

---

# A Hybrid Recommendation Integrating Semantic Learner Modelling and Sentiment Multi-Classification

---

Rawaa Alatrash<sup>1,\*</sup>, Rojalina Priyadarshini<sup>1</sup>, Hadi Ezaldeen<sup>1</sup>  
and Akram Alhinnawi<sup>2</sup>

<sup>1</sup>*Department of Computer Science and Engineering, C.V. Raman Global University, Bhubaneswar – 752054, Odisha, India*

<sup>2</sup>*Department Computer Science and Engineering, University of Bridgeport, Bridgeport, CT, USA*

*E-mail: rawaa.alatrash@gmail.com; 20080005@cgu-odisha.ac.in;*

*rojalinapriyadarshini@cgu-odisha.ac.in; hadi.talal@gmail.com;*

*aalhinna@my.bridgport.edu*

*\*Corresponding Author*

Received 24 August 2021; Accepted 22 December 2021;  
Publication 12 April 2022

## Abstract

Enhancing virtual learning platforms need to adapt new intelligent mechanisms so that long-term learner experience can be improved. Sentiment Analysis gives us perception on how a specific scientific material is suitable to be recommended to the learner. It depends on the feedback of a similar learner taking many factors under consideration such as preference, knowledge level, and learning pattern. In this work, a hybrid e-learning recommendation system is proposed based on individualization and Sentiment Analysis. A new approach is provided for modelling the semantic user model based on the generated semantic matrix to capture the learner's preferences based on their selections of interest. The extracted semantic matrix is used for text representation by utilizing ConceptNet knowledge base which relies on

*Journal of Web Engineering, Vol. 21.4, 941–988.*

doi: 10.13052/jwe1540-9589.2141

© 2022 River Publishers

contextual graph and expanded terms to represent the correlation among terms and materials. On the extracted terms from semantic user model, Word Embeddings-Based-Sentiment Analysis (WEBSA) must recommend the learning materials with highest rating to the learners properly. Variant models of (WEBSA) are proposed relying on Natural Language Processing (NLP) to generate effective vocabulary representations along with the use of qualitative customized Convolutional Neural Network (CNN) for sentiment multi-classification tasks. To validate the language model, two datasets are used, a tailored dataset that has been created by scraping reviews of different e-learning resources, and a public dataset. From the experimental results, it has been found that the lowest error rate is achieved with our customized dataset, where the model named CNN-Specific-Task-CBOWBSA outperforms than others with 89.26% accuracy.

**Keywords:** Hybrid recommendation, semantic user modeling, contextual graph, sentiment analysis, word embeddings, deep learning.

## Abbreviation

Word Embeddings-Based-Sentiment Analysis (WEBSA)  
 Natural Language Processing (NLP)  
 Convolutional Neural Network (CNN)  
 Amazon Book Hybrid Recommendation dataset (ABHR-2)  
 CNN variant models (CNNMs)  
 Contextual Graph ( $CG$ )  
 Semantic matrix ( $S$ )  
 Material's Concepts (MC)  
 Intersection Set (IS)  
 User Model ( $U_t$ )  
 Skip-Gram (SG)  
 Continuous Bag of Word (CBOW)  
 Skip Gram Based Sentiment Analysis (SGBSA)  
 Continuous Bag of Words Based Sentiment Analysis (CBOWBSA)

## 1 Introduction

Learning processes need to be fast and just-in-time for a satisfactory user's experience. Speed is required to process a large number of learners' data

to select suitable content of the learning material “highly specified, not too general” [1]. Also, a powerful intelligent mechanism needs to be executed for automatic linking such material with the learner’s scientific field within a reasonable time. It contributes to solve the problem of losing time for finding the individual needs of materials from the huge contents inside e-learning environments [2].

In the existing learning systems, learners spend a lot of time and effort trying to find relevant and interesting learning resources. It becomes vital to adopt an effective recommendation system initiated by learner modelling to achieve more efficient learning experience for learners [45]. Most of the existing systems are not built adaptively and their associated components are not presented correctly to the learner. Thus, more intelligent mechanisms and hybrid recommendations are required in an e-learning platform. Hence, it is necessary to consider the users’ role to understand their perceptions and tolerance levels of the usability of the resources in such a way that would enable extracting their sentiments online.

The current work proposes a hybrid recommendation system initiated by user model and combined with Sentiment Analysis model for personalized e-learning recommendations. A new method is proposed to develop the semantic user model automatically where the user interests are represented as a dense vector of relative relations. For formulating this, e-content of scientific materials associated to different primitive terms have been collected. Textual data for each material is represented as a sequence of concepts using ConceptNet English part. Based on these extracted concepts the contextual graph of materials has been built. An expansion on the primitive terms has been applied based on ConceptNet concepts by taking into account specific edges. The edge weight is taken from ConceptNet where there is an edge link between the term and the set of expanded concepts based on a specific relation type. Then, an intersection between two sets has been considered; the set of the material’s concepts and the set of primitive terms with their expanded concepts to obtain the concepts that belong to both sets. The semantic relation matrix has been built based on materials and primitive terms leveraging association rules and weights of relation types captured by ConceptNet. It gives the ratio of correlation materials to terms. Based on the user selections of materials and semantic relation matrix, the semantic user model is constructed. The system WEBSA is evaluated over thirteen suggested models where different language models are integrated with variant CNN models. Word Embeddings techniques; Continuous bag of words (CBOW) and Skip-Gram (SG) in Natural Language Processing

(NLP), is used to generate effective vocabulary representations, along with the use of qualitative customized Convolutional Neural Network (CNN) for sentiment multi-classification tasks. A tailored-domain dataset was created by scraping reviews of different e-learning resources from Amazon.com. Based on the extracted learner's preferences and findings of Sentiment Analysis based models, the system is able to recommend relevant e-learning resources to the learner properly. The acquired results are better than the existing methods.

In the current work, the major contributions are enlisted below:

- Proposed a new framework of hybrid e-learning recommendation System based on individualization and Sentiment Analysis, where e-learning resources with higher ratings are to be recommended according to the semantic user model.
- The enhancement is done on the semantic matrix based on contextual graph and expanded terms using ConceptNet, that is representing the correlation between the terms and materials.
- Semantic user modelling has been refined relying on the extracted semantic matrix which takes care of the learner's interests, which are embedded to reinforce the proposed recommender engine.
- The recommender system (WEBSA) is enriched with two enhanced language models based on word embeddings techniques to get better word representations, leveraging the goodness of diverse CNN constructions which predicted the accuracy as 89.26% for CNN-Specific-Task-CBOWBSA. In turn, the system provides personalized recommendations in terms of fuzzified reviews which are categorized into five classes unlike polarity classification.
- A customized dataset has been built named "Amazon Book Hybrid Recommendation dataset (ABHR-2)" that has been collected for the Education domain along with thirteen CNN variant models (CNNMs) have been trained and evaluated on ABHR-2 and compared with different public datasets of precomputed vectors along with textual public dataset 'SST-1'.

Proposed models are evaluated by different scenarios which resulted in the best performance as concluded in Table 1.

Section 2 depicts the related work. The methodology and the proposed methods of WEBSA have been outlined and presented in Section 3. Section 4 describes the experimental work. The results are discussed in Section 5. Summarising the work is illustrated in Section 6.

## **2 Literature Study**

### **2.1 User Modelling Methods**

Developing the user model provides a way to understand the user's interests in an implicit way. It helps to express preferences between different alternatives within the e-content groups, depending on the problematic decision for addressing personality behavior patterns in counseling learners [49]. Several user models have been developed in recommendation systems based on various multi-criteria methodologies. These models can be classified into various classifications depending on the format of the model used and the procedure for creating it. The user modelling can be categorized into two methods; (A) "statistical method-based user model" [3] and (B) "semantic method-based user modelling" [4].

Many researchers conducted their work in the field of searching and recommending suitable content in e-learning management system, and many others tried to improve general purpose Information Retrieval (IR) systems based on statistical methods like; Term Frequency–Inverse Document Frequency (TF-IDF) [3], Latent Dirichlet Allocation (LDA) [5], sentimental features [6], and Hashtag frequency [7].

On another context, by use of semantic methods we can help to improve the recommendation results to the users. This aids to find suitable document regardless the literal identification of contents. The semantic relations between terms and content are deduced utilizing DBpedia and WordNet Synsets in [8] by applying a generative ML-based model, the user model is to be a vector of relative relation to a set of e-learning categories to provide personalized e-learning content.

An entity-based semantic user model conducted by [9] to extract the entities from users' posts and the news articles, where these posts and the mainstream articles have been linked to enrich the text on the social web.

The authors in [10] proposed generic social media model of recommendations based on geotagging. The user's profile has been reinforced with a deeper embedding of semantics obtained from DBpedia to match user's interests with items. Also, the semantic relations were employed to expand the user model in order to fit the recommendations with a user's needs model [11].

Automated methods are used by ConceptNet to extract concepts and relationships from many renewable sources like DBpedia, WordNet, OpenCyc, and Open-source common sense, while preserving the original source of each relationship within it. Reliance on these renewable knowledge resources

permits an increase in the number of concepts and relationships within ConceptNet continuously [12]. Each relationship between two concepts can be represented in the form of tuple (concept1, relation, concept2), and it is considered one of the richest bases of knowledge in semantic terms.

It is obvious that the extraction of relations between the e-content and the concepts connected to the particular material needs some deep semantic analysis. Moreover, the relation type between these concepts is considered. That is reflected in this work through building the semantic user model based on the semantic relations of the contextual graph that is captured from a rich network like ConceptNet [13].

## **2.2 Importance of Sentiment Discovery and its Presence on E-learning Platforms**

Sentimental factors have a critical role in adaptive instruction in emotional aspects [14]. Learners' sentiment and subjectivity towards study resources, methodological and extracurricular activities are a vital part of the e-learning trends. Learners should be guided through the learning process according to their particular needs like specific courses or any specialized scientific materials. That can be done in many ways, such as using sentiment analysis technologies, which help in delivering immersive learning experiences to learners [15].

[16] presents the user attitudes toward themselves, others, learning resources, and online courses affect their ability to achieve educational goals.

Some studies make use of NLP techniques for detecting the emotion and its impact on the motivation and academic students' results [50, 51].

Cobos et al. [17] have developed a content analyzer system (edX-CAS) based on NLP techniques to detect the sentiments reflected through online learning courses in order to detect the subjectivity towards courses for the appropriate recommendations. Whereby using several classifiers in parallel with an evolutionary approach named (EvoMSA) were developed for Sentiment Analysis [18].

Opinion mining and emotion recognition were applied to an intelligent learning environment [19], which leads to an effective teaching-learning process. Emotional attitudes also affect students' abilities by affecting the ongoing changes of the interactive e-learning environment. Moreover, the education with a real motivation leading to form appropriate psychological trends which are more feasible than education relies on the acquisition of knowledge only [20].

Salazar et al., [21] conducted a literature study on the impact of recommender systems in capturing the influence and role of emotions, as well as the necessity to adopt hybridization recommendations to recommend learning content within learning platforms.

### **2.3 Technical Directions of Sentiment Analysis**

Using CNN for tasks of sentence classification [22–25, 46] is regarded as a perfect algorithm that is achieved great findings in this field. Initially, it is invented for computer vision scope. After that, it takes a way on the techniques such as search query retrieval [26], sentence modeling [27], and semantic parsing [28].

The first pioneer workers who employed CNN for various NLP tasks were Collobert and Kim [22, 29]. Wherein all these approaches and techniques are combined and integrated to translate them into a high-dimensional space. The research work [30] proposed several models that have been used to represent the paragraph “Paragraph vector” (PV). Product embedding is given in the penultimate layer of recurrent neural network (RNN) with gated recurrent unit (GRU). Both feature vectors and PV are fed to the classifier Support Vector Machine (SVM) to predict the new rating. Web service used to predict the mismatching between the predicted rating and the actual rating and send feedback to the reviewer.

The authors in [31] presented different algorithms by using Decision Tree and SVM to classify book reviews, comparing the rating of different books based on polarity, and eight basic emotions.

The conducted work in [32] utilized Common features existed in bag-of-words or n-grams, and their TF-IDF and then compare it with CNNs.

It is extensively explored for mining and summarizing opinions of social media on the Web, which is full of positive and negative people’s opinions, sentiments, thoughts, and behaviours, in order to leverage them later in the field of education [33].

To extract features, deep convolutional networks have been effectively deployed. Many word vector learning models have been reported, including word2vec [35, 36], FastText [42], and GloVe [48] as well as utilizing transformer-based pre-trained models for learning the sentence embeddings such as DistilBERT [52].

