.807
		100	150	0.814	0.806	0.874	0.891	0.908	0.908
		50	100	0.808	0.84	0.83	0.908	0.92	0.91
		100	150	0.88	0.809	0.909	0.942	0.929	0.93

4.3.4 Experimental Results of the Proposed Document Classification Multimodal

In addition, Tab. 10 illustrates the precision, recall and f-measure of the classification results of the document classification multimodal for the unstructured PDF class, semi-structured JSON class and semi-structured XML class in the case of using the TFIDF and CBOW text vectorization techniques. The results indicate that the performance of the proposed multimodal based on the stacked ensemble technique gives better results, compared to those reached by any of those models individually.

The high results found by the study are due to applying the proposed technique, which is a combination of the RNN and CNN techniques. Actually, it makes use of the advantages of the both techniques. It is also intended to capture the contextual information with the recurrent structure. Moreover, it helps construct the representation of the text through using the CNN and Bi-directional Neural Networks that allocate the forward state part to be responsible for the positive time direction and the backward state part to keep track of the opposite direction.

Finally, the researchers have used the stacked ensemble technique to combine a set of trained meta-models. The outputs of the previously trained models are merged with the concatenation function to generate the final classification outcome. Prior to that, the researchers made feature extraction using Word2Vec and TF-IDF Word2Vec to capture the position of the words in the text (syntactic) and to capture the meaning of the words (semantics). Therefore, word2vector, according to the achieved results above, shows the best outcomes.

Table 9: Classification results of Bi-LSTM for predicting XML documents

Hyper parameters				Classification results-TFIDF			Classification results-CBOW		
Epochs	Element vector	#Hidden layers	Patch size	Precision	Recall	F-measure	Precision	Recall	F-measure
1-4 50	50	50	100	0.859	0.865	0.863	0.851	0.848	0.890
		100	150	0.819	0.828	0.819	0.858	0.870	0.881
		50	100	0.853	0.849	0.852	0.890	0.889	0.868
		100	150	0.708	0.719	0.691	0.713	0.708	0.729
	100	50	100	0.81	0.819	0.809	0.829	0.838	0.848
		100	150	0.79	0.819	0.859	0.928	0.919	0.929
		50	100	0.65	0.66	0.65	0.90	0.89	0.91
		100	150	0.853	0.856	0.86	0.91	0.903	0.92
100	50	50	100	0.88	0.909	0.91	0.90	0.926	0.91
		100	150	0.85	0.838	0.84	0.932	0.916	0.940
		50	100	0.809	0.82	0.83	0.918	0.928	0.92
		100	150	0.61	0.60	0.62	0.91	0.89	0.909
	100	50	100	0.87	0.89	0.89	0.839	0.849	0.85
		100	150	0.84	0.86	0.849	0.89	0.90	0.91
		50	100	0.86	0.86	0.85	0.928	0.91	0.92
		100	150	0.923	0.91	0.87	0.939	0.94	0.93

Table 10: Classification results of the multimodal based on the TFIDF and CBOW techniques

Vectorization technique	Unstructured PDFclass			Semi-structured JSON class			Semi-structured XML class		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
TF-IDF	0.905	0.926	0.934	0.914	0.928	0.909	0.920	0.930	0.919
CBOW	0.959	0.940	0.940	0.956	0.960	0.950	0.959	0.960	

5 Conclusion

The classification task is an important issue with regard to machine learning, given the growing number and size of datasets that need sophisticated classification. Therefore, the researchers have proposed an automatic document classification multimodal for categorizing multi-typed textual documents. In addition, the proposed multimodal combines three individual classifiers: DNN, RCNN and Bi-LSTM, based on the stacked ensemble technique. The purpose of adopting this multimodal is to make managing and sorting the textual documents easier. This is especially useful for publishers, financial institutions, insurance companies or any industry that deals with large

amounts of content. Moreover, the proposed automatic document classification model realizes a significant reduction in the time consumed on manual data entry, in costs and also in the turnaround time for document processing. Additionally, it ends up in an accurate, efficient and more objective classification where it applies semantic classification based on deep learning classification. Furthermore, the evaluation results show that a combination of the models and the parallel learning architecture used has consistently resulted in accuracy higher than that obtained through using conventional approaches and individual deep learning models.

Finally, the researchers aim in future studies to empower the feature extraction and representation stage through using an effective glove technique. Moreover, the researchers intended to extend the feature level through embedding multivariate analysis and dimensionality reduction technique to specify which subspace the data approximately lies in and to find uncorrelated features. In addition, the researchers plan to develop a test data generative model for an automated testing tool and embed the proposed automatic classification model as a pre-integral part of the generative model to classify different kinds of documents before generating the test data for each type.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Madani, O. Boussaid and D. E. Zegour, "Semi-structured documents mining: A review and comparison," *Procedia Computer Science*, vol. 22, pp. 330–339, 2013.

[2] M. Ikonomakis, S. Kotsiantis and V. Tampakas, "Text classification using machine learning techniques," *WSEAS Transactions on Computers*, vol. 4, no. 8, pp. 966–974, 2005.

[3] A. Khan, B. Baharudin, L. H. Lee and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of Advances in Information Technology*, vol. 1, no. 1, pp. 4–20, 2010.

[4] M. Heidarysafa, K. Kowsari, D. E. Brown, K. J. Meimandi and L. E. Barnes, "An improvement of data classification using random multimodel deep learning (RMDL)." *International Journal of Machine Learning and Computing*, vol. 8, no. 4, pp. 298–310, 2018.

