EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

Efficient Course Recommendation using Deep Transformer based Ensembled Attention Model

A. Madhavi[1,*], A. Nagesh[2] and A. Govardhan[3]

[1]Department of CSE, VNR Vignana Jyothi Institute of Engineering and Technology, Telangana, Hyderabad, India
[2]Department of CSE, Mahatma Gandhi Institute of Technology, Telangana, Hyderabad, India
[3]Department of CSE, Jawaharlal Nehru Technological University, Telangana, Hyderabad, India

RETRACTED

Abstract

The exponential development of online learning resources has led to an information overload problem. Therefore, recommender systems play a crucial role in E-learning to provide learners with personalised course recommendations by automatically identifying their preferences. In addition, e-Learning platforms such as MOOCs and LMS have been criticised for their low course completion rates, and one of the primary reasons is that they do not provide personalised course recommendations for users with varying interests. Rapidly locating the courses that users are interested in on enormous e-Learning platforms can have a significant impact on the quality of learning and the dissemination of knowledge to the learner. This paper examines the most prevalent recommendation techniques utilised in E-learning. We examined how to apply Deep Transformer based Ensembled Attention Model (DTEAM) on e-Learning recommendation system in order to achieve personalized course recommendations. The proposed recommendation model uses BERT as its foundation integrated with MLM and Transformers. Predicted course recommendations are more aligned with the interests of users. Our experimental results proved that traditional recommendation algorithms, such as collaborative filtering and item-based filtering are incapable of producing superior results. The consequence of the research can assist students in selecting courses according to their preferences and improve their learning calibre.

Keywords: Course Recommendation, Deep Transformer based Ensembled Attention Model (DTEAM), BERT, MLM, Transformers

Received on 24 November 2023, accepted on [illegible] November 2023, published on 20 December 2023

Copyright © 2023 A. Madhavi *et al.*, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetel.4470

1 Introduction

Amazon, Myntra, Flipkart, and eBay are just few of the many modern e-commerce platforms from which consumers may choose from a dizzying array of products. Customers have a hard time finding what they're looking for when they have to search for it manually across multiple internet stores. Here, the Recommender System (RS) plays a key role in assisting customers and making suggestions based on their past purchases.[1][2]. E-government, e-business, e-commerce/e-shopping, e-library, e-learning, e-tourism, e-resource services, and e-group activities are just some of the areas where these RS methods have been put to use [3]. Mobile, cloud, social media networks, and traditional PCs are the four environments where these apps are deployed [2].

RS is a filtering algorithm that determines the likelihood that a user will favour a specific item based on their previous interactions with that item [4][5]. As can be shown in Figure 1, RSs have a significant effect on any online business by recommending the most relevant and interesting items from a large database to the user based on the user's stated preferences and interests. Based on how they come up with their suggestions, recommendation methods may be broken down into four categories: Hybrid systems, which combine Collaborative Filtering (CF), Demographic Filter (DF) and Content Based Filtering (CBF) are becoming increasingly popular.

[*] Corresponding author. Email: madhavi_a@vnrvjiet.in

RETRACTED

[6]. In order to anticipate and create the suggestion, the content-based RS constructs a user profile through the analysis of specific item attributes [7][8]. This means the user is more likely to receive recommendations for similar products. Users of the same age, gender, and geographical location are assumed to share similar interests in the Demographic-based approach. As a result, this technique for making suggestions forecasts various products for various demographic profiles. The CF technique has achieved amazing success in terms of accuracy, and it has become one of the most used recommendation systems. The foundation of CF approaches lies in the analysis of ordinal feedback data from previously served users in order to foretell the recommendation. Two processes are used by CF to generate the recommendation: memory-based CF and model-based CF [2]. In order to find people who are most like the Active User (AU), memory-based CF looks at their past ratings [9][8]. Similar users' past ratings are used to make predictions for the suggestion [10]. In model-based CF, models such Bayesian networks, fuzzy algorithms, clustering models, and Genetic Algorithms (GA) are trained using past rating data to make predictions in the recommendation-making process [11, 6]. When more than one form of RS is combined, the resulting system is called a hybrid RS [7][12][13].

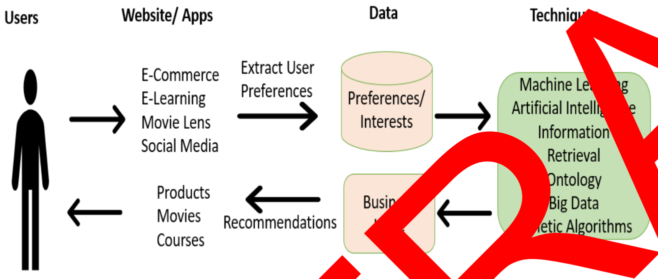


Fig 1: General working scenario of Recommendation System

As the field of ICT has advanced rapidly, the MOOCs platform has become one of the most widely used online educational resources [68]. Personalised course recommendation [14][15][16] based on learner preferences has become the major research endeavour to address these difficulties, as the proliferation of online courses has made it difficult for students to choose those that best suit their needs and interests. CF algorithms [19,20] are used in conventional course recommendation systems [illegible] to uncover latent information about a learner's interests. Using neural recommendation algorithms grounded in deep learning techniques [21,22], these approaches deliver solid results. The Neural Attentive Recommendation Model (NARM) [23] is one such model that can track users' sequential actions and extract their primary goals for taking the class. In addition, the fundamental recommendation model based on attention networks is trained in tandem with the Hierarchical Reinforcement Learning (HRL) model [24], which reduces noisy courses through learner profile model. When consumers have signed up for a wide variety of classes, the course recommendation performances have room to grow. However, HRL fails to deliver satisfactory results since it disregards the user's stated preferences.