Further, a study for Sentiment Analysis to extract features based on creating ontologies accompanied by Word2vec and CNN has been presented by [34].

Customized e-learning actually needs to expand the potentials of the existing models through hybridization methods. It contributes to having knowledge of the potential user's interests and sharing the Sentiment Analysis-based knowledge from other learner similar in their knowledge level and learning pattern.

Deep Learning has recently been applied in various fields, including computer vision, big data [54], and fog computing [53, 55] and NLP, as a consequence of its excellent performance in Deep neural language networks that have effectively utilized to extract features.

### 2.3.1 Word embedding methods

Word Embeddings can be utilized to obtain dense word vectors, where more information is translated and mapped from high-dimensional data into a lower-dimensional vector space. It infers the semantic relationship between language words through the words and their neighbors in the dimension space.

**Word2Vector:** premises on the distribution hypothesis developed by [35] to represent the words in a massive corpus. It associates each word with a continuous vector in N-dimension space based on the space it appears in, to be then utilized for purpose of similarity. It employs the probabilistic prediction method to deduce syntactic and semantic information of the language words to determine the relationship among them. It includes two techniques Skip-Gram (SG) and Continuous Back of Word (CBOW).

- **SG:** is a kind of unsupervised learning because the word embeddings are not labeled as it used the surrounding words for the prediction [36]. SG can predict the neighboring words based on the current word. A certain word can be fed into Neural Network, while the output is the probability of each word of the vocabulary given a certain word. Word Embeddings can be acquired through the weights of the hidden layer. SG aims to maximize the predicting probability of context words to be from the target word context for the dataset pairs.
- **CBOW:** utilizes the surrounding words to predict the target word. Its input is a group of context words while the output is only one word, that is, the target word [36]. The sack of CBOW is to maximize the predicting probability of the target word to be the context words target for the entire training pairs of the dataset, which the motivation has been derived from. All of Mikolov [35], Kim [22] and Kumar [34] have conducted various CNN modules using Word Embeddings techniques,



some of them have been employed for sentiment multi classification that our research objects have been focused on.

There are no notable works applying fine-grained sentiment analysis in recommendation systems, which might be a critical component in academic sentiment analysis. The majority of studies focus on polarity categorization in sentiment analysis.

As a result, fine-grained sentiment classification becomes challenging to manage in many circumstances for meaningful recommendations on the web. Learners must have a deeper insights comprehension of the educational resources accessible online, which involves the multi-classification of education materials. Thus, a robust recommender system is required to deliver e-content recommendations in the shape of fuzzified evaluations that are categorized into multiple classes demonstrated in the existing work, as contrary to polarity classification. The outcomes of this study can benefit learners by positively influencing their thoughts and attitudes that are enriched by an appropriate recommendation process. Furthermore, the majority of researchers used a generic dataset that was generated from broad terms and common purposes, which did not lead to advances in the language model that would be used in the e-learning recommendation process. These studies are aimed at generating practical recommendations for applying the existing system of learner individualized recommendations in an e-learning environment. Thus, our goal is to maximize the utility of customized CNN structures for the challenge of fine-grained sentiment classification. That highlights the primary goal of categorizing e-learning materials into fuzzy sets of 5 groups, which is one of the major issues of current studies, rather than the Sentiment Analysis polarity. The findings of our research in this study are based on predicting the ratings of learning material to be used in the recommendation process of our proposed model.

### **3 The Proposed Framework of the Hybrid Recommender System**

The proposed hybrid recommender system is relying on three main models; Semantic User Model, WEBSA and Recommendation Process that are integrated for the recommendation methodology. The system relies on inferring the user model to extract the user preferences based on semantic user modelling that is depicted in Section 3.1. According to user interests, the system predicts the new rating of the reviews belonging to materials using WEBSA

combined with the enhanced language models as described in Section 3.2, in order to recommend relevant e-learning resources with high ratings to the learner.

### 3.1 Semantic and Context-based User Modelling

Building the user model is initiated by semantic-based representation of the textual e-content. Therefore, e-content of scientific materials were collected from several primitive terms as mentioned in Section 3.2.4, wherein the e-content is to be represented as a series of concepts.

The material concepts have been extracted for each scientific e-content to build the contextual graph (*CG*). This is done by applying various procedures to the text material such as sentencizing the text, eliminating punctuations, and removing digits then extracting the nouns existed in material's text to form the material's concepts. After that, semantic relationships between these concepts are captured as one concept appears in the context of another concept. *CG* of each material is represented as a graph of contextually related concepts based on the contextual relations (edges) that link the material's concepts with each other. This is modeled properly in the Pseudo code for extracting material concepts and building the materials graph.

The input is a set of consequent words from the text material and the output is the material's concepts that form the graph, as follows:

$$MC = \{c_1, c_2, \dots, c_n\} \quad (1)$$

To enrich the semantic linking between the primitive terms and materials, it is suggested to expand the primitive terms. For each term, a set of concepts that are semantically connected to this term, are captured from ConceptNet knowledge graph [13].

ConceptNet includes more semantic information and knowledge about the entities. ConceptNet with version 5.7 can be accessed by REST API. Each term is connected to various concepts based on diverse relations exist in ConceptNet. Only few types of relations/edges can be considered, which are; "Derived\_From, Has\_Context, Related\_To, Is\_A, Part\_Of, Member\_Of, Has\_A, Used\_For, Capable\_Of, At\_Location, Synonym, Causes or Defined\_As".

Each two concepts are linked to each other with an edge that has a weight. The weight of the concept is determined by the weight of the edge between the term  $t$  and the concept  $c_i$  connected to this term. The term is linked to multiple concepts based on above relation types, as well as there may be

multiple edges linked between the two concepts. In this direction, we have taken all the concepts related to this term into account based on mentioned relation types.

In this suggested expansion method, the input is the primitive terms and the output is the term with the expanded concepts captured from ConceptNet that is a set of concepts related to this term, as formulized in (2):

$$C_t = \{c_i: \exists e_{(t,c_i)}\} \quad (2)$$

Where  $e$  is an edge represents the relation type in ConceptNet. Then, an intersection is considered between the set of material's concepts ( $MC$ ) and  $C_t$ , in order to find the set of concepts that are related to both  $MC$  and  $C_t$ , as following:

$$IS = \{MC \cap C_t\} \quad (3)$$

Thus, a value of  $W_{e_{(c_i,t)}}$  is defined as the weight of the edge that represents a semantic relation type linked between the term  $t$  and expanded concepts  $\{c_i = c_1, c_2, c_3 \dots c_i\}$  taken from ConceptNet. The edge values of semantic relations in ConceptNet have a maximum value of 30, so to adjust them between 0 and 1, we divide by thirty. The cube power is observed experimentally as the basis and explanation are shown in [37].

So that, we take the summation of normalized  $W_{e_{(c_i,t)}}$  for all  $c_i$  included in  $IS$  as follow:

$$w_c = \sum_{c_i \in IS} \left( \frac{W_{e_{(c_i,t)}}}{30} \right)^3 \quad (4)$$

If  $t$  is included in  $MC$ , thus  $t$  has contextual relation with some concepts from  $MC$  like  $c_1$ . That means  $t$  appears in the context of  $c_1$  where there is an edge ( $e$ ) link between ( $t$  and  $c_1$ ) in the  $CG$ . We find out the semantic relation between ( $t$  and  $c_1$ ) in ConceptNet based on the semantic relations mentioned above. Then we take the weight value of this relation type corresponding to the edge  $e_{(t,C_1)}$  from ConceptNet. This is repeated for all the concepts  $c_n$   $\{c_n = c_1, c_2, c_3 \dots c_n\}$  that have a contextual relation with this term in  $CG$ .

The weight of  $t$  ( $W_t$ ) for all concepts  $c_n$  would be defined by taking the summation of all edge weights as in (5):

$$W_t = \sum_{c_n \in MC} \left( \frac{W_{e_{(t,C_n)}}}{30} \right)^3 \quad (5)$$

This procedure is repeated for all the primitive terms.

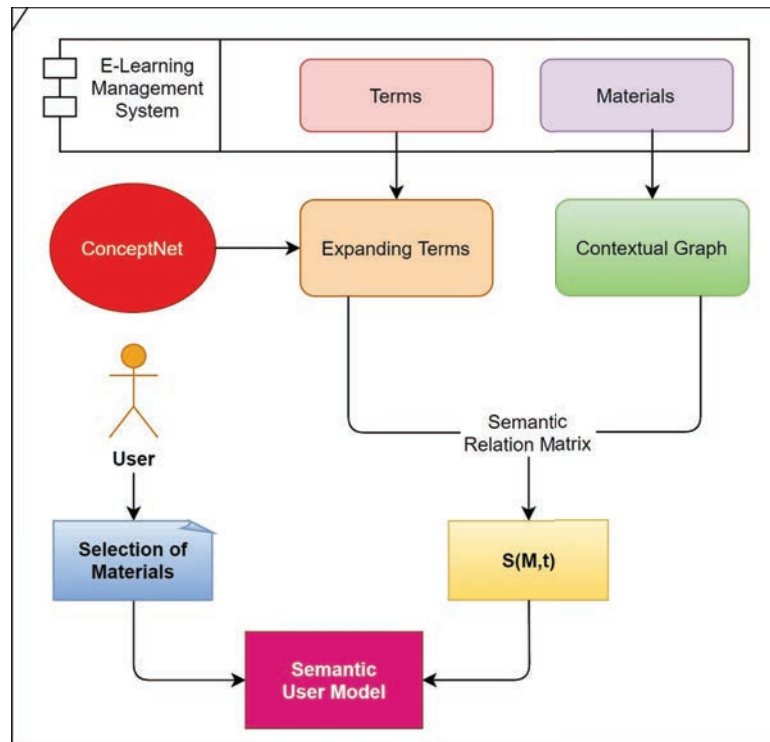
A semantic matrix  $S[M * T]$  is provided, determine each relative value  $S(m, t)$  of semantic relations between the primitive terms  $T = \{t_1, t_2, \dots, t_i\}$  and the materials  $M = \{m_1, m_2, \dots, m_n\}$ . That results in the relation between term and material, which can be calculated as in formula (6):

$$S(m, t) = W_c + W_t \quad (6)$$

Each  $S(m, t)$  value corresponding to the relative semantic relation for  $t$  with respect to  $m$ . The semantic matrix of values  $S(m, t)$  that link terms to materials is built and utilized to deduce the user model.

### 3.1.1 Expanded conceptual learner model

User model (Figure 1) is a vector of a set of user terms that can be determined based on the learner's behavior. If the user selected one material, then from  $S$  matrix, we take the term with the highest value that link between this material



**Figure 1** Architecture of deducing the semantic user model.

$m$  and the term  $t$ , where the user model is defined as follow:

$$U_t = \{(t, \max S(m, t), t \in T\} \quad (7)$$

where  $m \in M$  set of materials selected by the user.  $T$  is set of terms where  $t$  is the  $i$ th term.

The user selects multiple materials ( $n$ ) then we take the terms with the highest values that link between these selected materials and all terms as given below in (8):

$$U_t = \{(t, \max S(m_{\{n\}}, t_i), t \in T\} \quad (8)$$

Where  $n \in [1, n]$  refers to the materials selected by the user.

That results in learner model is represented as a vector of weighted terms sorted with respect to the values of  $\max S(m_n, t_i)$ , as it is formulated in (9):

$$U_t = \{(t, \text{Sort}(\max S(m_n, t_i)), t \in T\} \quad (9)$$

Based on the current or new user's selections of materials, the user's interests are deduced and represented as a dense vector of weighted terms to form the user model.

For the recommendation process, we regard the first top 5 terms that have the highest values  $\max S(m_n, t_i)$  and represent the top individual interests deduced by the proposed semantic user model.

Depending on those first top 5 terms inferred by the semantic user model, the recommender system recommends the e-learning resources that obtained the highest ratings to the learners, through the recommendation process based on Sentiment Analysis model, as mentioned in Section 3.2.3.

For example: given a semantic matrix  $S(M, T)$  in Table 3.1 that link between terms and materials

	Terms $S(M, T)$					
Materials	$t_1$	$t_2$	$t_3$	$t_4$	$\dots$	$t_{40}$
$m_1$	0.714	0.428	0.389	0.637	$\dots$	0.566
$m_2$	0.517	0.556	0.377		$\dots$	0.732
$m_3$	0.124	0.817	0.424	0.147	$\dots$	0.248
$\dots$	$\dots$	$\dots$	$\dots$		$\dots$	$\dots$
$m_n$	0.168	0.318	0.147		$\dots$	0.204

The interests of a learner  $u$  are represented as a set of weighted values corresponding to a series of concepts; each value measures the learner's interest in a specific concept, so let  $U_t$  be a set of possible concepts  $t_m$  of the learner model as follows:

$$U_t = \{t_1, t_2, \dots, t_i\}$$

Such as “deep\_learning” and “semantic\_analysis” would be represented by  $t_1, t_2$  respectively, and so on.

