International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Application-Oriented Reviews and Products of Parallel Deep Neural Networks in Recommendation Systems

Kanchan Sitaram Pradhan*[1], Dr. Khan Vajid Nabilal[2], P. N. Varalakshmi K.[3], Dr. Pavitar Parkash Singh[4], Jagadish S. Kallimani[5], Dr. Asma A. Shaikh[6]

Submitted: 08/05/2023 **Revised**: 15/07/2023 **Accepted**: 04/08/2023

Abstract: Data scarcity is a problem for recommender systems, and cross-domain recommendation has been shown to be an effective solution. Previously, this strategy has worked well. We've done some work in the past to improve Deep Learning IA-CNN-based Personalized Product Recommendation Systems using Hybrid Deep Learning-based Hybrid Techniques. An approach to cross-domain deep neural networks is presented in this paper, which utilizes both deep neural networks and cross-domain recommendations (CD-DNN). With the help of user reviews and product metadata, CD-DNN can accurately predict ratings for many different kinds of products. If you want CD-DNN to be able to accurately predict future ratings of users or items in your target domain and other source domains, you'll need to train it on data from both of those places. By maximizing the accuracy of prediction predictions, several parallel neural networks are trained to learn the latent factors for both user features and item features, as well. When CD-DNN and other domain features are combined, a single latent space mapping for user attributes is produced. By doing so, the network's users benefit from improved performance. The proposed CD-DNN outperforms other current recommendation approaches, and it also addresses data scarcity by incorporating data from a variety of sources, including the Amazon datasets.

***Keywords*:** *Convolutional Neural Networks, Cross-Domain Recommendation, Rating Prediction, CD-DNN*

1. Introduction

In the information age, we face a variety of challenges, including digital overload. Data collected from various sources can be used to help consumers find the products and services they recommend. Recommender systems can send other customers recommendations based on the regular activities of their users, such as music listening, public transportation route planning, and online shopping. Predicting a customer's rating of a product is a significant challenge for the recommender system to overcome. Collaborative filtering is used in the vast majority of today's recommender systems [1–5]. The underlying principle is that in the future, important decisions will be made by people who share similar values. Collaboration filtering is currently the most popular recommendation method, because it relies on primary customer sources. There are a lot of problems with data partitioning because there are so few customer ratings for products. There is also the "cold start problem," which makes it difficult to make recommendations for the most recent users or items because there are no historical records.

In addition, content-based recommendation is a popular approach that looks for items with similar attributes based on what the user has previously enjoyed [6–8]. According to the example above, the process may recommend Mito Anna if Mj is a popular film with similar themes (for example, adventure). If there are similarities between Anna and Brown, the system may also recommend things that they both liked (such as being of the same age, being of the same gender, and having the same educational background). Because the method extracts features from personal preferences and items without the need for a large number of review records, there is almost no data scarcity when using content-based recommendations. There are no more gaps in data because of this. Consequently, it is possible that recommendations generated for new customers may be inaccurate due to difficulties in obtaining user profiles in this situation. Collaborative filtering with metadata has become a popular solution to the data scarcity problem because content information, such as evaluations or item metadata, is typically associated with users and items. The addition of this data has improved the accuracy of recommendations based on user data sets, addressing the issue of data scarcity while

[1] *Assistant Professor, Dept of Computer Engineering, MIT Academy of Engineering, Alandi, Pune. Email: kanchan.dhote@mitaoe.ac.in*
[2] *Associate Professor, Dept of Computer Engg, KJEI's KJ College of Engineering and Management Research,Pune.*
Email: kvajid12@gmail.com
[3] *Research Scholar, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India, and Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India*
Email: pnvaralakshmik@gmail.com
[4] *Professor of Management, Lovely Professional University.*
Email: pavitar.19476@lpu.co.in, Orcid Id: 0000-0001-9334-1952
[5] *Professor and Head, Department of Artificial Intelligence and Machine Learning, M S Ramaiah Institute of Technology, Bangalore, India*
Email: jagadish.k@msrit.edu.
[6] *Assistant Professor, Marathwada Mitra Mandal's College of Engineering, Pune. Email: asma.ayyaj.shaikh@gmail.com*
Orcid id: 0000-0002-2474-8537
* *Corresponding Author Email: kanchan.dhote@mitaoe.ac.in*

also improving the accuracy of those recommendations. Deep neural networks [11, 12] as well as topic modelling [9, 10] have been used to propose the inclusion of content information in the algorithm. Cross-domain recommendations [13] are another option. These are recommendations that come from one domain to another. User preferences and product properties can be gleaned from other industries and used for more personalised and reasonable services. Most commonly, it's used to suggest related videos and online exercises to students who have bought study books, as well as to suggest hotels in accordance with travellers' preferred vacation spots. Thus, a substantial amount of information about user behaviour has been gathered in the mature domain as a result of this. To address the recommender system's scarcity and cold start issues as well as to significantly improve user satisfaction and experience, it is possible to share and supplement information from different domains.