The adaptability of the recommendation model allows for a number of methods to be used in assessing learner behaviour [25] on e-learning systems. For instance, Chen et al.'s [26] hybrid recommender model is one such example. It does this by employing item-based CF to unearth groups of pertinent things which are subsequently filtered by a sequential pattern mining algorithm in consideration of commonalities in the sequence of study. Furthermore, Wan and Niu [27] proposed a self-organization-based recommendation model, wherein learning objects are simulated as intelligent objects/entities using the self-organization theory, and the objects in these simulations naturally interact with one another.

Despite their widespread use in course recommendation, the aforementioned approaches all share a critical flaw: they fail to account for students' ever-evolving tastes and requirements as they progress through courses. In other words, these techniques aren't great at extracting the user's choice in every interaction, especially if the learner is interested in a wide variety of subjects and their preferences shift over time. In this scenario, the adaptivity of the recommendation model is low since these techniques are not very good at keeping up with consumers' shifting tastes.

The remainder of the paper is structured as follows: In Section 2, a concise literature review of course recommendation is provided. The proposed technique is then described in detail in Section 3. We conducted experiments using actual data and reported the outcomes in Section 4. Section 5 concludes the document.

2. Related Works

Numerous studies have been conducted on course recommendation in e-Learning platforms. With the exponential growth of ICT, MOOCs, and Learning Management Systems (LMS), online learning resources have exploded. Due to disparities in cognitive ability and preferences, learners are unable to rapidly identify and select the learning resources in which they are interested and required [28]. Therefore, it is imperative to develop an intelligent model that accurately and efficiently recommends useful and engaging learning resources to students.

In this section, we examine extant relevant research in the following fields: (1) Conventional algorithms for course recommendation consist of context-based, content-based, collaborative filtering (CF), and hybrid recommendations. (2) Course recommendations based on Artificial

Intelligence (AI) and Machine Learning include reinforcement learning, sequential recommendation, Deep Learning-based recommendation, Attention Neural Networks, and Graph Neural Network-Based Recommendation.

2.1 Conventional course recommendation algorithms

For content-based course recommendations to work, it is necessary to first obtain a learner vector that displays the learner's preferred courses, then develop an algorithm to process course reviews and obtain a course vector, and finally determine the degree of similarity between the learner vector and the course vector and recommend the courses that are most similar to the learner vector. By omitting superfluous words, Zhang et al. [29] improved their method for determining the degree of similarity between two inquiries. One way they achieved this was by suggesting queries depending on how closely students read a certain piece of text. Based on these associations, Son and Kim [30] developed a novel method of course recommendation. It takes into consideration the various connections between courses by employing clustering techniques from network analysis. Ng et al. [31] employed matrix factorization and content-based algorithms to generate recommendations for children's literature.

2.1.2 CF based Course recommendation method

There is a lot of research on CF recommendation methods in the literature. This study includes both learner-based CF and CF based on learning resources. The learning resource's critical functions to look at the features of each resource, come up with a method to figure out how similar they are, find the most similar resource, and suggest it to the learner who needs it. It predicts the best learning resources based on the ratings and reviews of learning resources in the data set. Learner-based CF is similar in how it works, but it uses learner CF to figure out how similar learners are to each other [32].

2.1.3 Context-based course recommendation method

Classes that are most relevant to the learner's current circumstances are often predicted using context-based recommendation algorithms, which also provide alternative class suggestions. Verbert et al. [33] studied the context-aware recommendation system within the field of technology-enhanced learning. They developed a contextual analysis dimension and framework. They also analysed existing context-aware guidance technologies and proposed future developments. Some specialists have also investigated the technological implementation and use of contextual suggestion.

2.1.4 Hybrid course recommendation method

The term "hybrid recommendation" is used to describe the practise of merging two or more recommendation algorithms; typically, this involves a content-based recommendation algorithm and a CF algorithm. To accomplish individualised course suggestions, Zapata et al. [34] employed LMS metadata information, LRM data and learner characteristics to influence their approach to content filtering, CF, and learner main search. In order to optimise the weight of the course's implicit attributes, Salehi et al. [35] used a genetic algorithm in conjunction with the nearest neighbour CF algorithm to create a learner interest tree from the course's explicit multi-dimensional attributes and the learner's historical rating of the course.

2.2 Artificial Intelligence (AI) and Machine Learning based course recommendation

Since the past ten years, Machine Learning & Artificial Intelligence and recommender systems have been research foci in the field of Computer Science.

2.2.1 Reinforcement Learning (RL)

The use of RL has spread widely across several fields [36-39]. Training compatibility [40,70], dynamic recommendation difficulties [41], and the challenge of artificial agent control [42,43] are all areas that Deep Reinforcement Learning (DRL) can effectively tackle. Investment decision making [44, 45] and stock trading [44, 45] are two areas where Recurrent Reinforcement Learning) (RRL) has been shown to excel. Hierarchal Reinforcement Learning (HRL) [46-48] uses hierarchical assignments or policies to address many issues concurrently. For instance, Zhang () et al. [49] built an HRL-based profile reviser and used the standard recommendation model to train it to filter out irrelevant classes from the user's history. However, they fail to take consumers' changing priorities into account, limiting the model's applicability. The RL algorithm may also be used to make suggestions with an explanation. For example, Wang et al. [50] developed a model-independent RL framework for producing phrase explanations using a customised attention-based neural network that dynamically influences explanation quality. By providing actual pathways with an RL algorithm over a knowledge graph, Xian et al. [51] suggested a reinforcement knowledge graph reasoning technique that combines recommendation and interpretability.