Thus, the learner model of a learner  $u$  is considered as a set of the weighted values as follows:

$$W_u = \{w_{t_1}, w_{t_2}, \dots, w_{t_{40}}\}$$

Where,  $w_{t_i}$  signifies the certitude to which the learner model is related to the term  $t_i$  with this given weighted value.

To deduce  $w_{t_i}$  values; suppose a learner  $u$  selected  $m_1$  and  $m_3$ , then from  $S(M, T)$  we find  $w_{t_1}$ :

$$\left. \begin{array}{l} S(m_1, t_1) = 0.714 \\ S(m_3, t_1) = 0.124 \end{array} \right\} \xrightarrow{S(m_1, t_1) > S(m_3, t_1)} w_{t_1} = 0.714$$

Then for  $w_{t_2}$ :

$$\left. \begin{array}{l} S(m_1, t_2) = 0.428 \\ S(m_3, t_2) = 0.817 \end{array} \right\} \xrightarrow{S(m_1, t_2) < S(m_3, t_2)} w_{t_2} = 0.817$$

And so on to find the remaining  $w_{t_i}$  values for all terms  $3, 4, 5, \dots, i$  of the user model, which would then have the following weighted values of the terms given below:

$w_{t_1}$	$w_{t_2}$	$w_{t_3}$	$w_{t_4}$	$\dots w_{t_m}$	$\dots w_{t_{40}}$
<b>0.714</b>	0.817	0.424	0.637	$< 0.424$	0.566

Then the recommendation process relies on the set of the Top-5 learner's interests (preferences) ( $P_u$ ) that should be selected as the following order;

$$P_u = \{w_{t_2}, w_{t_1}, w_{t_4}, w_{t_{40}}, w_{t_3}\}$$

Which is to be with the values given below:

$w_{t_2}$	$w_{t_1}$	$w_{t_4}$	$w_{t_{40}}$	$w_{t_3}$
<b>0.817</b>	0.714	0.637	0.566	0.424

### **3.2 Proposed Model for Sentiment Analysis WEBSA**

This explains the structure of WEBSA that comprises of language model (Subsection 3.2.1) integrated with CNN model, as well as the system procedure of training and recommendation process as illustrated in Subsections 3.2.2 and 3.2.3 respectively. The Dataset, corpus pre-processing and representation depicted in Sections 3.2.4 and 3.2.5.

In addition, the Subsection 3.2.6 also explains the CNN model's construction and rest of the details are described in Section 3.2.6.

#### **3.2.1 Proposed methods of language model**

Two suggested language models are presented in; Skip Gram Based Sentiment Analysis (SGBSA) and Continuous Bag of Words Based Sentiment Analysis (CBOWBSA). Each of which implemented using Word Embeddings techniques, included in NLP based on the entries of information text. The output of each is being the input for CNNs separately that form the basic step to boost the recommendation process.

##### **3.2.1.1 SGBSA**

SGBSA uses SG technique to address the problem of Word Embeddings in order to mutate the corpus into vectors. It utilizes the target word to predict the surrounding words. For the rare words and phrases, it gives better representation than frequent ones. The text reviews of our dataset became the input of this method, whereas the output vectors, that are so-called "pre-computed vectors", forming the input of CNNs.

##### **3.2.1.2 CBOWBSA**

CBOWBSA, as the name suggests, can address the problem of extracting the vectors out of the corpus by using the CBOW technique. It takes the surrounding words to predict the target words. Performing well for representing frequent words while less for rare ones. Its input is the vocabulary of ABHR-2 and its output which are called "pre-trained vectors" shaping the entries of CNNs

#### **3.2.2 Training process**

In order to train the CNN models, the text reviews of ABHR-2 mentioned in Subsection 3.2.4 have been used. Wherein each is labeled by using a rating score. These rating scores are captured by the learner that corresponds to a specific learning resource. The word embeddings extracted from the text

reviews of the existing learning resources are fed to CNNs models where the rating score is the target. The regularization is conducting on the output by utilizing L2-norm [38, 39] and Dropout technique [40].

### 3.2.3 Recommendation process

The recommendation process is based on incorporating the learner's preferences that are extracted from Semantic User Model depicted in Subsection 3.1.1 as well as employing CNN for turning the text information to features which are extracted from the input text reviews of the e-content with the corresponding rating records marked by the learners.

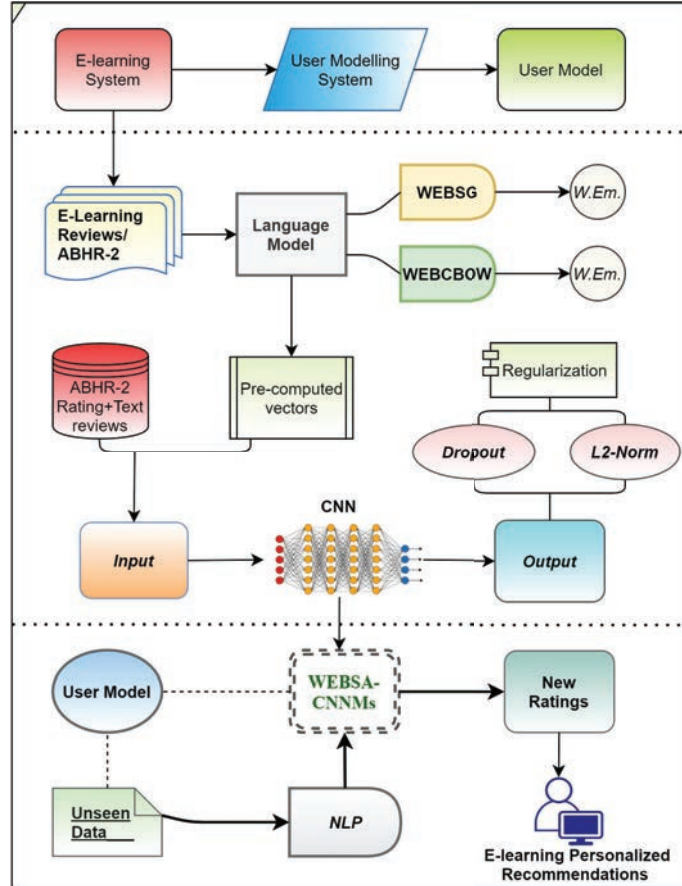
The recommendation process is based on the predicted ratings of e-learning object based on the reviews of similar learners and their knowledge level, where it is resourced by the learner's preferences as shown in Figure 2.

The predicted rating score is used to classify the text reviews acquired from e-content, which will denote whether it is fulfilling a good rating to be used in predicting well-learning resources to the learner or not.

- First, for the current or new specific learning recourse, the new ratings of each learning object are extracted using CNN models.
- Second, in order to get the final predicted rating of the e-content that is related to the scope of learner needs, the mean is applied to the entire predicted ratings that have been extracted from the text reviews regarding this specific learning resource.
- After that, top  $k$  learning resources that achieved the high ratings (here  $k = 2$ ) are to be selected for each term in the list of learner terms that was extracted from the semantic user model, which form 10 learning resources for all user's terms.
- Ultimately, the learning resources with high ratings out of all the learning resources and related to the user preferences is recommended to the learner.
- When a new data is admitted, the model must predict the books with the highest rating to the learner relying on his/her interests.

Thus, this proposed recommendation algorithm could be used in order to evolve the existing recommender approach within the e-learning domain. Moreover, it can be utilized as a new recommendation system.





**Figure 2** Architecture of the proposed recommendation system.

### 3.2.4 Data sets

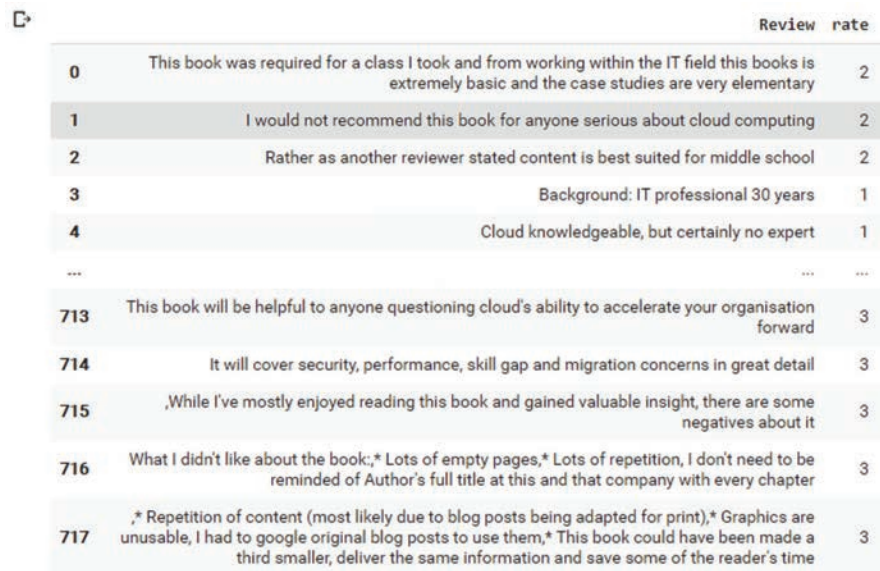
Textual data were collected from the e-content of 120 materials that belong to 50 primitive terms such as {‘Image Processing’, ‘Deep learning’, ‘Cyber Security’, etc.}, based on recommendation of 75 experts in our university. These gathered materials and the linked term reflect the actual learner’s needs in real classes. The collected data is used to investigate the proposed approach of building the Semantic User Model and for applying the semantic text analysis to generate the semantic matrix as mentioned in Section 3.1.

Our own dataset named ABHR-2 [41] has been created based on primitive terms by scraping different pages of Amazon.com in order to get books

reviews by utilizing Python Script. The dataset includes around 80000 records that comprise customers' reviews for books with its corresponding rating wherein, these corresponding ratings have taken an encoding from '1' to '5' where each value is mapped from very negative to very positive.

For example: the review<sup>1</sup> text is "[‘particularly’, ‘strangely’, ‘present’, ‘math’, ‘machine’, ‘learning’, ‘dumbed’, ‘way’, ‘using’]", whereas the rating corresponds to this sentence is '2' as given in ABHR-2. In the above example, the review<sup>1</sup> text represents a sentence in ABHR-2 with its corresponding rating.

Thus, resulted in ABHR-2 consists of various sentences/users' reviews with their corresponding rating out of 5, as it is presented in the Figure 3.



	Review	rate
0	This book was required for a class I took and from working within the IT field this books is extremely basic and the case studies are very elementary	2
1	I would not recommend this book for anyone serious about cloud computing	2
2	Rather as another reviewer stated content is best suited for middle school	2
3	Background: IT professional 30 years	1
4	Cloud knowledgeable, but certainly no expert	1
...	...	...
713	This book will be helpful to anyone questioning cloud's ability to accelerate your organisation forward	3
714	It will cover security, performance, skill gap and migration concerns in great detail	3
715	,While I've mostly enjoyed reading this book and gained valuable insight, there are some negatives about it	3
716	What I didn't like about the book,* Lots of empty pages,* Lots of repetition, I don't need to be reminded of Author's full title at this and that company with every chapter	3
717	,* Repetition of content (most likely due to blog posts being adapted for print),* Graphics are unusable, I had to google original blog posts to use them,* This book could have been made a third smaller, deliver the same information and save some of the reader's time	3

**Figure 3** Sample of ABHR-2.

**FastText** vectors dataset [42] that contain pre-trained word vectors is publicly available and trained on Common Crawl and Wikipedia based on character or n-gram-level, which are trained for 157 languages. For our purpose, we used the English part. It is used to extract vectors out of the corpus.

<sup>1</sup>An example sentence taken from our Dataset, corresponded to text information of the review.

**GloVe** is an unsupervised algorithm that generates to represent words as vectors [48]. The representations are trained using the statistics of the aggregated global word-word co-occurrence information from a text, and the resultant representations highlight intriguing linear substructures in the space of the word vector.

**DistilBERT** is a BERT-based Transformer model that is compact, quick, inexpensive, and light [52]. During the pre-training process, knowledge distillation is used to minimize the dimension of a BERT model. It is a smaller neural network that approximates Google’s BERT.

### 3.2.5 Pre-processing procedure for the text information

Varied pre-processing tasks have been implemented on the ABHR-2 such as clean it from noise, Tokenize Vocabulary out of corpus and Vectorize the sentences into vectors.

