Cloud Computing and Data Science
http://ojs.wiserpub.com/index.php/CCDS/

Review

# Text Classification Using Deep Learning Models: A Comparative Review

**Muhammad Zulqarnain[1,*], Rubab Sheikh[1], Shahid Hussain[1], Muhammad Sajid[1], Syed Naseem Abbas[1], Muhammad Majid[1], Ubaid Ullah[2]**

[1]Faculty of Computing, The Islamia University of Bahawalpur, Punjab, Pakistan
[2]College of Computing, Riphah International University, Faisalabad Campus, Pakistan
Email: zulqarnain@iub.edu.pk

**Received:** 14 August 2023; **Revised:** 7 October 2023; **Accepted:** 7 October 2023

**Abstract:** With the fast popularization and continued development of web pages on the Internet, text classification has become a very serious problem in organizing and managing large amounts of digital text data in documents. The deep learning approaches have been applied in several areas of text classification with comparative and outstanding results. In this article, we analyzed and gave comprehensive reviews of the different deep learning models for text classification tasks. Based on the literature review survey, this paper addresses three various deep learning models and declares their gaps and limitations. We have evaluated the various classification applications and provided a small discussion on the available Deep Neural Networks (DNN) frameworks for the implementation of text datasets. The work presents guidance for future research to regulate more significance that can be distributed for the better area of this research. In summary, our study presented the main implications, identified potential directions for future research, and highlighted the challenges within this specific research field. Additionally, our aim is to acquaint readers with the various subtasks and relevant literature related to the text classification process. By engaging with our discussion, we aspire to inspire readers to explore novel and enhanced techniques for text classification, applicable across diverse domains.

***Keywords*:** deep learning, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Belief Networks (DBN), text classification

## 1. Introduction

With the quick popularization and continued development of the internet online technology such as the World Wide Web and electronic documents in digital format, most online technology and information relies on text form, so text classification (TC) has become the concentration key point of Web information retrieval and information filtering [1]. Most information (over 79%) is stored in text form; text mining is believed to have a high potential commercial value. The knowledge is discovered from many different sources of information; yet, unstructured data texts remain the large easily available source of information [2]. In recent years, the TC techniques have been extensively used in different fields.

Text classification automation has emerged as a crucial challenge for large organizations aiming to handle vast volumes of data. It serves as a pivotal technology for efficiently organizing and managing extensive online resources [3]. Several traditional machine learning and statistical approaches have been proposed for text categorization, such as


DOI: https://doi.org/10.37256/ccds.5120243528

Bayesian classifier, support vector machine (SVM), K-nearest neighbor (KNN), and neural networks [4]. Classification techniques have drawn realization in many applications including image classification, text filtering, spam filtering, email categorization, and text classification [5]. However, the information on websites can offer fast growth and brings it big challenge to the conventional method of web data analysis. Several machine learning techniques have been applied to analyze web data and time series prediction [5] but are facing challenges with the continued increasing amounts of web data.

A consistent demand exists for a methodology capable of extracting valuable insights from a diverse collection of textual documents, each associated with different subjects' classes for specific research purposes, as illustrated in Figure 1.

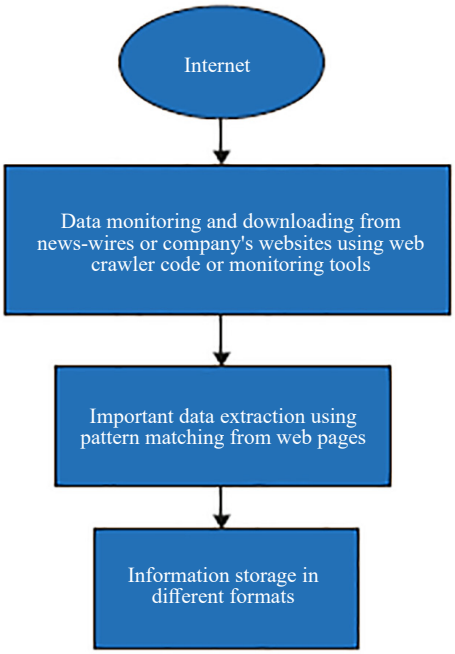


**Figure 1.** Observing and retrieving pertinent textual documents for specific subject categories

Rule-based techniques, depicted in Figure 2, employ predefined rules to categorize text into different classes. For instance, the label "Football" is assigned to any document containing the words "Business", "Technology", or "Sports". However, these approaches necessitate in-depth domain knowledge and pose challenges in maintaining the systems. In contrast, data-driven methods learn to make categorizations based on past data, offering an alternative solution.

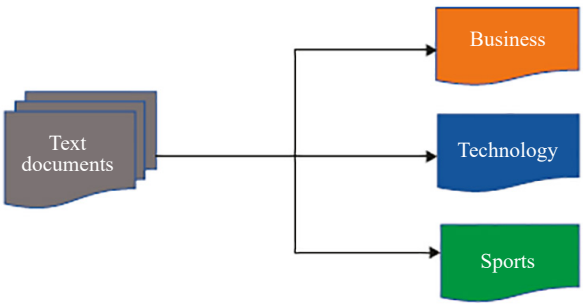


**Figure 2.** During the text classification process, suitable predefined classes or labels to text documents

Recently, deep learning approaches have attained remarkable results in different fields of Natural Language Processing (NLP), such as sentiment analysis, spam filtering, question answering, and text categorization [6]. Now the question arises: how do we determine the best model for text classification tasks among the available options? Based on previous results and model characterizations, we have three contenders: the hierarchical model (CNN), the general-purpose approach (DBN), and the sequential approach (RNN). For challenging NLP categorization tasks, such as text classification, the RNN approach seems to be a preferred choice. Its performance has been demonstrated to outperform as compared to other approaches, especially when dealing with text classification and sentiment analysis tasks that rely on key phrases. On the other hand, the DBN model can also be a suitable option for NLP tasks, including text classification. One of its key strengths lies in its ability to learn multiplex features through hidden layers, allowing it to demonstrate complex functions and patterns within the data. Each hidden layer unit learns statistical connections among units in the lower layer, leading to increasingly intricate representations in the higher layers [7]. As for the RNN approach, it excels in sequence-to-sequence sequential modeling tasks, such as language modeling, and it has the capability to capture flexible contextual dependencies [8]. In conclusion, when selecting the most appropriate model for a specific NLP task such as text classification, one should consider the characteristics and strengths of each model. CNN is a solid choice for challenging NLP classification tasks, while DBN can offer advantages with its ability to learn complex features. Meanwhile, RNN is well-suited for sequential modeling tasks that require capturing context dependencies.

In this study, a comparison was conducted between Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), and two highly practical types of Recurrent Neural Networks (RNN), namely LSTM and GRU, for text classification tasks. The research experiment aimed to systematically analyze their performance in classification. The study identified two main findings from the research experiment: (1) Complementary Information: CNNs and RNNs were observed to provide complementary information for text classification tasks. The choice of architecture that performs better depends on the significance of semantically understanding the entire sequence. However, the research experiment highlighted certain deficiencies of standard RNNs, such as the issues of gradient vanishing and exploding. These issues make the training of RNN challenging in two ways: (i) they are not well-suited for processing very long sequences when using hyperbolic tanh activation function, and (ii) they exhibit instability when using the rectified linear unit (ReLU) as an activation function. Fortunately, RNN variants such as LSTM and GRU demonstrated the ability to overcome these problems. (2) Impact of Hyperparameters: The study also investigated the impact of various hyperparameters on the model's performance. It was observed that changes in the learning rate resulted in comparatively smooth performance variations. On the other hand, altering the batch size and hidden layers size led to significant variations in the results. In conclusion, the study highlighted the complementarity of CNN and RNN for text classification tasks and emphasized the importance of using more advanced RNN types like LSTM and GRU to overcome the gradient vanishing and exploding issues. Moreover, the research experiment demonstrated the sensitivity of model performance to specific hyperparameters such as learning rate, batch size, and hidden layer size.