2.2.2 Sequential Recommendation

The goal of sequential recommendations is to predict a user's next behaviour by looking at their previous actions in a series [52]. Markov chains are a useful tool for

RETRACTED

RETRACTED

modelling sequential behaviour [53,54]. One such example is Moling et al.'s [55] channel suggestion model, which uses the user's listening patterns to infer implicit feedback. Methods like knowledge-enhanced Gated Recurrent Units (GRU) [57] and the session-based RNN approach [58] have contributed to the rise in popularity of Recurrent Neural Networks (RNN) for sequential recommendation [56]. Since the learning procedure is analogous to a Markov chain of sequential behaviours, sequential recommendation algorithms [59,60] also perform well in a digital classroom. By combining context awareness, sequential pattern mining, and the CF algorithm, for instance, Tarus et al. [61] proposed a sequential recommendation framework that makes use of the learner's context and sequential access patterns. A hierarchical sequential decision procedure was built by Zhang et al. [16] to improve the sequential recommender system's course suggestions.

2.2.3 Deep Learning-based recommendation

In the field of recommender systems, deep learning technology has made remarkable strides in recent years [69]. When used to the course recommendation algorithm, deep learning technology improves efficiency in processing learning record data, better captures the deep-level characteristics of both learners and courses and helps to prevent data sparseness and cold-start issues. One drawback of deep learning is that it requires a lot of processing power, which might lead to an issue with the findings being hard to understand. modern deep learning recommendation model (DLRM), a specialised parallelization approach that takes advantage of model parallelism on embedding tables to reduce memory requirements. In addition, the completely connected layer is calculated using data parallelism, which significantly boosts performance. NCF model architecture was proposed. Deep learning and the CF recommendation algorithm form the backbone of this strategy.

2.2.4 Attention Neural Networks

Recent years have also seen the proposal of neural attention networks [23,62] to tackle a wide range of challenging problems in deep learning. For the purpose of modelling both changing user behaviours and contextual social impacts, the dynamic graph attention neural network [63] has been designed. Additionally, the dynamic attention-integrated neural network [64] predicts users' shifting preferences and integrates these with other key aspects into a single framework for recommending news articles. These dynamic attention models have a number of problems, one of which is that they take too long to run. For CF methods, Attentive Collaborative Filtering (ACF) [65] is an attention network that integrates with the Bayesian personalised ranking loss to deal with implicit feedback. However, due to the softmax function being used in ACF, item-specific attention weights tend to be somewhat variable. The Neural Attentive Item Similarity (NAIS) model [21] addresses this issue by reducing the dispersion of attention weights by differentiating between the relative contributions of new and old items to a user's preferences.

2.2.5 Graph Neural Network-Based Method

Due to its superior graph structure data learning, Graph Neural Network (GNN) technology has become popular in many fields. Most recommender system data is graph based. Bipartite graph can be used to model user-item interaction data, which is more comprehensible than a matrix. Graph algorithms naturally model user item interaction data. Many recommender systems have used graph algorithms because GNN is good at representation learning and can model user-item bipartite graphs.

In the past decade, recommender systems have advanced rapidly from factorization to models based on deep neural networks. When compared to other recommendation methods, those based on GNNs perform exceptionally well across the board. There are three reasons why GNN-based recommenders are effective: 1. Modeling Complex Relationships, 2. Information Aggregation, 3. Handling Cold Start and Sparse Data. The traditional recommender systems can only draw from a small subset of these data sources, their performance suffers because crucial data is being ignored. GNN standardises data processing by modelling data usage as a network of nodes and edges. The degree to which a user and an item are alike is a major factor in the reliability of recommendation systems. The collaborative filtering effect is only recorded implicitly since most training data consists of interaction records with only closely linked items. Consequently, only first-order connection is significant. The effectiveness of recommendations decreases without high-order connection. High-order connectivity is captured by GNN-based models. Multi-hop graph neighbours make a natural expression of collaboration filtering. Using recommender systems only with the desired behaviours may backfire. GNN-based models can improve recommendation performance by include various off-target activities, such as search and add to cart, thanks to semi-supervised signals over the graph.

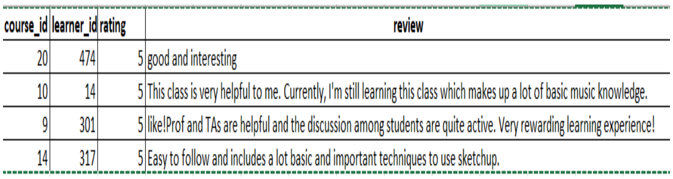
3. Proposed Methodology

The purpose of this research paper is to investigate how to apply the Deep Transformer based Ensembled Attention Model (DTEAM) to a course recommendation system in order to meet the personalised course recommendation requirements of users.

We present a personalised recommendation method that takes the BERT as its foundation integrated MLM and Transformers. This section will outline the process in great depth, focusing on the following points. We begin by discussing the process of gathering and cleaning data.

RETRACTED

Second, we present the model structure of the transformer as a whole as it pertains to the multi-head attention process. At last, we provide classification using soft max layer.



course_id	learner_id	rating	review
20	474	5	good and interesting
10	14	5	This class is very helpful to me. Currently, I'm still learning this class which makes up a lot of basic music knowledge.
9	301	5	like!Prof and TAs are helpful and the discussion among students are quite active. Very rewarding learning experience!
14	317	5	Easy to follow and includes a lot basic and important techniques to use sketchup.

Fig 2: Details about data found in E-Khool dataset

3.1 Data Pre-processing

Deep learning models rely heavily on datasets. After analysing and exploring various popular e-Learning platforms, such as Edx, MIT, and Coursera, we discovered that the e-Learning platforms contain numerous learner groups and extensive learning resources, which is the type of information that is more pertinent to our research objectives. Therefore, we decided to utilise the E-Khool [66] dataset to conduct experiments to validate the model's performance. The E-Khool dataset includes reviews and ratings of the courses as shown in the figure1. These informational contents are the real-world data documented by students on the https://ekhool.com/ website. The majority of the dataset's data is stored in csv format. Consequently, we employ the course and user related dataset information for subsequent experiments.

Due to the presence of noise in the data, we must pre-process the data prior to conducting the experiment. The pre-processing steps are common to those of other research studies. We begin by removing absent, noisy data, such as user information.