Each vocabulary has been matched to a dictionary value that maps to an integer while the key of the dictionary is the vocabulary themselves.

Since CNN accepts the sentences with fixed length, these vectors are then padded with zero value when it's necessary. The generated vectors are used as an input to CNNMs.

For example:

The Embeddings vector of the previous review<sup>1</sup> is as follows:

[illegible]

The rate corresponds to review<sup>1</sup> is form of one-hot vector as below:

	The corresponding rating				
sentence <sup>1</sup> index in Dataset	1	2	3	4	5
<u>345</u>	0	1	0	0	0

### 3.2.6 Construction of CNN based model

Diverse architectures of CNN have been employed in order to obtain a significant recommendation system as mentioned in Section 3.2.3. A seven-layer-based CNN have been considered in order to build the models.

Consider a series of words  $x_i \in \mathfrak{R}^k$  that forming a sentence  $S$  where,  $x_i$  is the  $i$ th word in  $S$  of length  $n$ . Where  $S$  is represented by  $n \times k$  matrix, and the  $i$ th row of the Sentence's matrix corresponding to  $k$ -dimensional word vector. Then the concatenated vector is represented by

$$x_{1:n} = [x_1, x_2, \dots, x_n], \quad \text{Where } x \in \mathfrak{R}^{nk} \quad (10)$$

In order to generate a new feature, an operation of convolving a filter  $w \in \Re^{v \times k}$  over window that comprises of  $v$  words are applied, whereby a convolution process is engaged in every convolutional layer.

Let  $c_i$  is a feature extracted from a kernel of words  $x_{i:i+v-1}$  as given:

$$C_i = \mathcal{G}(w.x_{i:i+v-1} + b) \quad (11)$$

Where,  $b \in \Re$  indicates to bias expression, and  $\mathcal{G}$  points the ‘ReLU function’ is non-linear activation function, this filter convolves over every prospect window of words  $\{x_{1:v}, x_{2:v+1}, \dots, x_{n-v+1:n}\}$  in order to create feature maps:

$$C = [C_1, C_2, \dots, C_{n-v+1}], \quad \text{where } C \in \Re^{n-v+1} \quad (12)$$

Then it followed by a max-over-time pooling scheme to take the summit value  $C' = \text{Max}\{C\}$ , where each corresponds to a specific filter. This concept is to capture the feature with the most significant value. The former procedure is used to extract one feature from one filter, while the model is using various filters with different window sizes to get several features.

The convolution process results in the vector:

$$z = [C'_1, C'_2, \dots, C'_j], \quad \text{where } j \text{ is the filter numbers.} \quad (13)$$

These obtained features ( $z$ ) forming the penultimate layer, that will constitute the input for a fully connected Softmax layer, wherein its output is being the probability of dispensing classes.

The output units categorized into five classes. In order to attain further tune of CNN parameters, the Back-propagation algorithm utilized, by the gradient of loss function “categorical\_crossentropy”. L2-norm and Dropout technique are utilized to address the problem of overfitting on the penultimate layer, by dropping out ‘p’ a ratio of hidden units randomly during forward-propagation at training time.

Usually, the forward-propagation in the output unit is computed by

$$y = w * z + b, \quad (14)$$

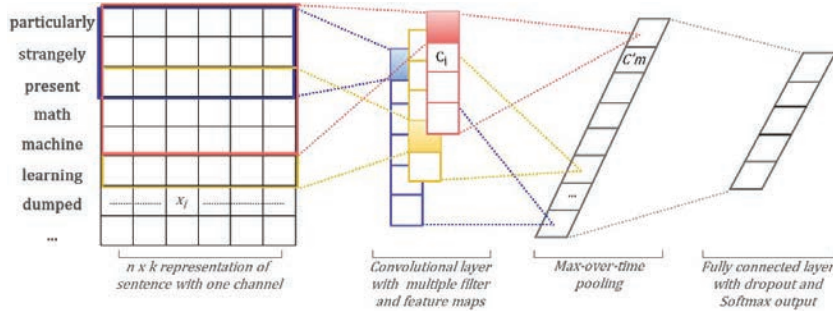
wherein, the  $z$  is the vector forming the penultimate layer. While with using dropout technique it becomes;

$$y = w * (z \oplus r) + b, \quad (15)$$

where  $\oplus$  refers to “Element-wise multiplication operator” is applied on the output.  $r \in R^j$  refers to Bernoulli random variables is ‘masking’ vector with probability of being 1, and the “dropout ratio” 0.5. has been used.

The weighted vectors that learned are rescaled by using  $p$  to be  $w = pw$  then  $w$  is utilized (without dropout) to appraise unseen data during testing time. Moreover, the weight vectors are bounded using  $l_2$ -norms and remeasured to have  $\|w\|_2 = s$  whenever  $\|w\|_2 > s$  after the step of gradient descent.

In the model with two-channels as mentioned in Figure 4, one channel is used the frozen model by keeping the weights static without training, and the second channel is used the non-frozen model by allowing the weights to be updated along away the training. While the backpropagation is being from one channel, and each filter is conducted to every channel. Moreover, the findings were added to calculate  $C_i$  in the formula number (11).



**Figure 4** The baseline of CNN model construction.

## 4 Experiments

### 4.1 Pre-trained Word Embeddings

A tie-up relation exists between the proposed hybrid recommender system performance and the accuracy of CNNMs achievements for Sentiment Analysis. Consequently, the focus is on the results of CNNMs through implementing powerful CNN models, and then measuring their performance.

The effectiveness of the proposed CNNMs based on analysis of sentiments has been verified through conducting several experiments on ABHR-2 for Sentiment Analysis.

In order to build WEBSA and investigate the system efficiency, the CNN input must be handled first. Since it treats data with the numerical format which is to be obtained by applying the language models to ABHR-2 to extract vectors out of vocabulary.

Variant experiments have been done on ABHR-2 dataset mentioned in Subsection 3.2.4 in order to build the language model. Word vectors are initialized by using two modules SGBSA and CBOWBSA which have been trained on ABHR-2, and implemented through the unsupervised neural language model. Both models premise on grouping the words with similar meaning closely and dissimilar meaning far away based on either “Euclidean or Cosine distance in their geometry space. In addition, using the pre-trained FastText vectors dataset to create embeddings out of words that are publicly available and trained on Common Crawl and Wikipedia. After that, these pre-trained vectors are used as an input to CNNMs.

ABHR-2 vocabulary forms the input of both models (SGBSA and CBOWBSA) while their output becomes the vocabulary pre-trained vectors. The two language models (SGBSA and CBOWBSA) of word embeddings have been implemented on the text reviews of the ABHR-2 in order to produce the word vectors out of the corpus. Those pre-trained vectors are leveraged to be fed in the input to CNNMs as described in Subsection 4.2. The words that are not seen in the pretrained set, are randomly initialized.

## 4.2 Implementation of Various CNNMs

Here, WEBSA relies on CNNMs for multi-class classification tasks. The system concentrates on predicting the books that obtained the highest rating and linked to the user interests that captured by Semantic User Model mentioned in Section 3.1.1. For this aim, several empirical studies have been implemented on ABHR-2 in order to produce the pre-computed vectors out of the corpus to feed it to CNNMs as mentioned in Subsection 4.1. That results in the output of different NLP techniques become the input of CNN models.

Another dataset of word embeddings named FastText is used to validate the effectiveness of the suggested pretrained vectors mentioned in Section 4.1.

Ultimately, these pretrained vectors obtained from SGBSA and CBOWBSA, that are trained on ABHR-2 from scratch or FastText that is publicly available are utilized as input to train CNNMs. These models are classified into two options: either to keep these pretrained vectors frozen without training throughout the training process as present in the frozen models, or by keeping these vectors updated all along the training process which are presented in Non-frozen models.

We conduct a slight modification in the CNN architecture in order to get more competence by having a model with two sets of pre-trained vectors.

Each set is treated as a channel and the filter is applied on both sets but the backpropagation is conducted only through one set that is presented in the Specific-Task models.

The dimensionality of pre-trained vectors was 300. The pre-trained vectors are classified into SGBSA and CBOWBSA that trained on ABHR-2, and FastText publicly available. Then, each of these techniques is applied on variant models that are named as follows:

- **CNN-Frozen-SGBSA:** ‘SGBSA’ word vectors were adopted in order to train the model. The word embeddings of entire words, as well as the unknown words that are randomly initialized, and kept rigid during the training operation while the other parameters are permitted to learn.
- **CNN-Non-Frozen-SGBSA:** same as the previous one barring the word vectors are allowed for fine-tune in every epoch.
- **CNN-Specific-Task-SGBSA:** A small variance in CNN structure was customized in order to build this model. Where two series of word vectors initialized by utilizing SGBSA. Each set is regarded as a channel and each filter is conducted on each channel. Because this model is capable to fine-tune one set and leaving another set rigid, hence the gradients are backpropagated through one set only.
- **CNN-Frozen-CBOWBSA:** It differs from the CNN-Frozen-SGBSA in the way of initializing the word vectors. Wherein in this model, the CBOWBSA technique is applied to dataset vocabulary to get the word vectors to train the CNN.
- **CNN-Non-Frozen-CBOWBSA:** Similar to the above model beyond the fine-tuning can be applied for every epoch during training.
- **CNN-Specific-Task-CBOWBSA:** To perform this model, two sets of pretrained word vectors created by CBOWBSA technique, are combined where each set forming a channel. Diverse filters and kernel sizes used for both channels. The gradients are passed through one channel, while the second channel kept static.
- **CNN-Frozen-FastText:** In this model, the precomputed word vectors are obtained by using the FastText dataset to perform CNN. Each task is kept frozen while the other parameters allowed to be trained.
- **CNN-Non-Frozen-FastText:** Alike the former model, except each task are allowed for being updated during the training process.
- **CNN-Specific-Task-FastText:** Two channels of pretrained word vectors are conducting by utilizing FastText dataset. Different filters with windows of several sizes are employed on the two channels. The

gradients only through the updated channel while the second channel left frozen.

- **CNN-Specific-Task-MIK:** To train this model, the same CNN structure as in the previous CNN-Specific-Task-CBOWBSA model was used, but it makes use of pre-trained vectors from the ‘Mikolov’ [36] public dataset. In this case, ABHR-1 vocabularies are mapped to Mikolov pre-trained vectors to constitute the input of this model, which is then compared to the suggested models.
- **CNN-Specific-Task-GloVe:** The same CNN architecture as in the earlier CNN-Specific-Task-CBOWBSA model was utilized to train this model, but it uses word vectors from the public dataset ‘GloVe’ [48]. In this scenario, ABHR-1 vocabularies are mapped to GloVe as the model’s input, which would then be compared to the proposed models.
- **CNN-Specific-Task-DistilBERT:** This model is trained depend on a CNN-Specific-Task-CBOWBSA design with transformer-based pre-trained models for learning sentence trappings [52]. In this instance, the ABHR-1 corpus is translated to DistilBERT then feed to the model, and the proposed models are then compared.

### 4.3 SGBSA and CBOWBSA with ABHR-2

This experiment consists of two levels as interpreted in Figure 5. Level 1, the proposed language models are implemented by using the text reviews in ABHR-2. These two models (SGBSA and CBOWBSA) have been enhanced by training them on different parameters that are reported in the results. Each one, SGBSA and CBOWBSA produces one dataset of word vectors. Those vectors are so-called pre-trained\pre-computed vectors. Each word maps to a pre-trained vector with floating numbers. Thus, two models of pre-trained vectors have been obtained from SGBSA and CBOWBSA respectively.

Level 2 is the pre-trained vectors of each module which are mapped to the corpus vocabulary, where each word is associated with a respective vector. Those pre-computed vectors are to be served as input to the CNNMs for the purpose of classification. The words which are not seen in the pre-trained set are randomly initialized.

The input of CNNMs is the text reviews in ABHR-2 along with their corresponding rating which are represented through the language model. The text reviews are mapped to the obtained pre-trained vectors and those are further utilized to train CNNMs. Eventually, the CNNMs input is the respective pre-computed vectors of the corpus vocabulary along with their rating scores (1 to 5).