With this new cross-domain recommendation system, we've been able to produce better results than ever before by fusing content information with cross-domain data. By combining user reviews and item metadata, a cross-domain deep neural network (CD-DNN) was created to solve rating prediction problems. This network can then be used to predict ratings (CD-DNN). Users and items from various domains can simultaneously share a common feature space created by parallel neural networks when using CD-DNN to learn latent factors. Users' reviews inform one model of a user feature, while item metadata from various domains informs another model of an item's properties. Both models can be found online. This layer is built on top of all neural networks, and it is used to predict each item's rating based on a combination of the user and the item's features. An accurate prediction can be made by using this method. Users and objects' latent features are linked in this layer using matrix factorization techniques [14], which were used as inspiration for the top layer.

We tested the proposed CD-DNN on four Amazon public datasets to ensure its effectiveness. CD-DNN outperforms current models and can alleviate data scarcity by using more information from multiple domains at the same time, according to the results of the experiments.

The following is a rundown of some of the project's most significant accomplishments. It's possible to model users and items collaboratively in a CD-DNN, which provides cross-domain recommendation, by using reviews and item metadata. More data from a broader range of sources is being used to address the issue of sparsity than ever before. (3) Our model outperformed leading models by a significant margin in experiments conducted on four publicly available datasets.

Here, you will find a breakdown of how this paper is going to be structured in the future. Text-based and cross-domain recommendation, as well as their advantages and disadvantages, and how they can be combined, are discussed in Section II of this paper. Section III is where we craft the research questions that will direct our efforts. We go into great detail in Section IV about the CD-DNN. CD-DNN model is examined and compared to current recommender system models and other recommender system models in Section V to show the model's usefulness and effectiveness. Section V: Section VI, on the other hand, marks the successful conclusion of the project.

2. Related Work

Deep learning-based recommendation systems, Textual recommendation systems and cross-domain recommendations are all discussed in this section.

i) Recommender Systems using Text Data

The problem of data scarcity in recommender system applications can be solved by incorporating information from text data. Collaboration filtering features were used to extract opinion from movie review text using several methods, according to a citation [15]. Methods used included: Recommendation systems for movies could gain an advantage by mining user reviews for movie opinions. There are distinct manual interactions for each method, despite this.

Techniques for uncovering hidden topics in review text have been employed by a large number of researchers. Latent rating dimensions and latent review topics were used in the study, according to the reference [9]. It was possible to extract highly understandable text labels for latent rating dimensions, as well as more accurate predictions of item scores, by mining review text. [16] Describes a matrix factorization model called TopicMF, which incorporates subjective ratings and review text. Using a traditional matrix factorization method in conjunction with a topic model, we were able to uncover the review text's hidden themes and make this conclusion. A probabilistic model presented in [17] uses collaboration filtering and topic modulation to improve search results. The distributions of user and topical preferences for movies, as well as the distributions of those preferences themselves, can be obtained.

In order to create a unified model that incorporates ratings and user reviews as well as other data, we used content-based recommendation and collaborative filtering. [18] goes into great detail about this design. We were able to uncover previously unknown review text topics and align them with rating dimensions using this method, which greatly improved our ability to predict review content in the future. Social matrix factorization and topic matrix factorization techniques were used by researchers at Reference [10] to evaluate the usefulness of data fusion in recommendation formulations. This was accomplished by using a random walk and an integrated latent topic model, and other methods, the authors of [19] claim to have developed an empirically supported probabilistic approach to making recommendations. A bag-of-words model, rather than the actual word order of the text, has frequently been overlooked in previous studies because of this. However, the order in which words are presented in the review is crucial. For the purpose of extracting text features, a convolutional neural network is used, and word embedding is used to build word vectors.

ii) Deep Learning Based Recommender Systems

When it comes to developing recommender systems, deep learning has recently been proposed as a method. Using a user-item rating matrix and denoising auto-encoders or restricted Boltzmann machines, some evidence suggests that a user and item profile can be generated. The neural network-based matrix factorization model from [12] was implemented in Matlab. As part of its development, programming was used in the implementation. Developing a user-item matrix that included both overt and covert comments from the user community was the first step in this process that needed to be completed. Using this matrix, a deep neural network could be trained to learn a low-dimensional embedding space that could be used for both people and things. People and things can both fit in here. A binary cross-entropy loss function was used to improve training results. [11] Proposes convolutional sequence embedding recommendations for top-N sequential recommendations based on convolutional sequence embedding that embeds a recent item as a "image" in time and latent spaces. Top-N sequential recommendations are based on convolutional sequence embedding. Wide and Deep Learning was proposed by [23] as a method for making recommendations. Because it utilized both deep neural networks and wide linear models, this approach provided advantages in terms of memory and generalization for the model. The number of applications purchased went up when only wide and deep models were used. The Deep and Cross network was proposed by [24] to predict click-through rate in machine learning. Improved learning methods for bounded-degree feature interactions are also part of the package, making it more effective than previous approaches.