In order to clean up the training data, we drop students whose total course load is less than 5. Since there is not enough information on the course to make learning efficient. The data pre-processing approach mitigates the issue of inadvertent suggestion outcomes brought on by the lack of information about the courses taken by the students. Furthermore, in natural language processing, semantic text similarity is a crucial task. Multiple meanings can be attributed to a single word, sentences can vary in length and complexity, and idioms abound in text languages. Bag-of-words and TF-IDF models are two examples of these basic models in NLP technology; nevertheless, they have limitations due to the specificity of phrases, which means that the purpose of the word order is ignored. To better capture the text's inner features, we pre-process the text's course content.

3.2 Model Design

Deep Transformer based Ensembled Attention Model (DTEAM)

BERT integrated with Masked Language Model (MLM) and Transformers are used for model ensemble to get the weighted average of better accuracy. Architecture of proposed Ensemble Attention Model (EAM) is shown in the Figure2.

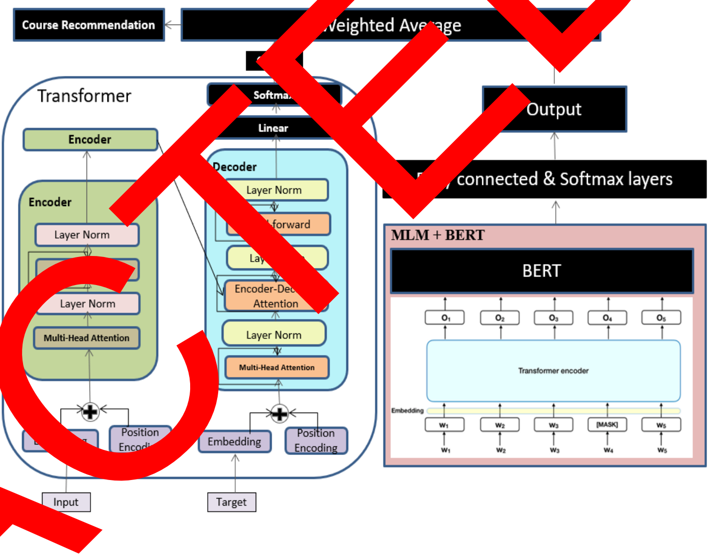


Fig 3: Architecture of proposed Deep Transformer based Ensemble Attention Model (DTEAM)

3.2.1 MLM+BERT

BERT (State of the art natural language model for NLP)

The abbreviation BERT means "Bidirectional Encoder Representations from Transformers." It is widely utilised in the field of language modelling as an attention model. This paradigm offers a multi-stage process for handling textual ambiguity. To begin, BERT is a model that makes use of the attention mechanism by giving different parts of the input data equal weight. This is because it is built on the Transformer architecture. Second, during training, BERT thoroughly absorbs information from both the left and right sides of a word's context. For a more nuanced understanding of sentence context, the model's ability to work in both directions is crucial. Masked Language Model (MLM) is a revolutionary approach that enables bidirectional training, and we applied it.

Working of BERT

BERT employs Transformer, an attention mechanism that discovers contextual relationships between words (or sub-words) in a text. The extraction of textual features from

RETRACTED

course descriptions is a crucial step for enhancing the performance of model recommendations. We use the BERT model [67] to derive features from textual data. Google's BERT can get vectorized versions of course review papers. The BERT model is a language model that is made by an Encoder and Transformer that work both ways. It specifies that the model can get a more accurate picture of a word vector by using information about the word before and after it [69].

To get their feature vectors, the network can pull out the features from the course review text. First, we use masking to give the weightage for words which are significant for prediction and then perform embedding to turn it into a low-dimensional vector with real values that the model can use. Lastly, the trained model is changed over and over again using the user's rating of the course as an input feature vector.

Masked Language Model (MLM)

Before being sent into BERT, over fifteen percent of each word sequence is altered by substituting [MASK] tokens in their place. The model then makes an effort to infer the hidden words' original value from the unmasked words' context. The following steps are necessary for output word prediction:

1. First, the encoder output is layered on top of a classification layer.
2. Second, by multiplying the output vectors by the embedding matrix, the vocabulary dimension is introduced.
3. Each vocabulary terms likelihood is determined using Softmax.

Next Sentence Prediction

When being trained with BERT, the model is given pairs of sentences and taught to recognize whether or not the second sentence is the next sentence in the original text. When training, the second sentence in half of the input pairings is taken directly from the original text, while the other half is chosen at random from the corpus. The hypothesis is that the randomly selected sentence will be completely dissimilar to the source sentence.

In order to train the model to differentiate between the two sentences, the input is handled as follows and it is illustrated in figure 3.

1. The [BEG] token introduces the first phrase, and the [SEP] token closes out the rest.
2. In order to determine if a token fits in Sentence A or Sentence B, we first embed them into a sentence. Sentence embeddings are fundamentally similar to token embeddings with a vocabulary size of 2.
3. Third, a positional embedding is assigned to each token to indicate its place in the sequence.

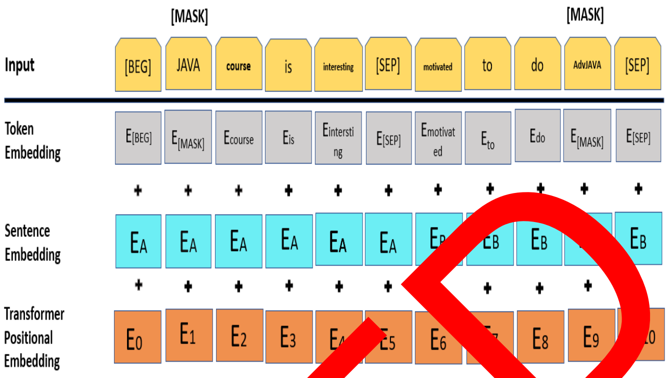


Fig 4. Embedding

The whole input sequence is sent into the transformer model to check if the second phrase is a continuation of the first. The output of the [BEG] token is transformed into a vector using a simple classification layer (learning matrices of weights and biases). The likelihood of the subsequent sequence prediction is computed via Softmax. In order to train the BERT model, we minimise the loss function that is the sum of the losses from Masked LM and Next Sentence Prediction.