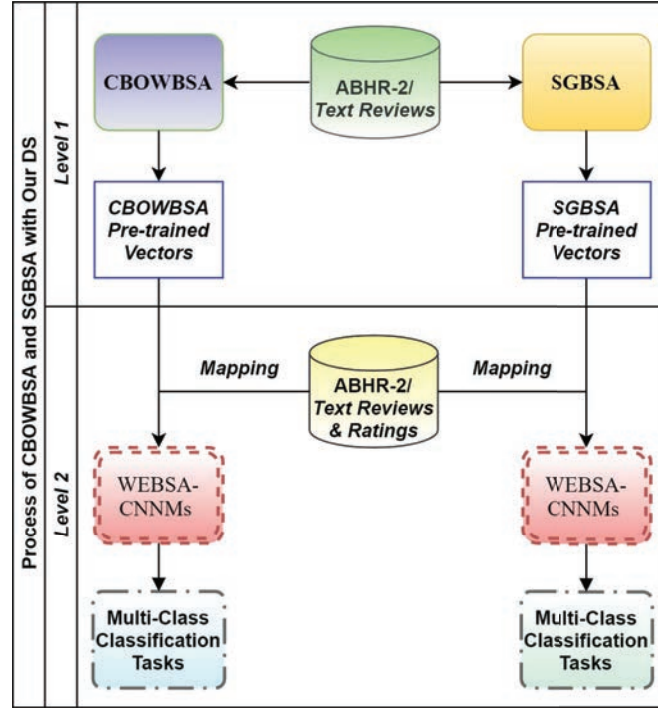
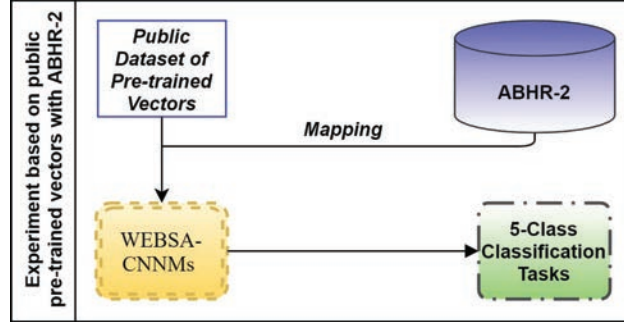


Figure 5 SGBSA and CBOWBSA with ABHR-2.

#### 4.4 Public Dataset of Pre-computed Vectors Mapping with ABHR-2

This experiment has also been conducted in order to evaluate the strength of our proposed language models. Instead of using the pre-computed vectors acquired by one of the suggested language models SGBSA and CBOWBSA, public dataset of 'FastText', 'GloVe' or 'Mikolov' is used here. Thus, in ABHR-2, the corpus vocabulary has been mapped to pre-trained vectors from 'FastText', 'GloVe' or 'Mikolov,' and then fed into the input for training various CNNMs as elaborated in Figure 6.

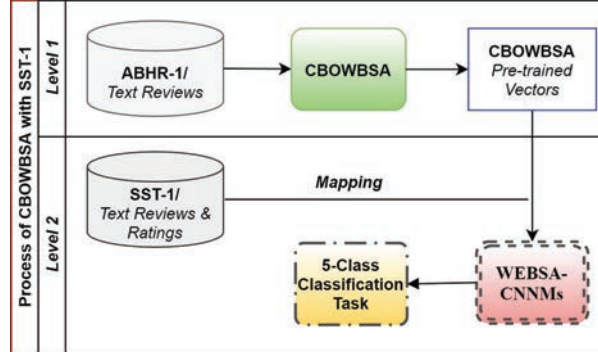
Furthermore, a transformer-based pre-trained DistilBERT model is used to learn sentence embeddings, where the ABHR-1 corpus is translated to DistilBERT and then fed to CNNMs.



**Figure 6** Experiment based on public pre-trained vectors with ABHR-2.

#### 4.5 CBOWBSA with Public Dataset ‘SST-1’

This experiment was carried out for evaluation against other public datasets ‘SST-1’ in order to check the robustness of the proposed language models. It involves two levels as given in Figure 7. Level 1, the system uses the pre-computed vectors of CBOWBSA that acquired the highest accuracy in this experiment for corpus representation as illustrated in Subsection 5.3.



**Figure 7** Experiment based on CBOWBSA pre-trained vectors with SST-1.

Level 2 – These CBOWBSA pre-computed vectors are utilized to vectorize SST-1 vocabulary. Each word of SST-1 is mapped to a particular pre-trained vector. Thus, the input of CNNMs is the respective pre-computed vectors of the SST-1 vocabulary along with their rating scores [1–5] for fine-grained sentiment classification. Words that do not appear in the pre-trained language model are randomly initialized.

## 5 Results and Discussion

Several experiments have been conducted in order to compare the performance of variant methods and techniques, then estimate the findings of the different CNNMs mentioned in Section 4.2 that have been trained on ABHR-2. Four evaluation measures are utilized accuracy, precision, recall, and F1 score as described in Table 1.

**Table 1** The Performance comparison of our models

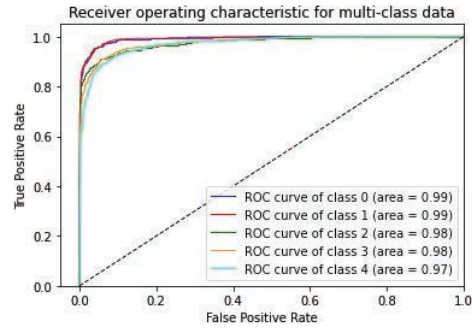
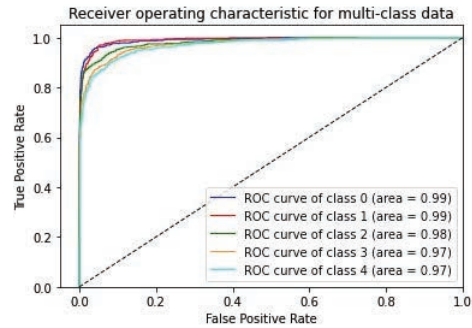
Datasets	Word Embeddings	Model	Accuracy	Precision	Recall	F1
ABHR-2	SGBSA	CNN-Frozen-SGBSA	49.72%	60.91%	49.88%	48.96%
		CNN-Non-Frozen-SGBSA	86.27%	86.81%	86.36%	86.23%
		CNN-Specific-Task-SGBSA	88.59%	88.64%	88.52%	88.44%
	CBOWBSA	CNN-Frozen-CBOWBSA	<b>52.32%</b>	<b>54.11%</b>	<b>52.72%</b>	<b>51.69%</b>
		CNN-Non-Frozen-CBOWBSA	<b>88.02%</b>	<b>88.08%</b>	<b>88.07%</b>	<b>87.93%</b>
		<b>CNN-Specific-Task-CBOWBSA</b>	<b>89.26%</b>	<b>89.38%</b>	<b>89.33%</b>	<b>89.25%</b>
	FastText [42]	CNN-Frozen-FastText	31.89%	43.48%	31.94%	27.42%
		CNN-Non-Frozen-FastText	85.24%	85.82%	85.40%	85.31%
		CNN-Specific-Task-FastText	87.99%	88.10%	87.88%	87.84%
	Mikolov [36]	CNN-Specific-Task-MIK	54.52%	54.03%	54.54%	54.21%
	GloVe [48]	CNN-Specific-Task-GloVe	83.22%	83.58%	83.13%	83.11%
	DistilBERT [52]	CNN-Specific-Task-DistilBERT	59.01%	34.82%	20.00%	14.84%
SST-1 [43]	CBOWBSA	CNN-Specific-Task-CBOWBSA	86.52%	86.63%	86.44%	86.43%
Hotel reviews from booking.com [34]	Mikolov [36]	CNN based [34]	88.52%	94.30%	85.63%	86.03%
Movie review [47]	Mikolov [36]	Their proposed model [46]	45.4%	—	—	—
SST-1 [43]	Mikolov [36]	CNN-multichannel [22]	47.4%	—	—	—

Various models of machine learning can be appraised by the “receiver operating characteristics (ROC) curve”. This is used to compare diagnostic tests by plotting the rate of true-positive against the rate of false-positive based on different threshold settings. For multiclass classification problems, we can draw ROC curve by considering every class as a binary prediction of the label indicator matrix (micro/averaging). It demonstrates the specificity and sensitivity and of the model to select classes as mentioned in Table 2.

**Table 2** ROC curve for evaluating the performance

Class \ Model	Accuracy	Very Negative	Negative	Natural	Positive	Very Positive
CNN-Specific-Task-SGBSA	98.13%	0.99	0.99	0.98	0.97	0.97
<b>CNN-Specific-Task-CBOWBSA</b>	<b>98.15%</b>	<b>0.99</b>	<b>0.99</b>	<b>0.98</b>	<b>0.98</b>	<b>0.97</b>
CNN-Specific-Task-FastText	97.52%	0.99	0.99	0.97	0.97	0.96
CNN-Specific-Task-MIK	83.83%	0.91	0.80	0.80	0.77	0.91
CNN-Specific-Task-GloVe	97.38%	0.99	0.99	0.98	0.97	0.95
CNN-Specific-Task-CBOWBSA with SST-1	97.35%	0.98	0.99	0.97	0.97	0.95
CNN-Specific-Task-DistilBERT	51.08%	0.50	0.48	0.52	0.48	0.52

The following Figures 8 to 14 show ROC curve performance for every class of the previous models.

**Figure 8** CNN-Specific-Task-SGBSA.**Figure 9** CNN-Specific-Task-CBOWBSA.

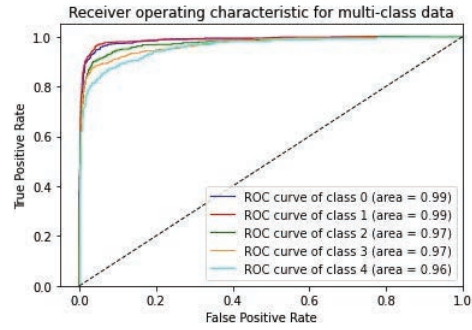


Figure 10 CNN-Specific-Task-FastText.

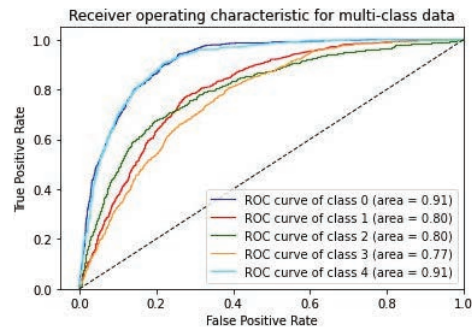


Figure 11 CNN-Specific-Task-MIK.

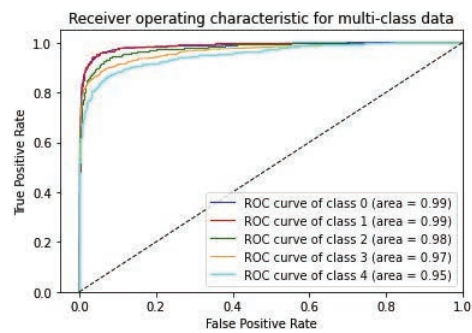
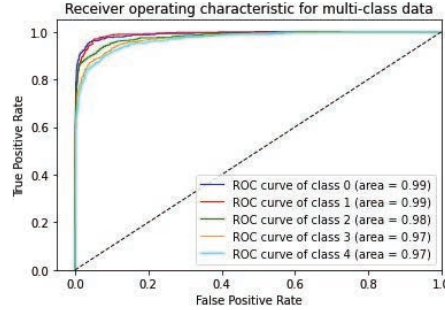
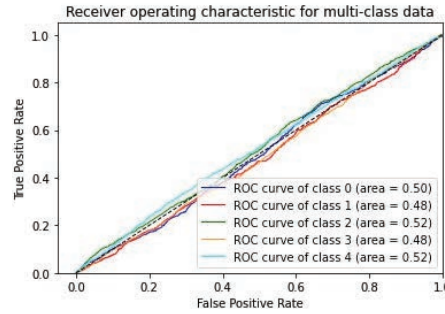


Figure 12 CNN-Specific-Task-GloVe.

**Figure 13** CNN-Specific-Task-CBOWBSA with SST-1.**Figure 14** CNN-Specific-Task-DistilBERT.

We can see from measuring the performance of the ROC curve that our augmented language models outperform other models.

We implemented several empirical studies to examine CBOWBSA and SGBSA for word initialization against sophisticated deep learning models based on ABHR-2.

For initializing word vectors, the system utilized the enhanced language models CBOWBSA and SGBSA that bring best representation for the corpus, which is clearly reflected in the performance of WEBSA. These pre-computed vectors resulted in more powerful performance that is presented in the findings of the CNNMs in terms of accuracy 89.26%. To assess the effectiveness of the improved language models, public datasets of pre-computed vectors FastText, Mikolov, and GloVe are used to train CNN-Specific-Task-FastText, CNN-Specific-Task-MIK, and CNN-Specific-Task-GloVe, which accomplish accuracy of 87.99%, 54.52%, and 83.22%, respectively; additionally, CNN-Specific-Task-DistilBERT, which is based on the DistilBERT transformer-based model, performs 59.01%, which is lower than the results obtained by CNN-Specific-Task-SGBSA, which is 89.26%,

and CNN-Specific-Task-CBOWBSA, which is 88.59%. Furthermore, these language models are compared to the state-of-the-art models [34] and [46] that reported an accuracy 88.52% and 45.4% respectively, both of which are lower than our developed models.