Our study investigated the three most extensively employed deep learning approaches, namely CNN, RNN, and DBN, which focus on overcoming the above-mentioned issues in text classification tasks. The effectiveness of each architecture depends on the significance of semantically comprehending the entire sequence. Within the realm of deep-learning, DNN has shown remarkable success, and the rapid progress of pre-trained word embeddings has opened up new avenues for natural language processing and various other tasks. The primary objective of this paper is to provide a comparative review and identify the research limitations in the area of text classification by utilizing various deep learning approaches. The main contributions of this study can be summarized as follows:

· We introduced the process and evolution of text classification, and presented the comparative analysis and research on main deep learning approaches based on their model architectures.

· We conduct a comparative review of over 6 widely used text classification datasets.

· Highlighting the advantages and drawbacks of different models utilized in the text classification process.

· Illustrating the research areas where further enhancements can be made to traditional approaches and suggest novel methods along with their potential applications across various domains.

The remaining sections of this paper are structured as follows: Section 2 presents the related work, while Section 3 illustrates the traditional deep learning models. In Section 4, we provided experimental design and dataset description. Gaps and discussion can be found in Section 5. Section 6 summarizes the conclusion and future direction of this study.

## 2. Related work

Classification of text has emerged as a widely adopted application, wherein spoken or written language is systematically sorted based on the content and characteristics present in documents and files [9]. This field, known as text classification, holds a prominent position in Natural Language Processing (NLP). As the volume of electronic documents and digital libraries from diverse sources, text categorization becomes an increasingly challenging task [10]. Managing and standardizing text data has become complex due to the rapid growth of unstructured online information and data. To address this issue, various machine learning and deep-learning algorithms have been developed to effectively process textual data and extract valuable insights from vast collections of information [11]. In recent studies, numerous deep learning approaches have been employed to address text classification challenges [12]. The inherent difficulty in conveying both syntactic and semantic content makes text classification a complex task. Consequently, the final results of text classification are influenced by a combination of classification approaches and feature representation methods.

The most precise techniques between them are the naïve Bayes and SVM. Deep learning models have been largely used for TC such as CNN for sentence classification and image processing. RNN for multi-tasking learning and language modeling, DBN for spam filtering, web and Chinese text classification, and AE have been used for text feature extractions and data mining. In this way, by using RNN-Max Entropy proposed a method for sentence and paraphrase detection. However, the application of a recursive neural tensor model to assess the sentiment of phrases and sentences introduces a fresh analytical approach. In contrast, a different study [13] explored a sequential architecture of a recurrent neural network (RNN) where words were employed as inputs. This approach utilized a bidirectional model along with a Max-pooling layer at its apex, effectively incorporating the RNN for constructing language models. To address the issue of preserving contextual information for classifications, a new RNN approach was proposed, incorporating the use of fast text.

Additionally, this approach enables the acquisition of textual features through the utilization of word embeddings to replace sentences or texts. The FastText linear technique has been widely employed for text categorization. To address memory consumption and training time, Minaee et al. [14] referred to an encoding method that utilizes CNN to learn character-level text representations effectively. For character-level text classification, Londt et al. [15] referred to a model based on character input transformed into fixed-sized one-hot vectors, then processed through a deep CNN approach with six convolutional layers and pooling computations, followed by three fully connected layers. However, the structure of their proposed method performs well mainly on large-scale datasets. Neural networks have been considered by Chung et al. [16] as language models due to their memory and Turing capability. They compared different variants of RNNs including long short-term memory (LSTM) and Gated Recurrent Unit (GRU) in their proposed networks. Moreover, Zhou et al. [17] incorporated a Bidirectional-LSTM (Bi-LSTM) model, which captures valuable text features with different timescales using a two-dimensional max-pooling layer.

Furthermore, the combination of CNN and LSTM has demonstrated remarkable outcomes in answer selection, utilizing an attention-based LSTM. Conversely, in a study comparing word2vec, CNN, GRU, and LSTM for sentiment classification of Russian tweets, the GRU model exhibited superior classification performance over LSTM and CNN [18]. Notably, [19] conducted experimental evaluations and concluded that there is no clear winner between GRU and LSTM in various multiple classification tasks. Across various multiple classification tasks, both models performed similarly, highlighting that tuning hyper-parameters like batch and layer size often play a more critical role than selecting the paradigm architecture. As previously indicated, researchers' summaries pertaining to deep learning models and techniques, along with a comparison of these techniques, assessment criteria, and the utilized datasets, are presented in Table 1. The outcomes and explanations of these findings are already detailed in the related work section.

**Table 1.** Summary of text classification approaches

| Existing studies | Proposed approaches | Comparative approaches | Evaluation criteria | Datasets |
|---|---|---|---|---|
| Kowsari et al. [20] | DNN | DBN, CNN | Accuracy | 1. WOS-11967, 2. WOS-46985 |
| Zulqarnain et al. [21] | ES-GRU | LSTM, GRU, CNN, SVM, NB | Accuracy, precision, recall, f-score, execution time | IMDB, 20NG, Yahoo, AG's News |
| Sarikaya et al.[22] | DBN-3 | DBN, DBN-1, DBN-2, SVM, Boosting | Accuracy | 3.2K and 5.6K sentences datasets |
| Alaa et al. [23] | RNN-Max entropy | GIS, IIS | Recall, precision, F-measure | Arabic text documents |
| Peslak et al. [24] | CNN | MLP, LSTM, CNN + LSTM | Accuracy | IMDB |
| Joulin et al. [25] | Fast-Text | Char-CNN, n-gram, BoW, VDCNN | Accuracy, running time | YFCC100M |
| Wang et al. [26] | CNN-SVM | CNN, KNN, SVM | Accuracy, Recall, F-measure, precision | 20-News Group corpus. |
| Iwasaki et al. [27] | AE + MF/GRU | LBP-TOP/SVM, LBP-TOP/LVM | Accuracy | OuluVS |
| Alshalif et al. [28] | ARDC | RDC, IRDC, SVM, KNN, MLP, MNB | Accuracy, precision, recall, f-score | R-21578, 20NG, TDT2 |

## 3. Deep learning models

In recent years, there has been a remarkable surge of interest in deep learning approaches, which are capable of learning layered and hierarchical representations from high-dimensional data. Deep learning approaches have demonstrated successful applications in the domain of natural language processing (NLP) such as spam filtering, image classification, sentiment analysis and semantic segmentation. The objective of this study is to conduct a review of deep learning approaches by identifying gaps and limitations based on recent literature surveys. The article explores three prominent deep learning approaches, namely RNN, CNN, and DBN. Ultimately, this research concludes by summarizing the findings from the literature review and discussing the limitations of the various deep learning approaches.

### 3.1 *Convolutional neural network*

CNN is an extensively applied deep learning model that was inspired by the visual cortex of animals. Recently, CNN has found significant applications in NLP systems, leading to remarkable achievements. Initially, they were primarily utilized for tasks like image classification, pattern recognition, and text classification. However, researchers have now extended their exploration to other domains, including object detection, text detection, and speech recognition [29]. The integration of CNN approaches with the NLP model to address bias-related challenges. Additionally, CNN has shown exceptional performance in classifying objects within images, showcasing their ability to generalize effectively in the context of image classification [30]. Furthermore, CNNs have been employed in text mining; however, for this purpose, they necessitate an extensive amount of training data.