3.2.2 Transformer

Since the introduction of transformers in 2017, there has been an explosion in the number of advanced models pertaining to transformer architecture. Transformers have been able to solve the majority of common NLP tasks with the greatest efficacy, and researchers are discovering more and more problems where they can be applied. Transformers are actually developed for sequence-to-sequence actions like machine translation, question answering, etc., it consists of an encoder and a decoder. The encoder accepts embeddings, which are the sum of regular embeddings and positional embeddings.

Input Embedding

Token embedding, positional embedding, and an encoding layer make up the input embedding of our model. The tokens of a sentence are broken down by our model. At the beginning of the token sequence, the special character [BEG] stores the semantic information of the whole input sequence. [SEP] is a special character that signals the end of a sentence sequence. Token i in the sequence is written as ti ϵ RH, where H is the number of hidden levels. When working with sequential data, position embedding is utilised to encode position information. Pi ϵRH is a convenient expression for the positional embedding. Here's the sequence you need to follow. In the high-dimensional space, the input embedding has the same dimension as token embeddings and positional embeddings. Ei ϵ RH is the resultant embedding when these two are multiplied together.

RETRACTED

Encoding

Multi-Head Attention

Multi-Head Attention is essentially composed of a multiple self-attention unit. This is because each attention centre will concentrate on a different aspect of the text. One unit can concentrate on the subject-verb relationship, while another can concentrate on the tense of the text, and so on. This is comparable to employing multiple filters within a single convolutional layer.

The Query, Key, and Value are the three parameters that feed into the Attention layer. W^Q, W^K, and W^V are three matrices of parameters. The inputs of the scaled dot product attention function are the query matrices Q, the key matrices K, and the value matrices V, which are used to calculate the self-attention value as shown in the equation 1.

$$Q, K, V = W^Q E, W^K E, W^V E$$
(1)

During the calculation of self-attention, the multi-head attention mechanism is employed, and Q, K, and V are linearly mapped to h groups in order to derive various H/h-dimensional vectors. The calculation for self-attention is as shown in equation 2

$$Self - Attention(Q^i, K^i, V^i) = Softmax\left[\frac{Q^i, K^{i^T}}{\sqrt{H/h}}\right]$$

(2)

The multi-head mechanism simultaneously executes the self-attention function on h groups of H/h-dimensional Q^i, K^i, and V^i. Each group generates a vector of output results, which are then spliced together as shown in equation 3. The linear transformation is utilized to restore the H-dimensional vector.

$$Multi - head\,(Q, K, V) = Concat\,(attention_1, \ldots, attention_h)W^H \quad (3)$$

Using a feed-forward neural network with two linear mapping functions and a nonlinear ReLU activation function, the output vectors of the multi-head self-attention operations are combined with the self-attention input X for layer normalization as the input. The feed-forward operation (equation 5) and the layer normalization operation (equation 4) can be written as follows.

$$Layer - normalization(X) = LayerNorm(X + multi - head(X))$$
(4)

$$F = f1\,(ReLU\,(f2(layer - normalization(X))))$$
(5)

Then, the input of the feedforward neural network is combined with the output F from the layer normalizer to form the input of the subsequent encoder. In our concept, L represents the total number of transformer encoders. More syntactic and semantic information about a phrase sequence can be obtained through the use of a multiple transformer encoder structure.

Decoding

The target sequence is fed to the Decoder stack's Output Embedding and Position Encoding, which generates an encoded representation for each word in the target sequence that encapsulates the word's meaning and position. This information is supplied to all three parameters, Query, Key, and Value, in the Self-Attention in the first Decoder, which generates an encoded representation for each word in the target sequence, which now includes the attention scores for each word. This is sent to the Encoder-Decoder's Query parameter after being processed by the Layer Norm applying attention in the Initial Decoder

Encoder-Decoder Attention

In addition, the Value and Key parameters in the Encoder-Decoder Attention receive the results from the last Encoder on the stack. Therefore, both the goal sequence (from the Decoder Self-Attention) and the input sequence (from the Encoder stack) are being sent into the Encoder-Decoder Attention. As a result, it generates a representation that incorporates both the input sequence's and the target sequence's attention scores for each sequence word. As it makes its way down the stack of decoders, the attention scores from both the encoder and the decoder are incorporated into the final word representation.

Classification

In order to transform the category label distribution into a probability distribution, we employ the softmax nonlinear activation function. The most likely intent label is the one that corresponds to the highest probability value. Intent label prediction is computed as follows:

$$label = arg\,max\,(Softmax\,(I))$$
(6)

Ensembling

A weighted average is an ensemble technique in which each model's contribution to the final prediction is weighted by the model's performance. When using model averaging, all members of the ensemble have an equal impact on the final prediction. The ensemble prediction in our proposed model has produced the final prediction by applying weighted average of muti-head attention-based transformer and MLM based BERT model.

RETRACTED

4. Results

In our experiments, we validate the translation personalised course recommendation model based on the BERT model using a publicly available dataset.

4.1. Experimental setup

4.1.1. Dataset description

This paper uses the E-khool learning platform dataset [66]. This dataset consists of a single file with one lakh rows, representing over 25 courses and 1,000 students. Here, Course ID, Date of Subscription, Learner ID, Ratings ranging from 1 to 5, Date of Ratings and Review, and Ratings ranging from 1 to 5 are specified.

4.1.2 Performance metrics

The adopted performance metrics are precision, recall, and F1-score. The respective definitions are as follows.