Moreover, the best language model CBOWBSA in our experiments is validated with CNN-Specific-Task-CBOWBSA that obtained better accuracy 86.52% than the CNN-multichannel model [22] implemented by using the same dataset ‘SST-1’. This prove the dataset created for a specific domain “Education domain” is better than the one that crated for a general one.

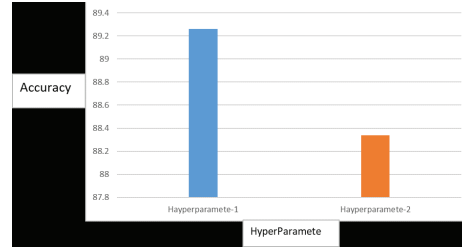
We expected to acquire giant performance by using pre-trained vectors (SGBSA and CBOWBSA) that have trained on ABHR-2, but the achievements were remarkable against the results of FastTest, Mikolov, GloVe and transformer-based model DistilBERT. Despite the simplicity of frozen vectors (CNN-Frozen), it worked very well. It gives the lowest accuracy against the remaining models with low overfitting.

The models with leaving vectors to finetune for each task (CNN-Non-Frozen-) show excellent results in the field of accuracy while it brings more overfitting. For acquiring competitive outcomes and gaining further improvements, we applied models with two channels (CNN-Specific-Task-), where it presents the highest accuracy and the lowest overfitting, and performed greater measuring.

The enhanced CNNMs have been trained and tested on various parameters as elaborated in Table 3, where their determinants have been reported based on the several experiments of the self-designed (CNN-Frozen,

**Table 3** CNNMs training parameters

	Hyperparamete-1	Hyperparamete-2
No. of convolutional layers	3	2
Filter sizes	3, 4, 5	3, 4, 5
Feature maps	200	190
L2-norm constraint	3	2
Dropout ratio	0.5	0.2
Batch size	49	40
“Dev set” size	10% out of the training set is randomly selected.	10% out of the training set is randomly selected.
Training the model	Stochastic gradient descent was used over shuffled mini-batches.	Stochastic gradient descent was used over shuffled mini-batches.
Optimizer	“Adam Optimization” [44].	“Adam Optimization” [44].
No. of iterations	50	30



**Figure 15** for optimizing the Hyperparameter of training CNN-Specific-Task-CBOW.

CNN-Non-Frozen-, CNN-Specific-Task-) with self-created (CBOWBSA and SGBSA).

The Figure 15 shows training CNN-Specific-Task-CBOW based on two sets of hyperparameters; Hayperparamete-1 and Hayperparamete-2 that is given above.

These parameters were optimized by utilizing “random search technique”. CNNMs are implemented by leveraging “Keras and Gensim Package” of Python.

#### **A. Experimental results by using CBOWBSA and SGBSA with ABHR-2:**

Comparing the previous models that depend on CBOWBSA and SGBSA can observe the CBOWBSA with three sophisticated deep learning models performed remarkable results. Moreover, the CNN-Specific-Task-CBOWBSA model produced brilliant findings in the field of accuracy 89.26% as shown in Table 1. The two models (CBOWBSA and SGBSA) are enhanced by training them on the various hyperparameters that where Dimensionality is 300, Window size between the present and forecast word is 20 and configuring the sample of frequency words taken is  $6e-5$ , negative sampling of noise words is 20, Min\_Count for the word frequency is lower than 5, the learning rate equal to 0.003, the hierarchical softmax for model is used  $hs = 1$ . These models are trained based on 300 Iterations.

#### **B. Experimental results by using public dataset of pre-computed vectors and ABHR-2:**

CNNMs perform better in simulation results with the suggested language models than Mikolov, FastText and GloVe when applied to ABHR-2 to train CNNMs for sentiment categorization. The CNN-Specific-Task-FastText model achieved an accuracy of 87.99%. on the other hand, CNN-Specific-Task-MIK acquire 54.52% accuracy, while for CNN-Specific-Task-GloVe accomplished 83.22% in the term of accuracy.



Additionally, the transformer-based model DistilBERT combined with CNN-Specific-Task- DistilBERT yielded accuracy 59.01%. Importantly, our investigative results outperformed the results given in [34] and [46] that reported an accuracy of 88.52% and 45.4%, respectively.

This demonstrates the efficacy of our language models, which accomplished the best accuracy.as shown in Table 1.

**C. Experimental results of using CBOWBSA with Public dataset:**

SST-1 vocabulary is mapped into CBOWBSA pre-computed vectors that are employed to train the CNN-Specific-Task-CBOWBSA model to validate the robustness of the suggested language models. Table 1 indicates that the experimental findings are 86.52% better than previous models constructed by Kim [22], which achieved 47.4% lower accuracy while utilizing the similar data set “SST-1.”

**Discussion:**

The proposed system is used to capture learners’ preferences through semantic user model described in Section 3.1.1 based on learner’s behaviors and the extracted semantic relation matrix. That is efficiently reflected on the findings of building the user model successfully.

WEBSA is a stepping stone of the recommendation process based on learner preferences taken by the semantic user model. The terms extracted from the user model are forming the true needs of the learner. Recommendation System must predict the suitable e-learning resources for the learner based on these preferences. The use of rich semantic network like ConceptNet leads more accurate results to generate the semantic matrix that gives the correlation between terms and materials. Thus, aid to build the user model that include more specific terms semantically related to the user’s interests. Moreover, it handles the “sparsity” problem since the user model has been represented as vector of relative relations.

It can be seen from results that WEBSA based model integrated with CNN-Specific-Task-CBOWBSA and CBOWBSA pretrained vectors effectively out-perform the rest models, where ABHR-2 is used. This resulted in the performance in terms of accuracy **89.26%**, Precision 89.38, Recall 89.33 and F1 89.25 is higher than the use of public dataset created general purpose that bring an accuracy 47.4% in [22].

Using the public datasets of pre-trained vectors FastText, Mikolov, and GloVe to obtain word embeddings to vectorize ABHR-2 vocabulary, lower accuracy results of 87.99%, 54.52%, and 83.22% were obtained in terms of CNN-Specific-Task-FastText, CNN-Specific-Task-MIK, and CNN-Specific-Task-GloVe models respectively, as well as the

transformer-based model DistilBERT combined with CNN-Specific-Task-DistilBERT produced an accuracy of 59.01% when compared to the results obtained by applying pre-trained vectors from CBOWBSA and SGBSA to train CNN-Specific-Task-CBOWBSA and CNN-Specific-Task-SGBSA models.

Additionally, the findings demonstrated the efficiency of the proposed language models when compared to [34] and [46], which employed the public dataset of pretrained vectors.

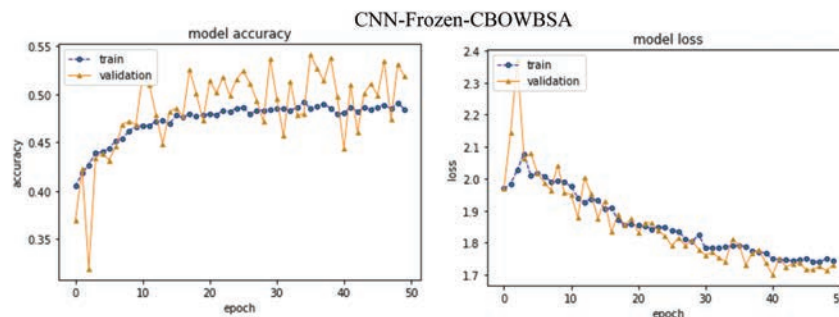
The impressive results and the performance of the ROC Curve justify and prove the validity and robustness of the pre-trained CBOWBSA and SGBSA vectors and give proof of epistemic power of ABHR-2.

Following Figures 16–18 elaborate the accuracy against epoch and loss against epoch for the CNNMs relied on best language model CBOWBSA for initializing word vectors that achieved the greatest accuracy with the lowest overfitting. Figures 19–21 reported the accuracy and Loss against epoch for CNNMs integrated with SGBSA for word vectorization. The CNNMs combined with FastText are shown in Figures 22–24 for accuracy and Loss against epoch. Eventually, the results of accuracy against epoch and loss against epoch are given by applying CNN-Specific-Task-MIK depend on Mikolov and CNN-Specific-Task-GloVe based on GloVe and presented in Figures 25 and 26. The loss and accuracy of transformer-based model DistilBERT combined with CNN-Specific-Task- CNN-Specific-Task-DistilBERT is given in Figure 27.

Moreover, in comparison CNNMs with each another, we observed the CNN-Specific-Task- with entire language models obtained the top results against CNN-Frozen- and CNN-Non-Frozen-.

Ultimately, CNN-Specific-Task-CBOWBSA achieved the greatest accuracy 89.26% with the lowest overfitting as seen in the Figure 18.

#### **CNNMs integrated with CBOWBSA Figures:**



**Figure 16** CNN-Frozen-CBOWBSA.

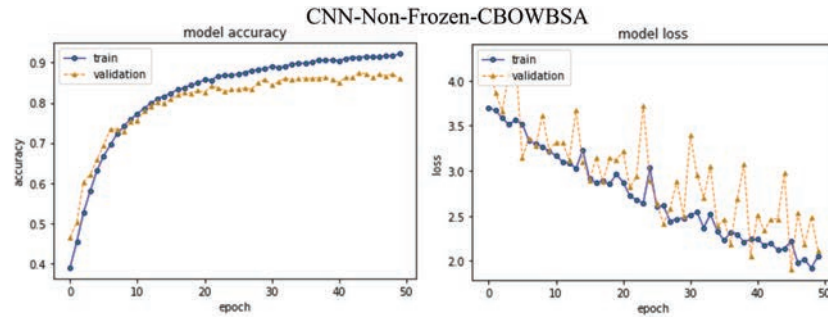


Figure 17 CNN-Non-Frozen-CBOWBSA.

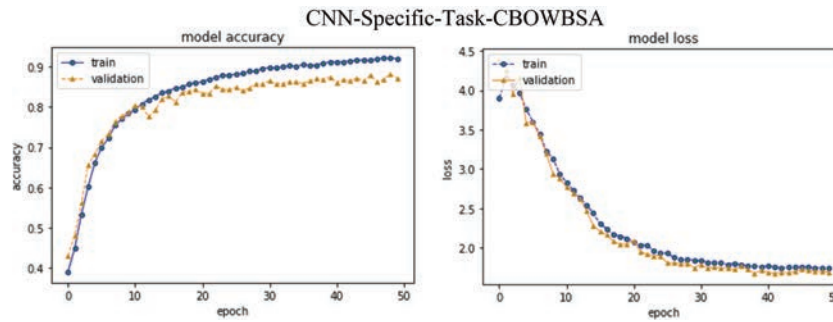


Figure 18 CNN-Specific-Task-CBOWBSA.

### CNNMs integrated with SGBSA Figures:

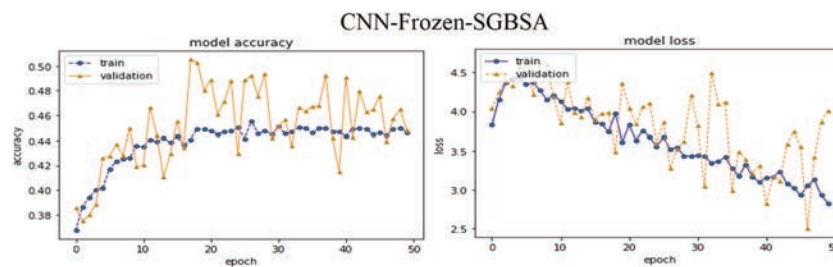
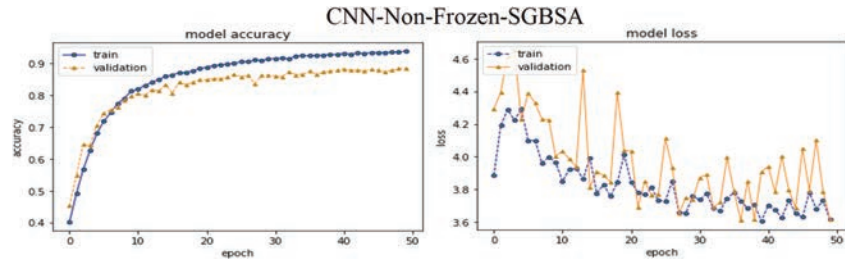
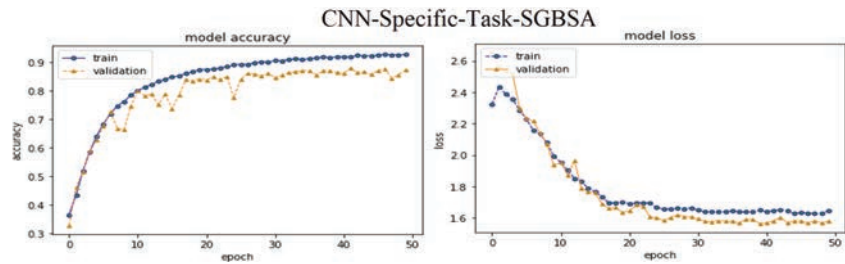


Figure 19 CNN-Frozen-SGBSA.