CNN consists of multiple layers of convolutions that incorporate nonlinear activation functions like ReLU or tanh and apply them to the outcomes. In contrast to classical feed-forward neural networks, where each neuron's input is linked to every output in the subsequent layer (referred to as a fully connected or affine layer), CNNs employ distinct strategies. They utilize convolutions across the input layer to calculate the output, employing local connections to process information from the input layer. Subsequently, each layer employs various kernels, often comprising hundreds or even thousands of filters, to amalgamate the computed outcomes. In the process of pooling or subsampling layers within CNN, as well as during the training phase, the network acquires the appropriate filter sizes according to the specific tasks. Figure 3 illustrated the traditional architecture of CNN model, To demonstrate, in scenarios like image classification [31], a CNN could grasp the skill of identifying edges from the initial raw pixel data in the initial layer. Subsequently, these detected edges might be employed to recognize basic shapes within the second layer. As the layers progress, these shapes can then contribute to the recognition of more complex attributes, such as facial shapes, in the

higher layers. The outcome of this layered approach is then inputted into a classifier that capitalizes on these higher-level features. In conclusion, it has been shown that deep learning employing Convolutional Neural Networks (CNN) can effectively grasp advanced textual concepts from character-level text representations.

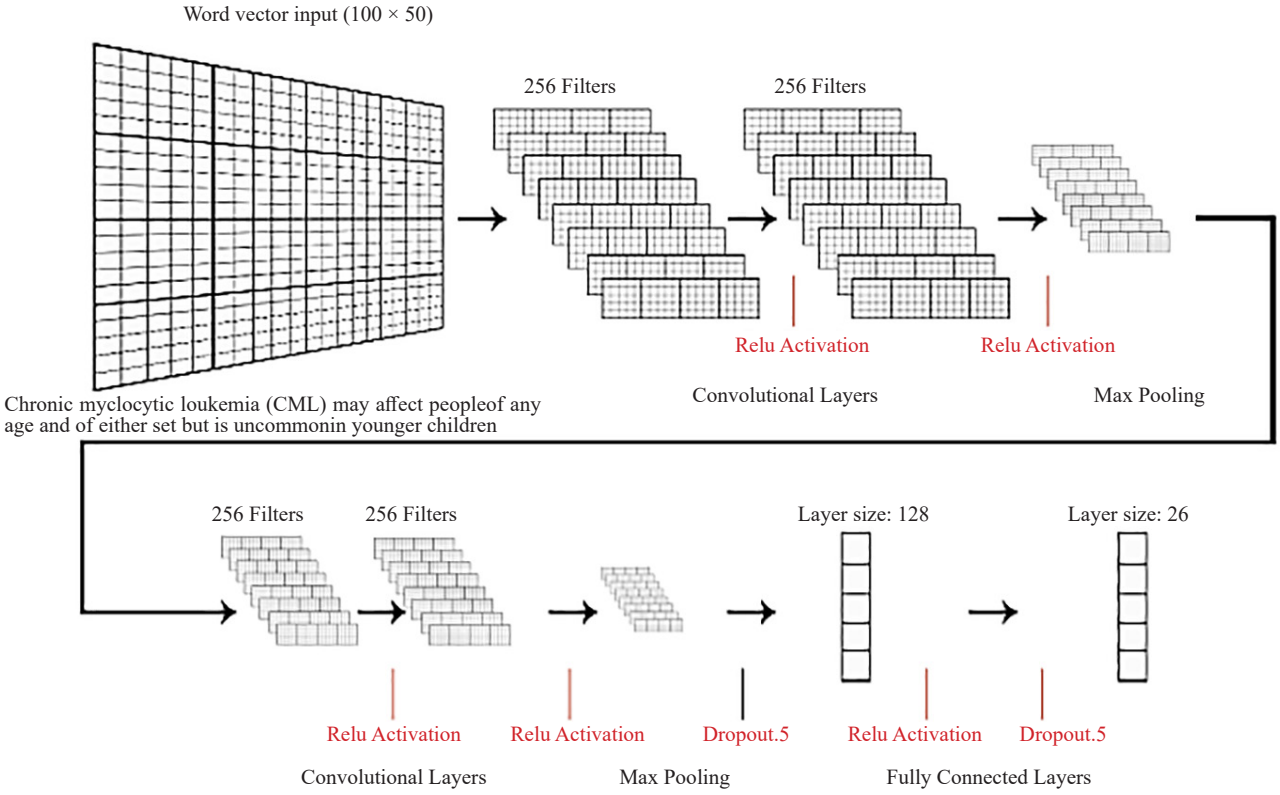


**Figure 3.** Traditional architecture of CNN model

## 3.2 *Recurrent neural network*

The concept of Recurrent Neural Network (RNN) was initially introduced by Hopfield in 1983 [32]. RNN is a type of supervised artificial neural network, that incorporates feedback connections between layers in the structure of a direct cycle. This cyclic arrangement enables RNNs to execute analogous operations for all nodes within a sequence. RNN is one of the most well-known deep learning approaches which performed outstanding results on any sequence of datasets such as speech recognition, audio classification, and sentiment analysis [33]. The utilization of Recurrent Neural Network (RNN) has been prominent in addressing Natural Language Processing (NLP) challenges due to their inherent recurrent structure, which proves to be a highly effective and apt algorithm for handling text of varying lengths [34]. The architecture of RNN holds particular significance in acquiring the capacity to understand temporal dependencies within text, whether they pertain to characters or words. The fundamental architecture of an RNN approach is illustrated in Figure 4. Provided a sequence of word vectors ($X_1$, ..., $X_t$), the process generates a sequence of hidden states ($h_1$, ..., $h_t$). Each hidden state is computed at time step $t$. Consequently, the output can be determined using the following procedure within an RNN framework:

$$O_t = \varphi\left(W_x x_t + U_o h_{t-1}\right) \tag{1}$$

$$H_t^l = \varphi\left(W_x h_{t-1}^l + U_H h_{t-1}^l\right) \tag{2}$$

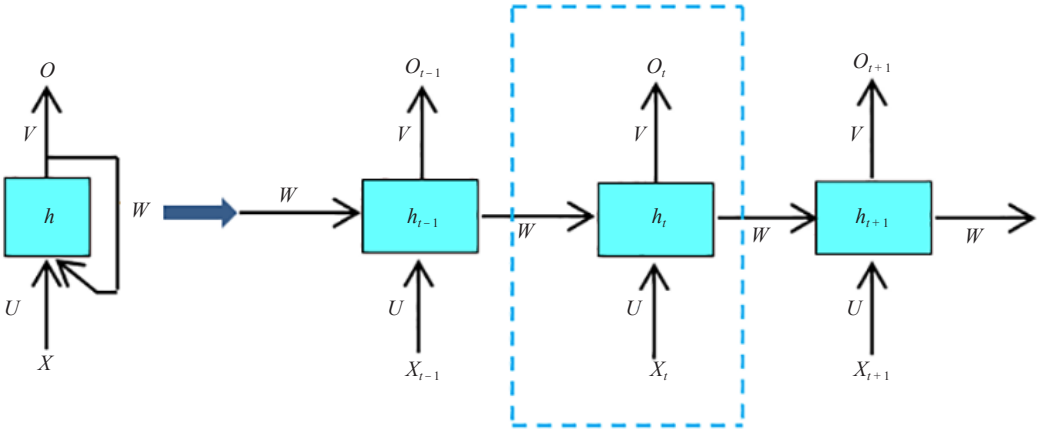


**Figure 4.** The conventional architecture of RNN model

The recurrent weights matrices are represented $U_O$ as $U_H$, while the input-to-hidden weights matrix is denoted as $W_x$. Additionally, $\varphi$ stands for any arbitrary activation function. Equations (1) and (2) illustrate the hidden layer's behavior, indicating its connection with the previous hidden layer activity, $h_{t-1}$. This relationship is nonlinear in nature because of the utilization of the logistic activation function $\varphi(\cdot)$. The RNN model explore saves all past text semantics in the memory of the hidden layer and performs work word to word. The architecture of RNN use memory and it has excellent sequence datasets that's why it has no doubt the capability to capture the semantics of a long sentence [35]. In the realm of deep learning, innovative research areas and gating mechanisms have emerged to enable the creation of potent deep models for Recurrent Neural Networks (RNNs). Among the notable RNN-based variants are the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU).