[5] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber *et al.,* "Hdltex: Hierarchical deep learning for text classification," in *16th IEEE Int. Conf. on Machine Learning and Applications*, Cancun, Mexico, IEEE, pp. 364–371, 2017.

[6] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.

[7] S. Lai, L. Xu, K. Liu and J. Zhao, "Recurrent convolutional neural networks for text classification," *Twenty-ninth AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, pp. 2267–2273, 2015.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] Z. Hameed and B. Garcia-Zapirain, "Sentiment classification using a single-layered BiLSTM model," *IEEE Access*, vol. 8, pp. 73992–74001, 2020.

[10] B. Jang, M. Kim, G. Harerimana, S. U. Kang and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, pp. 5841, 2020.

[11] A. Aizawa, "An information-theoretic perspective of TF-IDF measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[12] W. Zhang, T. Yoshida and X. Tang, "A comparative study of TFIDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011.

[13] D. Dessì, R. Helaoui, V. Kumar, D. ReforgiatoRecupero and D. Riboni, "TF-IDF *vs.* word embeddings for morbidity identification in clinical notes: An initial study," *1st Workshop on Smart Personal Health Interfaces, SmartPhil, CEUR-WS*, vol. 2596, pp. 1–12, 2020.

[14] F. Rustam, I. Ashraf, A. Mehmood, S. Ullah and G. S. Choi, "Tweets classification on the base of sentiments for US airline companies," *Entropy*, vol. 21, no. 11, pp. 1078, 2019.

[15] Z. Yun-tao, G. Ling and W. Yong-cheng, "An improved TF-IDF approach for text classification," *Journal of Zhejiang University Science*, vol. 6, no. 1, pp. 49–55, 2005.

[16] M. Leszczynski, A. May, J. Zhang, S. Wu, C. R. Aberger *et al.,* "Understanding the downstream instability of word embeddings," in *Proc. of the 3rd MLSys Conf.*, Austin, TX, USA, pp. 262–290, 2020.

[17] T. Menon, "Empirical analysis of CBOW and skip gram NLP models," Bachelor of Science (B.S.) in Computer Science and University Honors, Portland State University, Portland, Oregon, 2020.

[18] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *1st Int. Conf. on Learning Representations*, Scottsdale, Arizona, USA, pp. 1–12, 2013.

[19] A. Novák, L. Laki and B. Novák, "CBOW-tag: A modified CBOW algorithm for generating embedding models from annotated corpora," in *Proc. of the 12th Language Resources and Evaluation Conf.*, Marseille, France, pp. 4798–4801, 2020.

[20] C. H. Huang, J. Yin and F. Hou, "A text similarity measurement combining word semantic information with TF-IDF method," *Chinese Journal of Computers*, vol. 34, no. 5, pp. 856–864, 2011.

[21] O. Melamud, J. Goldberger and I. Dagan, "Context2vec: Learning generic context embedding with bidirectional lstm," in *Proc. of the 20th SIGNLL Conf. on Computational Natural Language Learning*, Berlin, Germany, pp. 51–61, 2016.

[22] H. Yang and C. Zong, "Learning generalized features for semantic role labeling," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 4, pp. 1–16, 2016.

[23] L. Yao, C. Mao and Y. Luo, "Graph convolutional networks for text classification," *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 33, pp. 7370–7377, 2019.

[24] R. A. Naqvi, M. A. Khan, N. Malik, S. Saqib, T. Alyas *et al.,* "Roman urdu news headline classification empowered with machine learning," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1221–1236, 2020.

[25] K. Yoon, "Convolutional neural networks for sentence classification," in *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1746–1751, 2014.

[26] X. Zhang, J. Zhao and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in Neural Information Processing Systems*, vol. 28, pp. 649–657, 2015.

[27] M. Zulqarnain, R. Ghazali, Y. M. Hassim, M. M. Yana and M. Rehan, "Text classification based on gated recurrent unit combines with support vector machine," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 4, pp. 3734–3742, 2020.

[28] G. Haralabopoulos, I. Anagnostopoulos and D. McAuley, "Ensemble deep learning for multilabel binary classification of user-generated content," *Algorithms*, vol. 13, no. 4, pp. 83, 2020.

[29] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi and L. E. Barnes, "Rmdl: Random multimodel deep learning for classification," in *Proc. of the 2nd Int. Conf. on Information System and Data Mining*, Lakeland, Florida, USA, pp. 19–28, 2018.

[30] Z. Ding, R. Xia, J. Yu, X. Li and J. Yang, "Densely connected bidirectional LSTM with applications to sentence classification," in *CCF Int. Conf. on Natural Language Processing and Chinese Computing*, Hohhot, China, Springer, pp. 278–287, 2018.

[31] T. Wang, L. Liu, N. Liu, H. Zhang, L. Zhang *et al.,* "A multi-label text classification method via dynamic semantic representation model and deep neural network," *Applied Intelligence*, vol. 50, no. 8, pp. 2339–2351, 2020.

[32] D. Cireşan and U. Meier, "Multi-column deep neural networks for offline handwritten Chinese character classification," in *Int. Joint Conf. on Neural Networks*, Killarney, Ireland, IEEE, pp. 1–6, 2015.

[33] J. Brownlee, "Deep Learning with Python: Develop Deep Learning Models on Theano and Tensorflow Using Keras," Vermont, Australia: Machine Learning Mastery, 2016. [Online]. Available: https://www.goodreads.com/book/show/34043770-deep-learning-with-python.

[34] M. Sokolova, N. Japkowicz and S. Szpakowicz, "Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation," in *Australasian Joint Conf. on Artificial Intelligence*, Hobart, Australia, Springer, pp. 1015–1021, 2006.