Precision: It is the ratio of true positives to the total number of positives. True positives and false positives are included in the total positives. The precision measure is written as follows:

$$\chi = \frac{Pos_{true}}{Pos_{true} + Pos_{false}}$$

(7)

where, χ is precision, Pos_{true} is true positives, and Pos_{false} signifies false positives.

Recall: This measure ratio of true positives to total of false negatives and true positives and the formula is given by,

$$\delta = \frac{Pos_{true}}{Pos_{true} + Neg_{false}}$$

(8)

where recall is signified as δ, and Neg_{false} symbolizes false negatives.

f-measure: This represents the average harmonic value between precision and recall. This also indicates a weighted measure of both recall and precision, which is presented as follows:

$$f_m = 2 \times \left(\frac{\chi \times \delta}{\chi + \delta} \right)$$

(9)

where, f_m denotes f-measure

Table 1

Model	Precision@5	Recall@5	F1 Score @5	Precision @10	Recall @10	F1 Score @10
RNN	14.35	11.34	12.67	12.75	17.85	14.88
LSTM	15.17	11.42	13.03	11.39	[illegible]	[illegible]
BERT	20.73	23.54	22.05	17.25	[illegible]	[illegible]
DARL	63.12	48.53	55.64	49.48	54.51	[illegible]
HRL	81.3	56	90.78	79.2	67	80.67
CNN-NegSeq	45	52	48	[illegible]	[illegible]	66
NoR-MOOCs	68	89	76	62	69	[illegible]
Proposed	**80.25**	**79**	**85.73**	**[illegible]**	**[illegible]**	**83.20**

4.1.3. Performance analysis

We evaluated and validated the experimental results compared to the baseline model to ensure the proposed strategy was effective. We use @k to denote the best k courses to take. Table 1 compares the proposed method with a number of other models, such as RNN, Long Short-Term Memory (LSTM), BERT, DARL, HRL, CNN-NegSeq, and NoR-MOOCs, as well as the classic recommendation methods User and Item-based Collaborative Filtering. Table 1 shows that our approach produces the best outcomes, proving the viability of the individualised course suggestion strategy put forward in this study. Meanwhile, it's obvious that common recommendation algorithms can't deliver top-notch outcomes, such as collaborative filtering and item-based filtering. This is mostly because there is room for improvement in the model's feature extraction capabilities, which means that changes in course materials are not fully accounted for.

The proposed methodology has been evaluated for 5 and 10 courses. The basic recurrent neural network models have shown poor performance than the recent transformer-based models. The proposed method has outperformed the existing methods for recommending top 10 courses. However, for precision @5 [R5] has shown slight better result than ensemble model, when recommendation of number of courses increased the accuracy has been reduced. The proposed methodology is able to persist the accuracy even after increasing the number of recommended courses. The graphical analysis of performance has been shown in the below Figure 5 and 6.

RETRACTED

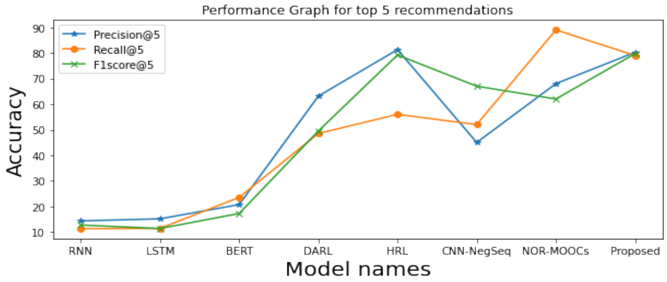


Fig 5: Proposed Model performance comparison for top 5 recommendations

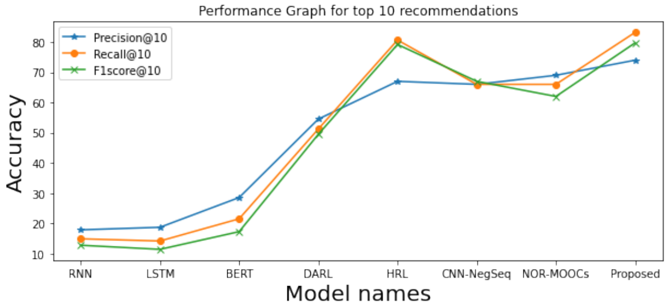


Fig 6: Proposed Model performance comparison for top 10 recommendations

5. Conclusion and Future scope

We experimented by using BERT, Multi-head attention transformers, and Masked language model to make individualised course recommendations. This approach is made for MOOCs, or massive open online courses, and it can tailor course suggestions to each user. To combat the inaccuracy of current recommendation systems, the suggested approach blends bidirectional encoder representations from the transformers model with the attention mechanism. The following are the main stages of our methodology. The dataset's collection and initial processing are presented initially. Second, the transformer model's bidirectional encoder representations are included into the framework for an ensembled recommendation model, which also makes use of a multi-head attention mechanism. The experimental outcomes prove the efficacy of the suggested framework for individualised suggestions. The scalability of the suggested technique allows for the elimination of the drawbacks of traditional large-scale trials.

Acknowledgements.
The authors send gratitude to the university for the facilities and support provided during the research.

References

[1] Alhijawi, B., & Kilani, Y. (2016). Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems. Proceedings of the IEEE/ACIS 15th international conference on computer and information science (ICIS). IEEEhttps://doi.org/10.1109/icis.2016.7550751.

[2] Alhijawi, B., & Kilani, Y. (2020). The recommender system: A survey. International Journal of Advanced Intelligence Paradigms, 15(3), 229–251. https://doi.org/10.1504/IJAIP.2020.105815.

[3] Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. User Modeling and User-Adapted Interaction, 25(2), 99–154. https://doi.org/10.1007/s11257-015-9155-5

[4] Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. Recommender systems handbook. Springer1–35.

[5] Sharma, L., & Gera, A. (2013). A survey of recommendation system: Research challenges. International Journal of Engineering Trends and Technology (IJETT), 4(5), 1989–1992.

[6] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012.