## 3.3 *Long short-term memory*

An LSTM unit, classified as a form of conventional RNN, was originally presented by German researchers Sepp Hochriter and Juergen in 1997 [36]. The LSTM network stands as a modification of the typical RNN, distinguished by its proficiency in grasping extended sequential data and sustaining error propagation throughout all its layers [37]. Within the LSTM architecture, distinctive internal memory blocks and gated mechanisms are incorporated, which effectively address two widely recognized issues associated with the conventional RNN: the vanishing gradient problem and the exploding gradient problem. In the context of LSTM, these memory blocks encompass memory cells featuring self-connections, thereby retaining the temporal state of the network. Additionally, special multiplicative units are introduced to regulate the flow of information.

In the context of LSTM, memory blocks are composed of memory cells that possess self-connections along with specialized multiplicative units designed to manage information flow. Each LSTM block is comprised of three distinct gates: an input gate, an output gate, and a forget gate [38]. The standard architecture of these LSTM gate blocks is illustrated in Figure 5. From a mathematical perspective, the connections between the inputs and the output gates of an LSTM are determined through a series of subsequent equations.

$$i_t = Sigm(W_{xi}x_t + U_{hi}h_{t-1} + b_i) \tag{3}$$

$$o_t = Sigm(W_{xo}x_t + U_{ho}h_{t-1} + b_o) \tag{4}$$

$$f_t = Sigm(W_{xf}x_t + U_{hf}h_{t-1} + b_f) \tag{5}$$

$$\hat{a}_t = tanh(W_{x\hat{a}}x_t + U_{h\hat{a}}h_{t-1} + b_{\hat{a}}) \tag{6}$$

$$c_t = f_t \times x_{t-1} + i_t \times \hat{a}_t \tag{7}$$

$$h_t = o_t \times tanh(c_t) \tag{8}$$

The training process yields computed weights and biases, denoted as $W_i$, $W_o$, $W_f$, $W_{\hat{a}} \in R^{m \times p}$, $U_i$, $U_o$, $U_f$, $U_{\hat{a}} \in R^{m \times m}$, $b_i$, $b_o$, $b_f$, $b_{\hat{a}} \in R^{m \times 1}$. The * operator signifies element-wise multiplication between vectors. Additionally, the activation functions Sigm and tanh correspond to element-wise logistic sigmoid and hyperbolic tangent functions, respectively.

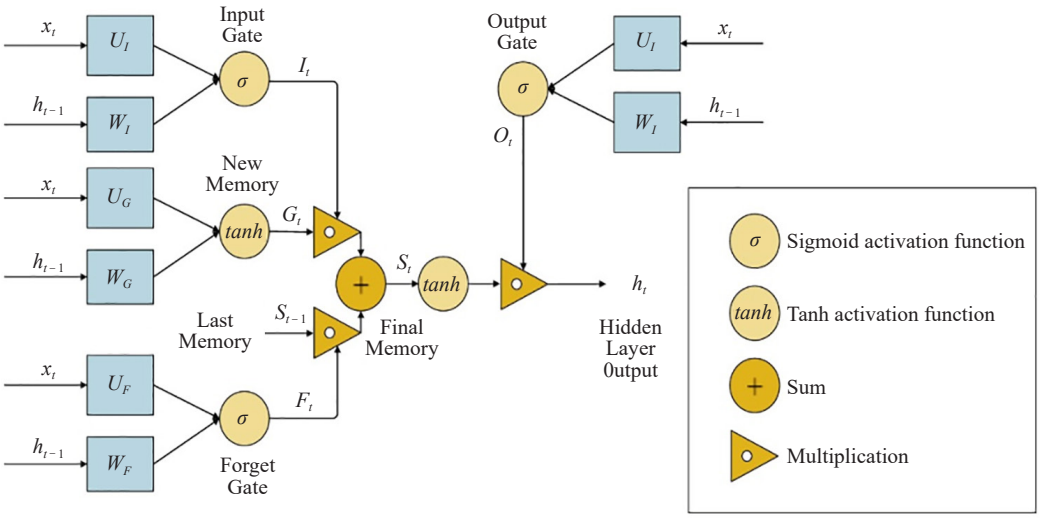


**Figure 5.** LSTM architecture framework diagram

## 3.4 *Gated recurrent unit*

The Gated Recurrent Unit, initially introduced by Chung et al. [39], tackles the prevalent problem of lengthy contextual connections that often result in gradient degradation within conventional, extensive RNN networks. This innovation has since evolved into a modern architecture, referred to as the "two gated mechanism" approach. This approach is aimed at enabling each recurrent unit to adeptly grasp dependencies across different time ranges. GRUs resolve this concern by retaining a form of "memory" from the preceding time step, which significantly aids the network in making accurate future predictions [40]. Figure 6 illustrates the traditional GRU architecture, which showcases how the update and reset gates are interconnected.

Nevertheless, the GRU utilizes its internal memory capacity to store and filter information, combining the input gate and forget gate into a unified update gate with inputs like the previous activation $h_{t-1}$ and the candidate state $h_t$. The GRU consists of three key elements: the update gate, the reset gate, and the candidate state. The corresponding equations for these components are as provided fellows:

Update gate

$$z_t = Sigm\,(W_{xz} x_t) + U_{hz}(h_{t-1}) \tag{9}$$

Reset gate

$$r_t = Sigm\,(W_{xr} x_t) + U_{hr}(h_{t-1}) \tag{10}$$

Candidate state

$$\hat{h}_t = tanh\left(W_{x\hat{h}} x_t\right) + U_{h\hat{h}}\left(r_t \times h_{t-1}\right) \tag{11}$$

Final output memory

$$h_t = \left(1 - z_t\right) \times h_{t-1} + z_t \times \hat{h}_t \tag{12}$$

The elements that need computation while the training is underway consist of $W_{xz}$, $W_{xr}$, $W_{x\hat{h}} \in R^{m \times p}$, $U_{hz}$, $U_{hr}$, $U_{h\hat{h}} \in R^{m \times m}$ and * denoted by element-wise multiplication.