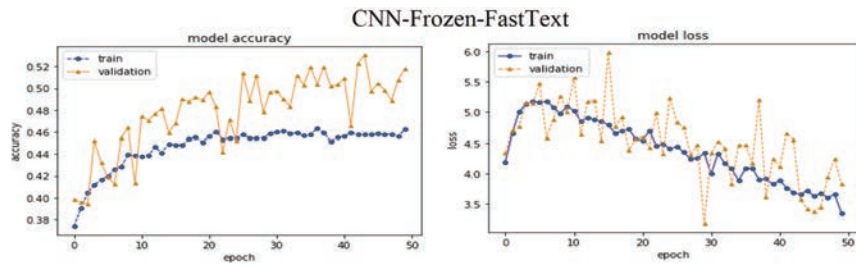


**Figure 20** CNN-Non-Frozen-SGBSA.

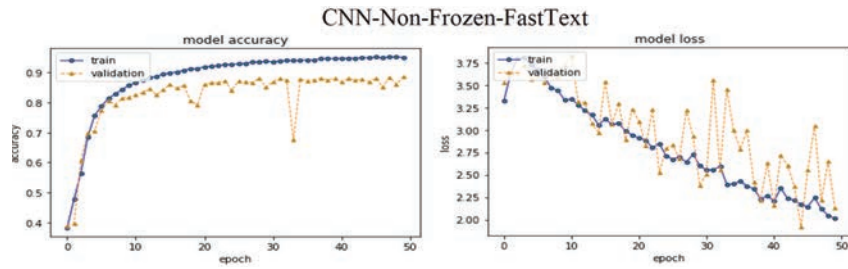


**Figure 21** CNN-Specific-Task-SGBSA.

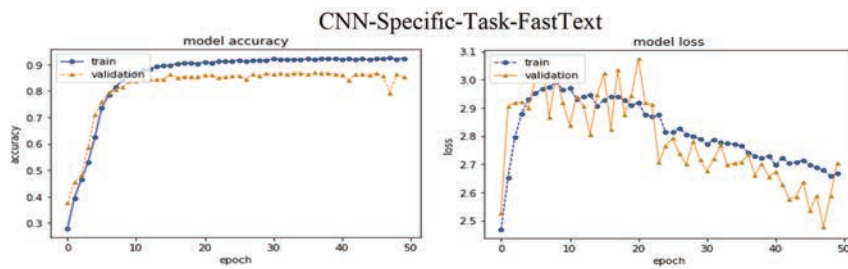
### CNNMs integrated with FastText Figures:



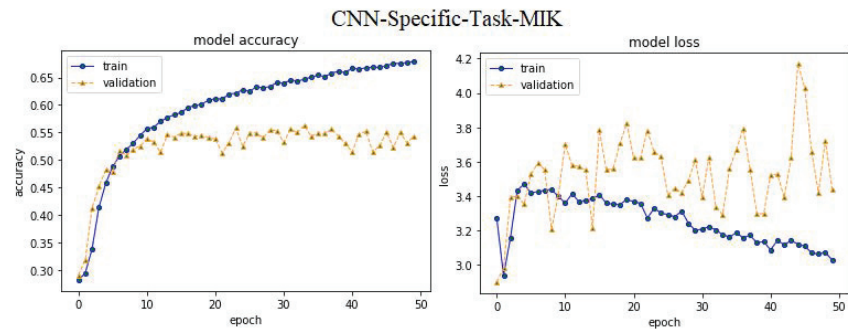
**Figure 22** CNN-Frozen-FastText.



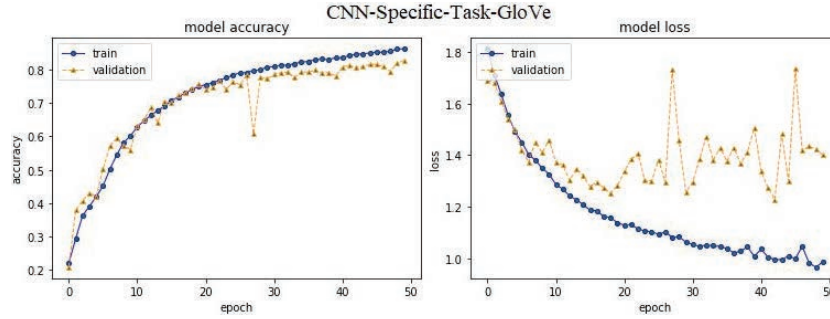
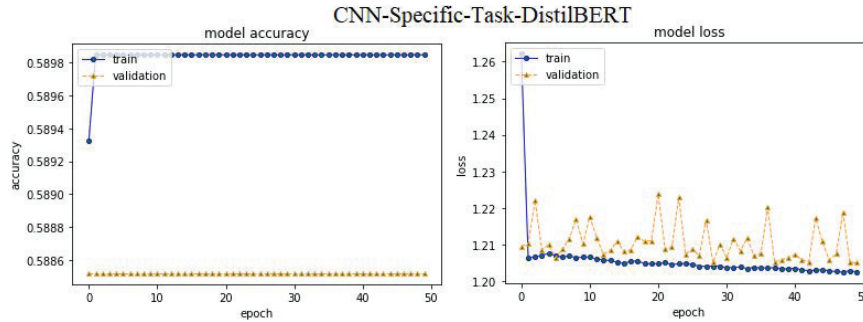
**Figure 23** CNN-Non-Frozen-FastText.



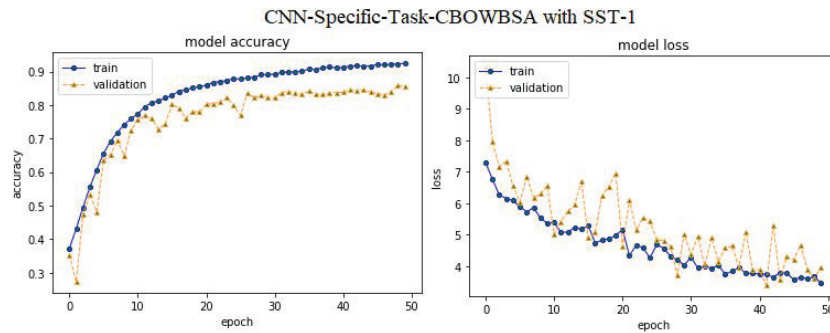
**Figure 24** CNN-Specific-Task-FastText.



**Figure 25** CNN-Specific-Task-MIK.

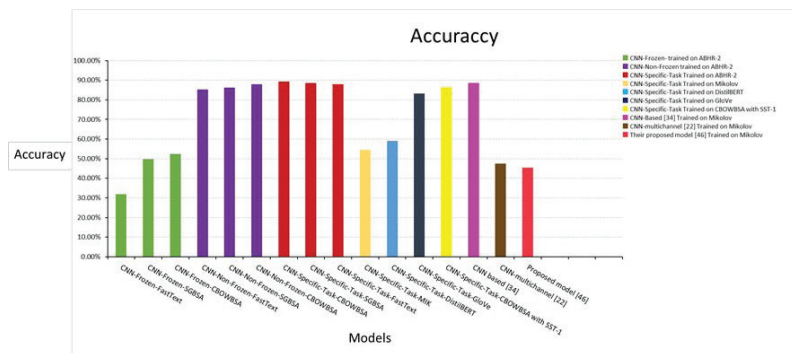
**Figure 26** CNN-Specific-Task-GloVe.**Figure 27** CNN-Specific-Task-DistilBERT.

Furthermore, in our experiments, the best language model CBOWBSA was utilized to vectorize the ‘SST-1’ vocabulary and validated using CNN-Specific-Task-CBOWBSA, which had a higher accuracy of 86.52%, as seen in Figure 28.

**Figure 28** CNN-Specific-Task-CBOWBSA with SST-1.

We observed striking results with evolved performance when applying Sentiment Analysis on the ABHR-2 created for e-learning domain. It is labeled with 5 classes compared to most researchers nowadays where they conduct their research using SA with polarity to classify reviews as positive or negative. This provides a more forward-looking and reinforced operation which brings to Sentiment Analysis process more challenging and powerful.

Summarizing, Utilizing CBOWBSA, SGBSA, FastText, Mikolov, GloVe, and DistilBERT to learn the embeddings, it has discovered that pre-trained vectors CBOWBSA and SGBSA trained on ABHR-2 for a particular subject produced better results than pre-trained vectors FastText, Mikolov, and GloVe trained on public data as well as the DistilBERT transformer model, via CNNMs performance. The best model out of CNNMs mentioned in Section 4.2 was CNN-Specific-Task-CBOWBSA that brings 89.26% accuracy, and for CNN-Specific-Task-SGBSA 88.59% accuracy, while for CNN-Specific-Task-FastText it is 87.99% and CNN-Specific-Task-Mik the accuracy obtained is 54.52% and CNN-Specific-Task-GloVe produces 83.22% and CNN-Specific-Task-DistilBERT outputted 59.01% as shown in Figure 29. Furthermore, these language models are compared to latest models [34] and [46], which report accuracy of 88.52% and 45.4% accordingly, which are both lower than our remarkable language models. The model CNN-Specific-Task-CBOWBSA is used to vectorize the vocabulary of textual dataset “SST-1” that reports an accuracy 86.52%. This demonstrates the efficacy of our language models that trained on ABHR-2 designed for the education domain in comparison to other models [22] that achieve accuracy 47.4% using same dataset.



**Figure 29** Accuracy comparison of our models based on our dataset against others.

The suggested recommender model can be utilized as a new system for recommendation. Moreover, it could be leveraged to evolve the current recommendation approach that exists in e-learning platforms.

## 6 Conclusion

Enhancing e-learning platforms requires an efficient recommender system to gain a successful learner recommendation process. That is presented in this work by adopting the various deep learning models and NLP techniques. Many studies addressed this concept from a narrower scope, disregarding several influential factors and other necessary components.

A hybrid recommender system has been suggested by integrating the semantic user model and WEBSA. The proposed model concentrates on predicting the relevant e-learning resource with highest ratings linked to the learner preferences leveraging CNNMs combined with NLP techniques. User Model is built in an implicit manner depending on the learner's behavior and semantic text representation with contextual graph to capture the learner terms. Therefore, semantic relations matrix linked between the terms and the materials was generated which relies on material concepts and terms expansion utilizing ConceptNet.

The WEBSA is able to classify the e-learning resources in a range of 5 classes and formulate the recommendation process based on the learner profile that is deduced by the semantic user model. To enrich the recommendation, dataset ABHR-2 is utilized that consists of users' reviews labeled with 5 classes to train CNNMs. This displayed significantly improved performance and resulted in meaningful outcomes comparing with other models presented in the literature. Our investigations support the well-established proof that leverages the pre-computed CBOWSA and SGBSA trained on ABHR-2 which give better representation of the corpus. This leads to the recommendation process in the right direction.

Future work will focus on finding more efficient deep learning networks. Other techniques can be explored such as Bert-Based models, Hidden Markov models with Part of Speech tagging in sentiment analysis. The semantic user model can also be investigated by utilizing different approaches. Explore an optimized transformer-based model rely on the BERT architecture for embedding learning.