[7] Alhijawi, B., Obeid, N., Awajan, A., & Tedmori, S. (2018). Improving collaborative filtering recommender systems using semantic information. Proceedings of the 9th international conference on information and communication systems (ICICS). IEEEhttps://doi.org/10.1109/iacs.2018.8355454.

[8] Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. User Modeling and User-Adapted Interaction, 25(2), 99–154. https://doi.org/10.1007/s11257-015-9155-5

Alhijawi, B. (2019a). Improving collaborative filtering recommender system results and performance using satisfaction degree and emotions of users. Web Intelligence, 17(3), 229241. https://doi.org/10.3233/WEB-190415.

[9] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012.

[10] Alhijawi, B. (2019b). Improving collaborative filtering recommender system results using optimization technique. Proceedings of the 3rd international conference on advances in artificial intelligence. ACMhttps://doi.org/10.1145/3369114.3369126.

[11] Gao, L., & Li, C. (2008). Hybrid personalized recommended model based on genetic algorithm. Proceedings of the 4th International conference on wireless communications, networking and mobile computing. IEEEhttps://doi.org/10.1109/wicom.2008.2152.

[12] Ho, Y., Fong, S., & Yan, Z. (2007). A hybrid ga-based collaborative filtering model for online recommenders.,. (pp. 200–203)

[13] H. Zhang, T. Huang, Z. Lv, S. Liu, Z. Zhou, MCRS: A course recommendation system for MOOCs, Multimedia Tools Appl. 77 (6) (2018) 7051–7069.

[14] P. Chang, C. Lin, M. Chen, A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems, Algorithms 9 (3) (2016) 47.

[15] M.E. Ibrahim, Y. Yang, D. Ndzi, Using ontology for personalised course recommendation applications, in: Proceedings of the 17th International Conference on Computational Science and Its Applications, 2017, pp. 426–438.

[16] F. Bousbahi, H. Chorfi, MOOC-Rec: A case based recommender system for MOOCs, Procedia Social Behav. Sci. 195 (3) (2015) 1813–1822.

RETRACTED

[17] X. Jing, J. Tang, Guess you like: course recommendation in moocs, in: Proceedings of the International Conference on Web Intelligence, 2017, pp. 783–789

[18] X. Zhou, S. Wu, Rating LDA model for collaborative filtering, Knowl.-Based Syst. 110 (2016) 135–143.

[19] S. Kabbur, X. Ning, G. Karypis, FISM: factored item similarity models for top-N recommender systems, in: Proc. ACM SIGKDD Conf., 2013, pp. 659–667.

[20] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, T.-S. Chua, NAIS: Neural attentive item similarity model for recommendation, IEEE Trans. Knowl. Data Eng. 30 (12) (2018) 2354–2366.

[21] X. Li, X. Li, J. Tang, T. Wang, Y. Zhang, H. Chen, Improving deep itembased collaborative filtering with Bayesian personalized ranking for MOOC course recommendation, in: Proceedings of International Conference on Knowledge Science, Engineering and Management, 2020, pp. 247–258.

[22] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1419–1428.

[23] J. Zhang, B. Hao, B. Chen, C. Li, H. Chen, J. Sun, Hierarchical Reinforcement Learning for Course Recommendation in MOOCs, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 435–442

[24] D. Wu, J. Lu, G. Zhang, A fuzzy tree matching-based personalized e-learning recommender system, IEEE Trans. Fuzzy Syst. 23 (6) (2015) 2412–2426.

[25] W. Chen, Z. Niu, X. Zhao, Y. Li, A hybrid recommendation algorithm adapted in e-learning environments, World Wide Web 17 (2) (2014) 271–284.

[26] S. Wan, Z. Niu, An e-learning recommendation approach based on the self-organization of learning resources, Knowl.-Based Syst. 160 (2018) 71–87

[27] J. Zhang, F. Gu, Y. J. Ji, and J. F. Guo, "Personalized scientific and technological literature resources recommendation based on deep learning," Journal of Intelligent Fuzzy Systems, vol. 41, no. 2, pp. 29[illegible]–2996, 2021.

[28] Zhang B, Yang FZ, Fan LN. Query recommendation based on document content user focuses on. Appl Mech Mater 2014;511:385–8

[29] Son J, Kim SB. Content-based filtering for recommendation systems using multiattribute networks[J]. Expert Syst Appl 2017;89:404–[illegible]

[30] Ng YK. CBRec: a book recommendation system for children using the matrix factorisation and content-based filtering approaches[J]. Int J Bus Intell Data Min 2020;[illegible]:129–49.

[31] Sakboonyarat S, Tantatsanawong P. Massive open online courses (MOOCs) recommendation modeling using deep learning[C]//In: Proceedings of the 2019 23rd international computer science and engineering conference (ICSEC). IEEE; 2019. p. 275–80

[32] Verbert K, Manouselis N, Ochoa X. Context-aware recommender systems for learning: a survey and future challenges[J]. IEEE Trans Learn Technol 2012;5(4): 318–35

[33] Zapata A, Menendez VH, Prieto ME. A hybrid recommender method for learning objects [J]. In: IJCA Proceedings on design and evaluation of digital content for education (DEDCE). 1; 2011. p. 1–7.

[34] Salehi M, Kamalabadi IN, Ghoushchi MBG. An effective recommendation framework for personal learning environments using a learner preference tree and a GA[J]. IEEE Trans Learn Technol 2013;6(4):350–63.

[35] M.L. Littman, Reinforcement learning improves behaviour from evaluative feedback, Nature 521 (2015) 445–451.

[36] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science 362 (6419) (2018) 1140–1144.

[37] A. Kara, I. Dogan, Reinforcement learning approaches for specifying ordering policies of perishable inventory systems, Expert Syst. Appl. 91 (2018) 150–158.