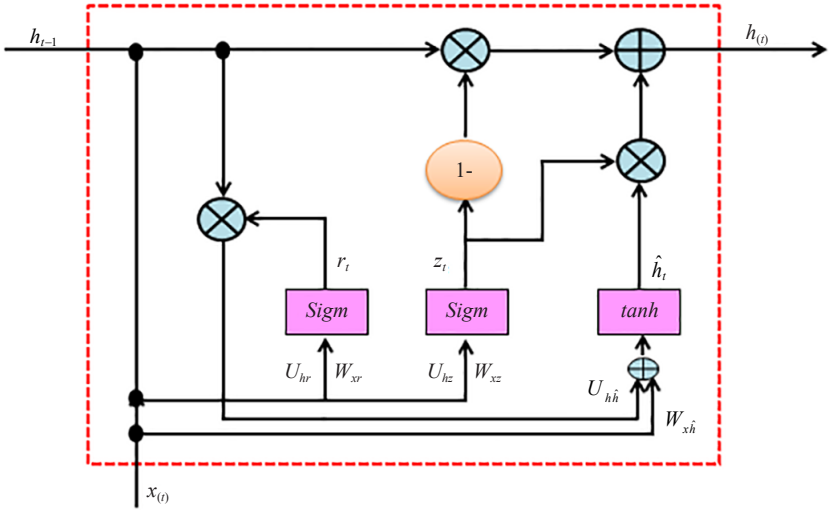


**Figure 6.** Traditional GRU architecture [41]

## 3.5 *Deep belief network*

DBNs have been used in various forms of data in generative deep learning approaches such as multiple areas of text classification. The DBN model obtains more complicated features to express data and can be learned the more features with hidden layers. A generative probabilistic model known as DBN comprises a single visible layer alongside multiple hidden layers [41]. In the DBN framework, each hidden layer unit acquires a statistical understanding of the units within the lower layer. Consequently, the representations in the upper layers tend to exhibit increased complexity. Figure 7 presents the basic structure of DBN and the equation:

Allow $v_i$ and $h_j$ to denote the conditions of visible node $i$ and hidden node $j$, respectively. In the case of binary state nodes, i.e., where $v_i$ and $h_j$ are within the set {0, 1}, $h_j$'s state is established at 1 with particular probabilities.

$$P_{h_j} = p\left(h_j = 1 \mid v\right) = \sigma\left(b_j + \sum_i w_{ij} v_i\right), \tag{13}$$

This likelihood is determined by the logistic sigmoid function $\sigma(x) = 1/(1 + exp(-x))$, where $b_j$ signifies the bias of $j$, and $v_i$ signifies the binary state. The weight $w_{ij}$ corresponds to the connection between $v_i$ and $h_j$.

Typically, the capacity of DBN higher modeling of shallow approaches with the number of the same parameters, but these are very difficult and harder to train, both as deterministic bottom-up discriminative models and top-down probabilistic generative models. Using backpropagation in discriminative training models, the learning process indicates a slow performance with multiple hidden layers and one of the most serious issues is overfitting [42].

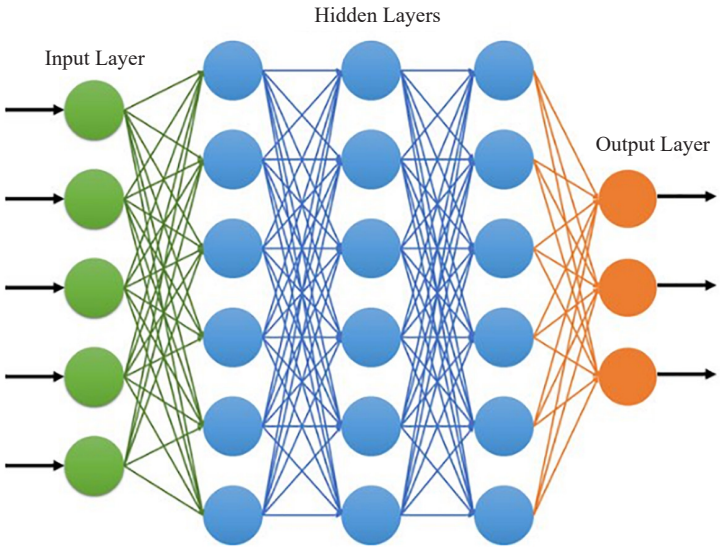


**Figure 7.** DBN architecture

# 4. Experimental design & datasets

An experimental setup was employed to assess the efficacy of four distinct deep learning methods on the text classification datasets. As such, this section provides a concise overview of both the text classification datasets and the experimental configuration.

## 4.1 *Datasets description*

The significant advancements in text classification can mostly be credited to the wealth of labeled datasets accessible. In this section, a summary of these datasets is provided, emphasizing their attributes like domains, diverse groupings, average sentence length, and dataset scale.

### 4.1.1 *Sentiment text classification*

The Stanford Sentiment Treebank (SST) dataset [43] contains movie reviews classified as either "positive" or "negative" sentiments. In this research, we employ this dataset, dividing it into three segments: 6,895 sentences for training, 878 for validation, and 1,830 for testing purposes. Similar to what is mentioned in reference [44], label phrases occurring within training sentences are treated as distinct training instances.

### 4.1.2 *CNAE-9*

In Sem-Eval 2012, specifically on task 7 [45], there is a dataset consisting of 1,080 business documents in free text format. These documents provide descriptions of Brazilian companies and are organized into 9 different subsets. The dataset underwent a preprocessing step where only letters were retained, and prepositions were removed from the texts. The CNAE-9 dataset was further divided into 760 documents for training purposes and 328 documents for testing purposes, with no separate validation set being utilized.

### 4.1.3 *Textual entailment*

In the Stanford Natural Language Inference dataset (SNLI) [46], there are pairs of statements, each comprising a premise and a hypothesis, associated with labels indicating their relationship ("entailment", "contradiction", or "neutral"). After excluding the pairs without labels, the dataset comprises 549,361 pairs for training, 9,814 pairs for validation, and 9,860 pairs for testing.

#### 4.1.4 *Health news in twitter*

The health-related information originates from a selection of open datasets found within the UCI data repository. This data was gathered through the utilization of the Twitter API and encompasses health news items obtained from over 15 prominent health news sources including BBC, CNN, and NYT. Once the data underwent processing, it was subsequently split into two segments: 70% for training purposes and 30% designated for testing.

#### 4.1.5 *20 newsgroups*

Derived from the raw format available in the UCI data repository [47], this dataset is characterized by balance and comprises 20 substantial classes. The dataset encompasses a total of 20,000 messages sourced from 20 distinct newsgroups. For the purpose of our investigation, we partition this dataset into three subsets: 14,000 messages for training, 2,500 sentences for validation, and 3,500 sentences earmarked for testing.

#### 4.1.6 *Reuters-21578*

Derived from the UCI data repository, this dataset has been utilized in multiple prior experimental research endeavors. Within the Reuters-21578 dataset, 15 classes with imbalanced sizes were employed. We organized the data into two main categories. Firstly, there's "Text classification" (referred to as TextC), encompassing SentiC, CNAE-9, and TextC1, which consists of 20NG and R-21578. Secondly, there's "SemMatch", which includes TE and HNT. Our objective in assessing these two categories is to identify foundational techniques commonly employed in CNNs, RNNs, and DBNs. Table 2 displays the summary of statistics for all the datasets.

**Table 2.** Text datasets description

| Datasets | No. of instances | No. of attributes | No. of Web hits | Area | Associated tasks |
|---|---|---|---|---|---|
| SentiTC | 3,000 | N/A | 100,816 | N/A | Classification |
| CNAE-9 | 1,080 | 857 | 50,866 | Business | Classification |
| TE | 569,028 | 21,000 | 63,121 | N/A | SemMatch |
| HNT | 580,000 | 25,000 | 25,174 | Computer | Classification |
| 20NG | 20,000 | N/A | 80,915 | N/A | Classification/Clustering |
| R-21578 | 21,578 | 05 | 139,119 | N/A | Classification |

### 4.2 *Implementation detail*

To systematically investigate the encoding capabilities of various traditional deep learning approaches, we conducted an experiment using six distinct datasets. The Python 3.6 programming language was employed for data preprocessing and manipulation, utilizing the Sklearn, numpy, and pandas packages. The implementation encompassed both traditional deep learning approaches and GRU networks, with the TensorFlow framework being utilized. TensorFlow is an open-source software library designed for numerical computations utilizing data flow graphs. The experimental design can be summarized as follows:

· Training was consistently initiated from scratch, with no utilization of supplementary information like pre-trained word embeddings.

· Training procedures followed a foundational setup, devoid of intricate techniques like batch normalization.

· Hyperparameters were tailored individually for each task and model, ensuring relevance and appropriateness.