## References

- [1] K. Mangaroska, M. Giannakos, ‘learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning’, *IEEE Transactions on Learning Technologies*, 12(4), 516–534, 2018.
- [2] V. Kubik, R. Gaschler, H. Hausman, PLAT 20 (1) 2021: Enhancing Student Learning in Research and Educational Practice, *The Power of Retrieval Practice and Feedback*, 2021.
- [3] S. Benzarti, R. Faiz, EgoTR: Personalized tweets recommendation approach. In *Computer Science On-line Conference*, pp. 227–238, 2015. Springer, Cham. [https://doi.org/10.1007/978-3-319-18503-3\\_23](https://doi.org/10.1007/978-3-319-18503-3_23)
- [4] L. Boratto, S. Carta, G. Fenu, R. Saia, Semantics-aware content-based recommender systems: Design and architecture guidelines. *Neurocomputing*, 254, 79–85, 2017. <https://doi.org/10.1016/j.neucom.2016.10.079>.
- [5] Y. Kim, K. Shim, TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation. *Information Systems*, 42, 59–77, 2014. Doi: 10.1016/j.is.2013.11.003.
- [6] W. Cui, Y. Du, Z. Shen, Y. Zhou, J. Li, Personalized microblog recommendation using sentimental features. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* pp. 455–456, 2017. IEEE. Doi: 10.1109/BIGCOMP.2017.7881756.
- [7] X. Zhou, S. Wu, C. Chen, G. Chen, S. Ying, Real-time recommendation for microblogs. *Information Sciences*, 279, 301–325, 2014. Doi: 10.1016/j.ins.2014.03.121.
- [8] H. Ezaldeen, R. Misra, R. Alatrash, R. Priyadarshini, Semantically enhanced machine learning approach to recommend e-learning content. *International Journal of Electronic Business*, 15(4), 389–413, 2020. <https://doi.org/10.1504/IJEB.2020.111095>.
- [9] F. Abel, Q. Gao, G. J. Houben, K. Tao, Semantic enrichment of twitter posts for user profile construction on the social web. In *Extended semantic web conference* pp. 375–389, 2011, May. Springer, Berlin, Heidelberg. Doi: 10.1007/978-3-642-21064-8\_26.
- [10] V. de Graaff, A. van de Venis, M. van Keulen, A. Rolf, Generic knowledge-based Analysis of Social Media for Recommendations. In *CBRecSys@ RecSys*, pp. 22–29, 2015, September.
- [11] G. Piao, J. G. Breslin, Exploring dynamics and semantics of user interests for user modeling on Twitter for link recommendations. In *proceedings*

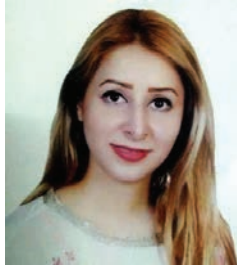
- of the 12th international conference on semantic systems pp. 81–88, 2016, September. Doi: 10.1145/2993318.2993332.
- [12] M. Keshavarz, Y H. Lee, Ontology matching by using ConceptNet. In Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference Vol. 2012, pp. 1917–1925, 2012.
- [13] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the AAAI Conference on Artificial Intelligence Vol. 31, No. 1, 2017, February.
- [14] Z. Han, J. Wu, C. Huang, Q. Huang, M. Zhao, A review on sentiment discovery and analysis of educational big-data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(1), e1328, 2020.
- [15] K. Mite-Baidal, C. Delgado-Vera, E. Solís-Avilés, A H. Espinoza, J. Ortiz-Zambrano, E. Varela-Tapia, Sentiment analysis in education domain: A systematic literature review. In International Conference on Technologies and Innovation pp. 285–297, 2018, November. Springer, Cham. [https://doi.org/10.1007/978-3-030-00940-3\\_21](https://doi.org/10.1007/978-3-030-00940-3_21).
- [16] A. ONAN, Sentiment analysis on massive open online course evaluations: A text mining and deep learning approach. *Computer Applications in Engineering Education*, 2020. <https://doi.org/10.1002/cae.22253>.
- [17] R. Cobos, F. Jurado, A. Blázquez-Herranz, A Content Analysis System that supports Sentiment Analysis for Subjectivity and Polarity detection in Online Courses. *IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje*, 14(4), 177–187, 2019. Doi: 10.1109/RITA.2019.2952298.
- [18] A. Magdy, L. Abdelhafeez, Y. Kang, E. Ong, M F. Mokbel, Microblogs data management: a survey. *The VLDB Journal*, 29(1), 177–216, 2020.
- [19] M L B. Estrada, R Z. R O. Cabada, Bustillos, M. Graff, Opinion mining and emotion recognition applied to learning environments. *Expert Systems with Applications*, 150, 113265, 2020. <https://doi.org/10.1016/j.eswa.2020.113265>.
- [20] N. Kiuru, B. Spinath, A L. Clem, K. Eklund, T. Ahonen, R. Hirvonen, The dynamics of motivation, emotion, and task performance in simulated achievement situations. *Learning and Individual Differences*, 80, 101873, 2020. <https://doi.org/10.1016/j.lindif.2020.101873>.
- [21] C. Salazar, J. Aguilar, J. Monsalve-Pulido, E. Montoya, Affective recommender systems in the educational field. A systematic literature review. *Computer Science Review*, 40, 100377, 2021.
- [22] Y. Kim, Convolutional Neural Networks for Sentence Classification. 2014r08r25, 2014. <https://arxiv.org/abs/1408.5882>.

- [23] R. Johnson, T. Zhang, Semi-supervised convolutional neural networks for text categorization via region embedding. *Advances in neural information processing systems*, 28, 919, 2015.
- [24] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun, Very deep convolutional networks for natural language processing, 2016. arXiv preprint arXiv:1606.01781, 2, 1.
- [25] R. Alatrash, H. Ezaldeen, R. Misra, R. Priyadarshini, Sentiment Analysis Using Deep Learning for Recommendation in E-Learning Domain. *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2020*, 123, 2020. [https://doi.org/10.1007/978-981-33-4299-6\\_10](https://doi.org/10.1007/978-981-33-4299-6_10).
- [26] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web*, pp. 373–374, 2014, April. <https://doi.org/10.1145/2567948.2577348>.
- [27] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, 2014. arXiv preprint arXiv:1404.2188. <http://arxiv.org/abs/1404.2188>.
- [28] W T. Yih, X. He, C. Meek, Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* pp. 643–648, 2014, June.
- [29] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE), 2493–2537, 2011.
- [30] N. Shrestha, F. Nasoz, Deep learning sentiment analysis of amazon. com reviews and ratings, 2019. arXiv preprint arXiv:1904.04096.
- [31] K S. Srujan, S S. Nikhil, H R. Rao, K. Karthik, B S. Harish, H K. Kumar, Classification of amazon book reviews based on sentiment analysis. In *Information Systems Design and Intelligent Applications* pp. 401–411, 2018. Springer, Singapore. [https://doi.org/10.1007/978-981-10-7512-4\\_40](https://doi.org/10.1007/978-981-10-7512-4_40).
- [32] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, 2015. arXiv preprint arXiv:1509.01626.
- [33] A M. Qamar, M. Alassaf, Improving Sentiment Analysis of Arabic Tweets by One-Way ANOVA. *Journal of King Saud University-Computer and Information Sciences*, 2020. <https://doi.org/10.1016/j.jksuci.2020.10.023>.

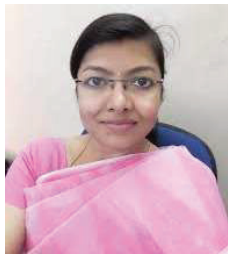
- [34] R. Kumar, H S. Pannu, A K. Malhi, Aspect-based sentiment analysis using deep networks and stochastic optimization. *Neural Computing and Applications*, 32(8), pp. 3221–3235, 2020. <https://doi.org/10.1007/s00521-019-04105-z>.
- [35] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013. arXiv preprint arXiv:1301.3781.
- [36] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, 2013. arXiv preprint arXiv:1310.4546.
- [37] E.C. Too, L. Yujian, P.K. Gadosey, S. Njuki, F. Essaf, Performance analysis of nonlinear activation function in convolution neural network for image classification. *International Journal of Computational Science and Engineering*, 21(4), pp. 522–535, 2020.
- [38] G E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012. arXiv preprint arXiv:1207.0580.
- [39] J. Moravec, A Comparative Study: L1-Norm Vs. L2-Norm; Point-to-Point Vs. Point-to-Line Metric; Evolutionary Computation Vs. Gradient Search. *Applied Artificial Intelligence*, 29(2), 164–210, 2015.
- [40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958, 2014.
- [41] R. Alatrash, H. Ezaldeen, rawaa123/Dataset GitHub Retrieved from <https://github.com/rawaa123/Dataset/>, 2021.
- [42] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146, 2017. [https://doi.org/10.1162/tacl\\_a.00051](https://doi.org/10.1162/tacl_a.00051).
- [43] R. Socher, A. Perelygin, J. Wu, J. Chuang, C D. Manning, A Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013, October.
- [44] D P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014. arXiv preprint arXiv:1412.6980.
- [45] H. Ezaldeen, R. Misra, R. Alatrash, R. Priyadarshini, Machine Learning Based Improved Recommendation Model for E-learning. In *2019 International Conference on Intelligent Computing and Remote Sensing (ICICRS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICICRS46726.2019.9555866>.

- [46] X. Ouyang, P. Zhou, C.H. Li, L. Liu, Sentiment analysis using convolutional neural network. In: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing (CIT/ IUCC/DASC/PICOM). IEEE, 2015, pp. 2359–2364. <https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.349>
- [47] B. Pang, L. Lee “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 115–124, 2005.
- [48] J. Pennington, R. Socher, and C. D. Manning, Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532–1543) (2014, October).
- [49] M. Abbasi, G. Montazer, F. Ghrohani, and Z. Alipour, Categorizing E-Learner Attributes in Personalized E-learning Environments: A Systematic Literature Review. *Interdisciplinary Journal of Virtual Learning in Medical Sciences*, 12(1), 1–21 (2021).
- [50] A. Nandi, F. Khafa, L. Subirats, and S. Fort, Real-time emotion classification using eeg data stream in e-learning contexts. *Sensors*, 21(5), 1589, (2021). DOI: <https://doi.org/10.3390/s21051589>.
- [51] N. Mejbri, F. Essalmi, M. Jemni, and B. A. Alyoubi, Trends in the use of affective computing in e-learning environments. *Education and Information Technologies*, 1–23 (2021). DOI: <https://doi.org/10.1007/s10639-021-10769-9>.
- [52] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, (2019). arXiv preprint [arXiv:1910.01108](https://arxiv.org/abs/1910.01108).
- [53] Priyadarshini, R., Barik, R. K., and Dubey, H. (2018). Deepfog: Fog computing-based deep neural architecture for prediction of stress types, diabetes and hypertension attacks. *Computation*, 6(4), 62.
- [54] R. Priyadarshini, R. K. Barik, C. Panigrahi, H. Dubey, and B. K. Mishra, An investigation into the efficacy of deep learning tools for big data analysis in health care. In *Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications* (pp. 654–666). IGI Global, (2020).
- [55] R. Priyadarshini, R. K. Barik, H. Dubey. “Fog-SDN: A light mitigation scheme for DDoS attack in fog computing framework.” *International Journal of Communication Systems* 33, no. 9 (2020): e4389.

## Biographies



**Rawaa Alatrash** is a Ph.D. scholar in Computer Science and Engineering at the C. V. Raman Global University (CGU), India. She received her Master's degree in Computer Science and Engineering from CGU, India. She has more than four years of industrial experience and worked on different Java based projects. Her main research interests include Deep Learning, Knowledge-based Systems, and Natural Language Processing, Sentiment Analysis, Recommendation Systems. She has publications in reputed Journals and International Conferences indexed in SCI and Scopus. She is a web developer, has done several remarkable projects in Web Intelligence and Sentiment Analysis. Reviewed papers in journals like River Publishers and Science Publishing Group.



**Rojalina Priyadarshini:** Completed her Ph.D in 2020 and currently is working as an Assistant Professor in the C.V.Raman Global University, Odisha. She has grabbed a Gold medal in her master degree. She has more than a decade of teaching and research experience. Apart from that she is an accredited AWS cloud educator and a certified Java faculty. Dr. R. Priyadarshini has published more than 50 research papers in peer-reviewed International Journals and conferences. Her areas of specialization are cloud and fog computing, Machine Learning and Cyber security.



**Hadi Ezaldeen** is a Ph.D. in Computer Science and Engineering from C.V. Raman Global University, India. He received his Master's degree in Web Science (MWS) from the Syrian Virtual University (SVU), Damascus, Syria. He has worked as a tutor (Assistant Professor), and an assistant supervisor of several graduation projects in the Faculty of Information Technology Engineering (ITE), at SVU. His research areas include Machine Learning, Semantic Analysis, Recommendation Systems, Natural Language Processing, and Text Mining. He has publications in reputed Journals and International Conferences indexed in SCI and Scopus. He has achieved interesting projects in the field of Web Science.



**Akram Ahinnawi** has graduated from master in Computer Science, School of Engineering at the University of Bridgeport, CT, USA at May 2013. He worked as a Web Developer, QA System Test Engineer, Senior Software Development Test Engineer, Senior Automation Test Engineer and now he has been working as a Senior Security Test Engineer at Fujifilm Medical System Inc. USA at Morrisville Office, NC, USA for 8 years.