[38] X. Wang, Y. Wang, D. Hsu, Y. Wang, Exploration in interactive personalized music recommendation: A reinforcement learning approach, ACM Trans. Multimed. Comput. Comm. Appl. 11 (1) (2014)[illegible]

[39] F. Liu, X. Li, H. Guo, R. Tang, Y. Ye, X. He, End-to-end deep reinforcement learning based recommendation with supervised embedding, in: Proceedings of the 13th ACM International Conference on Web Search and Data Mining, 2020, pp. 384–392.

[40] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N.J. Yuan, X. Xie, Z.J. Li, DRN: A deep reinforcement learning framework for news recommendation, in: Proceedings of the 27th International Conference on World Wide Web, 2018, pp. [illegible]76

[41] [43] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[42] J.K. Gupta, M. Egorov, M. Kochenderfer, Cooperative multi-agent control using deep reinforcement learning, in: G. Sukthankar, J. Rodriguez Aguilar (Eds.), Autonomous Agents and Multiagent Systems, Springer, Cham, 2017, pp. 66–83

[43] S. Almahdi, S.Y. Yang, An adaptive portfolio trading system: A riskreturn portfolio optimization using recurrent reinforcement learning with expected maximum drawdown, Expert Syst. Appl. 87 (2017) 267–279.

[44] P. Gabrielsson, U. Johansson, High-frequency equity index futures trading using recurrent reinforcement learning with candlesticks, in: IEEE Symposium Series on Computational Intelligence, 2015, pp. 734–741.

[45] P. Basile, C. Greco, A. Suglia, G. Semeraro, Deep learning and hierarchical reinforcement learning for modeling a conversational recommender system, Intell. Artif. 12 (2) (2018) 125–141.

[46] O. Nachum, S.S. Gu, H. Lee, S. Levine, Data-efficient hierarchical reinforcement learning, in: Advances in Neural Information Processing Systems, 2018, pp. 3307–3317.

[47] A.S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, K. Kavukcuoglu, FeUdal networks for hierarchical reinforcement learning, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3540–3549.

[48] J. Zhang, B. Hao, B. Chen, C. Li, H. Chen, J. Sun, Hierarchical Reinforcement Learning for Course Recommendation in MOOCs, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 435–442.

[49] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, X. Xie, A reinforcement learning framework for explainable recommendation, in: Proceedings of IEEE International Conference on Data Mining, 2018, pp. 587–596.

[50] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, Y. Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 285–294.

[51] H. Fang, D. Zhang, Y. Shu, G. Guo, Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations, ACM Trans. Inf. Syst. 39 (1) (2021) 1–42.

[52] J. Tang, K. Wang, Personalized top-N sequential recommendation via convolutional sequence embedding,

RETRACTED

in: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018, pp. 565–573.

[53] X. Xin, A. Karatzoglou, I. Arapakis, J.M. Jose, Self-supervised reinforcement learning for recommender systems, in: Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 931–940.

[54] O. Moling, L. Baltrunas, F. Ricci, Optimal radio channel recommendations with explicit and implicit feedback, in: Proceedings of the 6th ACM Conference on Recommender Systems, 2012, pp. 75–82.

[55] L. Wang, W. Zhang, X. He, H. Zha, Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation, in: Proc. ACM SIGKDD Conf., 2018, pp. 2447–2456.

[56] J. Huang, W.X. Zhao, H. Dou, J.-R. Wen, E.Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, in: Proceedings of the 41nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 505–514.

[57] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, in: Proceedings of the International Conference on Learning Representations, 2016.

[58] P. Wang, Y. Fan, L. Xia, W.X. Zhao, S. Niu, J. Huang, KERL: A knowledgeguided reinforcement learning model for sequential recommendation, in: Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 209–218.

[59] J.K. Tarus, Z. Niu, A. Yousif, A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining, Future Gener. Comput. Syst. [illegible] (2017) 37–48.

[60] J.K. Tarus, Z. Niu, D. Kalui, A hybrid recommender system for e-learning based on context awareness and sequential pattern mining, Soft Comput. 22 (8) (2018) 2449–2461.

[61] X. Zhang, S. Li, L. Sha, H. Wang, Attentive interactive neural networks for answer selection in community question answering, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3525–3531.

[62] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 555–563.

[63] L. Zhang, P. Liu, J.A. Gulla, Dynamic attention-integrated neural network for session-based news recommendation, Mach. Learn. 108 (10) (2019) 1851–1875.

[64] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T.-S. Chua, Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 335–344.

[65] E-khool learning platforms dataset, https://ekhool.com/, accessed on November 2022

[66] Devlin J., Chang M.W., Lee K., Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018: 1–8

[67] Madhavi, A.; Nagesh, A.; Govardhan, A. A Study on E-Learning and Recommendation System. Recent Adv. Comput. Sci. Commun. Former. Recent Pat. Comput. Sci. 2022, 15, 748–764.

[68] Bifeng Li, Gangfeng Li, Jingxiu Xu, Xueguang Li, Xiaoyan Liu, Mei Wang, Jianhui Lv. "A personalized recommendation framework based on MOOC system integrating deep learning and big data" , Computers and Electrical Engineering, 2023,Elsevier Publishers.

[69] Yuanguo Lin, Shibo Feng, Fan Lin, Wenhua Zeng, Yong Liu, Pengcheng Wu. "Adaptive course recommendation in MOOCs" , Knowledge-Based Systems, 2021, Elsevier Publishers

[70] RAO, P.S., NAGINI, S., Sudheer, D., BAPIRAJ, V., VARDHAN, M.V. and HARSHITHA, [illegible] 2022. STUDENT PERFORMANCE ANALYSIS FOR OUTCOME BASED EDUCATION. International Journal of Early Childhood Special Education, 14(5).

[71] Devulapalli, S., Venkatesh, B. and [illegible]ula, R., 2023. Business Analysis During the Pandemic Crisis Using Deep Learning Models. In AI-Driven Intelligent Models for Business Excellence (pp. [illegible]-80). IGI Global.

.