The simulations were executed on a machine powered by an Intel Core i7-3770 CPU, 3.40 GHz, coupled with 8 GB of RAM. Experimental results in terms of accuracy and detailed information regarding all experimental parameters are illustrated in Table 3. Hyperparameters are adjusted for parameters such as hidden size, mini-batch size, learning

rate, maximum sentence length, and the ranking loss in HNT is optimized.

**Table 3.** Experimental results in the term of accuracy and along with experimental parameters

| Tasks | Datasets | Models | Performance | Lr | Hidden | Batch | SentLen |
|---|---|---|---|---|---|---|---|
| Text-C | SentiC (acc) | DBN | 86.44 | 0.2 | 30 | 64 | 60 |
| | | CNN | 86.25 | 0.2 | 30 | 32 | 60 |
| | | GRU | 88.38 | 0.1 | 20 | 64 | 60 |
| | | LSTM | 86.62 | 0.2 | 20 | 64 | 60 |
| | CNAE-9 | DBN | 77.82 | 0.12 | 75 | 32 | 24 |
| | | CNN | 77.67 | 0.12 | 75 | 32 | 24 |
| | | GRU | 78.72 | 0.10 | 70 | 128 | 24 |
| | | LSTM | 77.02 | 0.1 | 70 | 128 | 24 |
| SemMatch | TC (acc) | DBN | 81.15 | 0.1 | 60 | 64 | 55 |
| | | CNN | 80.74 | 0.1 | 60 | 64 | 55 |
| | | GRU | 82.28 | 0.1 | 50 | 32 | 65 |
| | | LSTM | 81.53 | 0.1 | 70 | 32 | 65 |
| | HNT (MAP & MRR) | DBN | 62.52, 63.58 | 0.01 | 40 | 64 | 45 |
| | | CNN | 62.83, 64.32 | 0.01 | 40 | 64 | 45 |
| | | GRU | 61.98, 63.04 | 0.1 | 60 | 128 | 45 |
| | | LSTM | 61.64, 62.90 | 0.1 | 60 | 128 | 40 |
| Text-C1 | 20NG (acc) | DBN | 92.02 | 0.01 | 100 | 40 | 60 |
| | | CNN | 91.48 | 0.01 | 100 | 32 | 60 |
| | | GRU | 92.95 | 0.001 | 80 | 64 | 60 |
| | | LSTM | 92.63 | 0.001 | 80 | 64 | 60 |
| | R-21578 (acc) | DBN | 90.04 | 0.01 | 90 | 50 | 60 |
| | | CNN | 90.47 | 0.01 | 90 | 70 | 60 |
| | | GRU | 91.34 | 0.001 | 100 | 64 | 60 |
| | | LSTM | 91.54 | 0.001 | 100 | 64 | 60 |

## 4.3 *Performance metrics*

### 4.3.1 *Accuracy*

Equation (14) is employed to determine the value of the metric known as accuracy, which represents the proportion of correct predictions made by machine learning algorithms [48] about the overall number of input instances.

$$Accuracy = \frac{TP_{(n)}^{(m)} + TN_{(n)}^{(m)}}{TP_{(n)}^{(m)} + FP_{(n)}^{(m)} + \mathrm{FN}_{(n)}^{(m)} + TN_{(n)}^{(m)}} \tag{14}$$

### 4.3.2 *Mean reciprocal rank*

Mean Reciprocal Rank (MRR) is an evaluation metric commonly used in information retrieval and ranking tasks to assess the effectiveness of a ranking system or search engine. It measures the quality of a ranked list of items by considering the position of the first relevant item within that list. It is defined by equation (15) as follow:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{15}$$

Where, $N$ is the total number of queries in the test dataset, $rank_i$ is the rank of the first relevant item for the $i-th$ query, if no relevant items are found, $rank_i$ is typically set to a predefined maximum rank or considered as infinity.

## 4.4 *Results and analysis*

In this section, we carried out experimental research for text classification tasks using various datasets along with their respective hyperparameters. We assessed the effectiveness of traditional deep learning algorithms in terms of accuracy (Acc) and Mean Reciprocal Rank (MRR). Across different implementation settings and experimental foundations, all models demonstrated strong performance in text classification. However, the GRU model illustrated exceptional results on the SentiC dataset. We compared its performance against baseline deep learning approaches such as DBN, CNN, and LSTM, as illustrated in Table 3 and Figure 8. In the case of textC1, both GRU and LSTM showed better performance as compared to DBN and CNN. Specifically, GRU exhibited superior outcomes on the 20NG dataset, while LSTM illustrated better results on the R-21578 dataset.

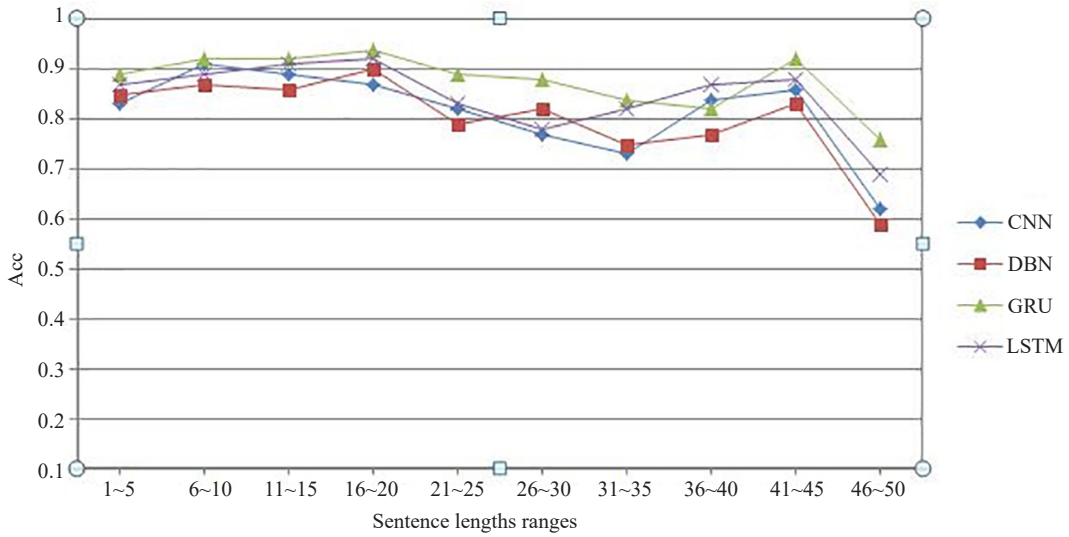


**Figure 8.** Distributions of different sentence lengths ranges and corresponding accuracy

Furthermore, our experiments led us to the conclusion that RNN models such as GRU and LSTM are most effective and well-suited for tasks involving long-range contextual dependencies and text classification. Specifically, when dealing with sentiment matching, unexpected observations came to light. In the domain of local feature extraction and position-invariant attributes, both CNN and DBN were initially deemed superior for capturing such aspects. These models exhibited strong performance on the SentiMatch dataset (referred to as HNT). Surprisingly, in our empirical investigations, RNN outperformed CNN and DBN, particularly in the context of 20NG and SentiC datasets. This superiority of RNNs became evident due to their ability to predict and ultimately generate relational outputs after comprehensively processing entire sentences.

In the subsequent phase, we evaluate the effectiveness of various deep learning approaches, including CNN, DBN, GRU, and LSTM, in terms of maintaining consistent performance across different sets of hyperparameter values. Table 3 illustrates how CNN, DBN, GRU, and LSTM perform under various combinations of learning rates, hidden layer configurations, and batch sizes. Notably, all the deep learning approaches exhibit relatively consistent behavior when confronted with changes in learning rates. Conversely, fluctuations in hidden layer sizes and batch sizes result in significant oscillations. It is worth highlighting that, in the sentiment analysis tasks of SentiTC and TextC, the performance of CNN consistently remains less as compared to DBN, GRU, and LSTM. However, CNN demonstrates superior performance on the HNT dataset for sentiment matching.

## 5. Gaps and discussion

This paper has evaluated a comprehensive assessment of deep learning approaches based on natural language processing tasks such as text classification. In sequence to differentiate and understand the three DL approaches. In this study, we describe their gaps and limitations concerning diverse properties. According to the literature review study, we found that some limitations in the traditional deep learning approaches for text classification tasks. The deficiencies of common neural network architectures, such as their high complexity, extended training periods, and associated implementation expenses. In contrast, conventional deep learning structures have a couple of significant drawbacks, such as the need for extensive computational training time and increased implementation costs due to concerns like overfitting. Moreover, traditional deep learning algorithms often fail to meet expectations as general-purpose solutions, primarily because they demand an extensive amount of training data.

CNN has been applied in various NLP tasks with remarkable achievements. However, their strength lies in hierarchical structures, making them particularly effective for tasks such as pattern recognition, image classification, and object detection. Yet, when it comes to modeling sequential units, CNN is not as suitable. These networks comprise multiple hidden layers to grasp distant relationships within data. Their complexity is evident from the multitude of layers, demanding substantial training data and time. Training a CNN necessitates substantial computational resources, especially a powerful GPU, without which training for intricate tasks can be sluggish.

**Table 4.** Comparative analysis of CNN, DBN, and RNNs models in the terms of strengths, weaknesses, application and performance metrics

| Models | Strengths | Weaknesses | Applications | Performance metrics |
|---|---|---|---|---|
| CNN | It delivers rapid predictions, excels with extensive datasets, and requires no human involvement in feature engineering. It is good at capturing syntactic features and simple patterns in text, such as n-grams and word co-occurrences. | Lack of Sequential Understanding: It is not inherently equipped to capture the sequential nature of text data. While they can capture local patterns within fixed-size windows (n-grams), they may struggle to understand long-range dependencies and relationships between words in a sentence. Computationally expensive requires a large data set for training. | Pattern recognition, image processing, text classification | Accuracy, Precision, Recall, F-measure |
| DBN | It is capable of learning hierarchical representations of data, and capture non-linear relationships in the data. It can perform automatic dimensionality reduction during the feature learning process. | It is computationally expensive, especially for large-scale text datasets, and perform well when you have a large amount of data. DBN has several hyperparameters that need to be carefully tuned. | Feature learning, image recognition, speech recognition, NLP | Accuracy, Precision, Recall, F-measure |
| RNN | It utilizes a feedback model, making it particularly suitable for time series problems and enabling more precise predictions compared to other artificial neural network (ANN) models. | Model training is a challenging and time-consuming process, often requiring significant time to uncover nonlinearity within the data, and it is susceptible to the issue of gradient vanishing. | Sentiment analysis, News classification, question answering, Topic labeling | Accuracy, Precision, Recall, F-measure |
| LSTM | Incorporates both short-term and long-term memory components into RNN, making it particularly well-suited for tasks involving sequential data, such as text classification and text generation in the context of NLP applications, and it operates efficiently with high computational speed. | It utilizes the backpropagation, model adds complexity and cost, elevating the dimensionality of the issue and rendering the search for the optimal solution more challenging. | Sentiment analysis, News classification, question answering, Topic labeling | Accuracy, Precision, Recall, F-measure |
| GRU | It exhibits faster learning and superior performance compared to LSTMs when trained with limited data, and requires fewer training parameters. They offer simplicity, making them more adaptable, without the need for additional memory units like extra gates when the network demands increased input. | It continues to face challenges in terms of sluggish convergence and constrained learning efficiency. It captures short-term dependencies in sequences but may struggle with capturing long-term dependencies effectively | Sentiment analysis, News classification, question answering, Topic labeling | Accuracy, Precision, Recall, F-measure |

RNNs prove effective in modeling sequential elements and are particularly suitable for tasks involving sequences. Opting for RNN is a judicious decision when confronted with sequence-related challenges such as language modeling and document-level sentiment classification. However, the study highlights a notable constraint of RNN, namely, the challenges of gradient vanishing and exploding. These hurdles complicate RNN training in two specific manners: firstly, it struggles to handle extensive sequences when employing the hyperbolic tanh activation function; secondly, adopting the rectified linear unit (ReLU) as the activation function renders the model quite unstable and takes a long time to find nonlinearity. Moreover, in the LSTM approach, the utilization of the backpropagation model adds complexity and cost to the problem, elevating its dimensionality and rendering the search for an optimal solution more challenging. Additionally, the Bi-LSTM, with its dual LSTM cells, further amplifies the implementation costs. On the other hand, slow convergence and limited learning efficiency are still issues in the GRU model. Table 4 illustrates the comparative analysis of deep learning approaches in terms of strengths, weaknesses, application, and performance metrics. Furthermore, the recurrent neural network approaches extremely hard to train and develop relations among two different sentences of tree structure. In this study, it is very important for the reader to note that the gaps which are only represent the general current findings of this study in the deep learning approaches.

## 6. Conclusion and future direction

In this study, we evaluated three various deep learning models for the task of natural language processing such as text classification. A brief overview of findings from existing literature is conducted to analyze the challenges associated with deep learning models. Additionally, we elucidate the architectures of these models and highlight their shortcomings and constraints. Although many achievements have been attained using deep learning models in classification tasks still there are some issues that need to be resolved. Finally, we showed real-world problems like complexity, overfitting and more issues briefly explained in gaps and discussion sessions. In conclusion, deep learning algorithms have demonstrated their effectiveness in decision-making for NLP-based applications, but they face limitations when dealing with symbols directly. Additionally, the training of such algorithms comes with a substantial computational expense.

In the future direction, it offers an opportunity to develop deep neural network architectures capable of incorporating linguistic, lexical, and word knowledge from various domains. We hope that our article will be a useful and fundamental vision for researchers beginning to work on deep learning for text classification.

## Conflict of interests

The authors declare there is no competing financial interest.

## References


[1] Liang M, Niu T. Research on text classification techniques based on improved TF-IDF algorithm and LSTM inputs. *Procedia Computer Science*. 2022; 208: 460-470.

[2] Li Q, Peng H, Li J, Xia C, Yang R, Sun L, et al. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 2022; 13(2): 1-41.

[3] Occhipinti A, Rogers L, Angione C. A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications*. 2022; 201: 117193.

[4] Zulqarnain M, Alsaedi AK, Ghazali R, Ghouse MG, Sharif W, Husaini NA. A comparative analysis on question classification task based on deep learning approaches. *PeerJ Computer Science*. 2021. Available from: https://doi.org/10.7717/peerj-cs.570.

[5] Alqahtani A, Ullah Khan H, Alsubai S, Sha M, Almadhor A, Iqbal T, et al. An efficient approach for textual data classification using deep learning. *Frontiers in Computational Neuroscience*. 2022; 16: 992296.

[6] Palanivinayagam A, El-Bayeh CZ, Damaševičius R. Twenty years of machine-learning-based text classification: A systematic review. *Algorithms*. 2023; 16(5): 236.

[7] Wu J, Yılmaz E, Zhang M, Li H, Tan KC. Deep spiking neural networks for large vocabulary automatic speech

recognition. *Frontiers in Neuroscience*. 2020; 14: 199.
[8] Liu PF, Qiu XF, Huang XJ. Recurrent neural network for text classification with multi-task learning. *Computer Science: Computation and Language*. 2016. Available from: https://doi.org/10.48550/arXiv.1605.05101.
[9] Li Q, Peng H, Li JX, Xia CY, Yang RY, Sun LC, et al. A survey on text classification: From shallow to deep learning. *Computer Science: Computation and Language*. 2020. Available from: https://doi.org/10.48550/arXiv.2008.00364.
[10] Zulqarnain M, Ghazali R, Ghouse MG, Mushtaq MF. Efficient processing of GRU based on word embedding for text classification. *JOIV: International Journal on Informatics Visualization*. 2019; 3(4): 377-383.
[11] Sayed M, Abdelkader H, Khedr AE, Salem R. A proposed deep learning based framework for arabic text classification. *International Journal of Advanced Computer Science and Applications*. 2022; 13(8): 305-313.
[12] Chen H, Wu L, Chen J, Lu W, Ding J. A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*. 2022; 59(2): 102798.
[13] Yogatama D, Dyer C, Ling W, Blunsom P. Generative and discriminative text classification with recurrent neural networks. *Statistics: Machine Learning*. 2017. Available from: https://doi.org/10.48550/arXiv.1703.01898.
[14] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*. 2021; 54(3): 1-40.
[15] Londt T, Gao X, Xue B, Andreae P. Evolving character-level convolutional neural networks for text classification. *Computer Science: Computation and Language*. 2020. Available from: https://doi.org/10.48550/arXiv.2012.02223.
[16] Chung J, Lee J, Yoon J. Understanding music streaming services via text mining of online customer reviews. *Electronic Commerce Research and Applications*. 2022; 53: 101145.
[17] Zhou C, Sun C, Liu Z, Lau F. A C-LSTM neural network for text classification. *Computer Science: Computation and Language*. 2015. Available from: https://doi.org/10.48550/arXiv.1511.08630.
[18] Yu S, Liu D, Zhu W, Zhang Y, Zhao S. Attention-based LSTM, GRU and CNN for short text classification. *Journal of Intelligent & Fuzzy Systems*. 2020; 39(1): 333-340.
[19] Zulqarnain M, Ghazali R, Ghouse MG, Hassim YM, Javid I. Predicting financial prices of stock market using recurrent convolutional neural networks. *International Journal of Intelligent Systems and Applications (IJISA)*. 2020; 12(6): 21-32.
[20] Kowsari K, Brown DE, Heidarysafa M, Meimandi KJ, Gerber MS, Barnes LE. Hdltex: Hierarchical deep learning for text classification. In: *2017 16th IEEE international conference on machine learning and applications (ICMLA).* Cancun, Mexico; 2017. p.364-371.
[21] Zulqarnain M, Ghazali R, Hassim YM, Aamir M. An enhanced gated recurrent unit with auto-encoder for solving text classification problems. *Arabian Journal for Science and Engineering*. 2021; 46(9): 8953-8967.
[22] Sarikaya R, Hinton GE, Deoras A. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2014; 22(4): 778-784.
[23] Al Sbou AM, Hussein A, Talal B, Rashid RA. A survey of arabic text classification models. *International Journal of Electrical and Computer Engineering (IJECE)*. 2018; 8(6): 4352-4355.
[24] Peslak A, Hunsinger S, Kruck S. Text messaging today: A longitudinal study of variables influencing text messaging from 2009 to 2016. *Journal of Information Systems Applied Research*. 2018; 11(3): 25.
[25] Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of tricks for efficient text classification. *Computer Science: Computation and Language*. 2016. Available from: https://doi.org/10.48550/arXiv.1607.01759.
[26] Wang Z, Wu Q. An integrated deep generative model for text classification and generation. *Mathematical Problems in Engineering.* 2018; 2018: 7529286.
[27] Iwasaki M, Kubokawa M, Saitoh T. Two features combination with gated recurrent unit for visual speech recognition. In: *2017 fifteenth IAPR international conference on machine vision applications (MVA).* Nagoya, Japan; 2017. p.326-329.
[28] Alshalif SA, Senan N, Saeed F, Ghaban W, Ibrahim N, Aamir M, et al. Alternative relative discrimination criterion feature ranking technique for text classification. *IEEE Access*. 2023; 11: 71739-71755.
[29] Soni S, Chouhan SS, Rathore SS. TextConvoNet: A convolutional neural network-based architecture for text classification. *Applied Intelligence*. 2023; 53(11): 14249-14268.
[30] Luan Y, Lin S. Research on text classification based on CNN and LSTM. In: *2019 IEEE international conference on artificial intelligence and computer applications (ICAICA).* Dalian, China; 2019. p.352-355.
[31] Jacovi A, Shalom OS, Goldberg Y. Understanding convolutional neural networks for text classification. *Computer Science: Computation and Language*. 2018. Available from: https://doi.org/10.48550/arXiv.1809.08037.
[32] Grossberg S. Recurrent neural networks. *Scholarpedia*. 2013; 8(2): 1888.

[33] Li C, Li H, Zhang G. Future frame prediction based on generative assistant discriminative network for anomaly detection. *Applied Intelligence*. 2023; 53(1): 542-559.
[34] Zulqarnain M, Ghazali R, Hassim YM, Rehan M. Text classification based on gated recurrent unit combines with support vector machine. *International Journal of Electrical and Computer Engineering*. 2020; 10(4): 3734-3742.
[35] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997; 9(8): 1735-1780.
[36] Van Houdt G, Mosquera C, Nápoles G. A review on the long short-term memory model. *Artificial Intelligence Review*. 2020; 53: 5929-5955.
[37] Zulqarnain M, Shah H, Ghazali R, Alqahtani O, Sheikh R, Asadullah M. Attention aware deep learning approaches for an efficient stress classification model. *Brain Sciences*. 2023; 13(7): 994.
[38] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Computer Science: Neural and Evolutionary Computing*. 2014. Available from: https://doi.org/10.48550/arXiv.1412.3555.
[39] Zulqarnain M, Ghazali R, Aamir M, Hassim YM. An efficient two-state GRU based on feature attention mechanism for sentiment analysis. *Multimedia Tools and Applications*. 2022; 1-26.
[40] Mohammed A, Kora R. An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*. 2022; 34(10): 8825-8837.
[41] Zhao H, Yang X, Chen B, Chen H, Deng W. Bearing fault diagnosis using transfer learning and optimized deep belief network. *Measurement Science and Technology*. 2022; 33(6): 065009.
[42] Zulqarnain M, Ghazali R, Khaleefah SH, Rehan A. An improved the performance of GRU model based on batch normalization for sentence classification. *IJCSNS International Journal of Computer Science and Network Security*. 2019; 19(9): 176-186.
[43] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. *Computer Science: Computation and Language*. 2014. Available from: https://doi.org/10.48550/arXiv.1404.2188.
[44] Hendrickx I, Kim SN, Kozareva Z, Nakov P, Séaghdha DO, Padó S, et al. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *Computer Science: Computation and Language*. 2019. Available from: https://doi.org/10.48550/arXiv.1911.10422.
[45] Bowman SR, Angeli G, Potts C, Manning CD. A large annotated corpus for learning natural language inference. *Computer Science: Computation and Language*. 2015. Available from: https://doi.org/10.48550/arXiv.1508.05326.
[46] Nam J, Kim J, Mencía EL, Gurevych I, Fürnkranz J. Large-scale multi-label text classification-revisiting neural networks. In: *Machine Learning and Knowledge Discovery in Databases*. Nancy, France; 2014. p.437-452.
[47] Sharif W, Samsudin NA, Deris MM, Aamir M. Improved relative discriminative criterion features ranking technique for text classification. *International Journal of Artificial Intelligence*. 2017; 15(2): 61-78.
[48] Rayyan A, Aburas MG, Al-Mousa A. Uniform resource locator classification using classical machine learning & deep learning techniques. *Cloud Computing and Data Science*. 2023; 4(1): 17-30.