Collaborative filtering refers to all of the methods discussed above that do not take into account text data and instead rely solely on rating information or user-item behavior interactions (such as clicks).

Similar to extraction of information from text data for recommendation purposes deep learning techniques are employed here. Incorporating queries and documents into a deeply structured semantic model, they devised an algorithm for determining whether or not a document is relevant to a particular query. To provide personalized recommendations to users, a similar method has been proposed [26]. A latent feature space was created by mapping user and item profiles to one another, ushering in an unprecedented level of semantic relevance between users and their intended purchases. To help with hash tag recommendations for tweets with both text and visual content, the authors of this study devised an attention sharing (co-awareness) network. The authors of [25, 26], and 27] all used neural networks to extract text information from documents, whereas the authors of [27] used an LSTM-based model to do so. Text features are extracted using convolutional neural networks in this study, which are then used to refine the text.

However, our proposed model differs from the previous models in a few ways. Convolutional neural networks extract text features for recommendation purposes as well as [28, 29]. It is possible to mine item features and user preferences at the same time by using review text, according to [28]. This method saves time and effort. It's a review-level recommendation. According to [29], the authors' Neural Attention Regression model explains their findings. I showed them how an attention mechanism can be useful in reviews and how they could put this knowledge to use. This model or the other ignores information about a product's metadata, which includes information about its specific characteristics and features. Their analysis is based on data from a single domain, so it's important to point that out. Data scarcity is reduced thanks to our model's capability to learn user features from a variety of domains.

iii) Cross-Domain Recommendation

Recommendation engines that use data from other fields to make recommendations can be built [13, 30]. Cross-Domain Topic Learning (CDTL) is a model developed by researchers to address issues such as lack of connectivity, complementary expertise, and skew ness in the topics covered by researchers. Latent factor models based on clustering can improve the performance of cross-domain recommendation, according to [31]. "[Citation is needed] Finding patterns across domains and figuring out how much information to share is easy with this tool. Using this tool, you can learn about the user rating patterns for each domain you're interested in. Machine learning is used to accomplish this. A multi-view deep learning model for recommender systems was also proposed. This model can be used in a wide range of situations and for a wide variety of problems. There are a few key differences between our and their systems, despite the fact that the overall architecture of both systems appears to be similar at first glance. Customers' reviews, which typically contain a wealth of information about their preferences, are ignored in favor of only considering the product's metadata. A convolutional neural network, which is used to learn user and item characteristics, is used to extract text features from the input text. Serendipitous product recommendations can be made with the help of a newly developed method [33]. The term "transfer learning transfer learning recommendation" was coined by the authors of [34] to describe their neural network-based approach to cross-domain recommendation. An additional proposal was made known as a "transfer of learning recommendation." A machine learning framework was used to successfully implement the first cross-domain transferable bandit policy [35].

Research shows that this approach has the potential to facilitate the transfer of prior knowledge from source recommender systems to target recommender systems by speeding up exploration and aiding in exploration. As a way to achieve a better balance between different domains, [36]'s authors proposed this model. Users' privacy was protected during data sharing because the authors of [37] did not divulge any information about the user's habits or activities. A new method called Neural Attentive Transfer Recommendation has been developed by researchers in order to extract useful signals from transferred item factors (NATR). In comparison to other cross-domain recommendation methods, we have developed a CD-DNN that extracts more precise user features from review text. As far as we can tell, there was only one study published in 2001 that examined the use of review text to improve cross-domain recommendation accuracy. Recommendations are made using both item reviews and article titles to facilitate cross-domain recommendation, according to the authors' Transfer Meeting Hybrid model. To extract semantic information from unstructured text, this model made use of a Java memory module and a Java-implemented transfer network.

They proposed an algorithm that relied solely on information from a single source, rather than considering both user reviews and item metadata. Our model's data does not have to come from a single place; rather, it can come from a variety of places.

3. Problem Formulation

Over the years, a variety of contexts and applications have investigated the problem of cross-domain recommendation. All the user's choices are taken into account when building his or her profile. To address the issues of cold start and data scarcity, recommender systems [13] have been proposed [14]. Using cross-domain recommendation technology, you can move data from one field to another easily and securely. Both improvement-targeted domains and knowledge-resource domains can be distinguished in the system. For an unknown number of users and a target domain with unknown number of items, we can produce a $R^t \varepsilon R^{m\times n}$ rating matrix with uncertain rating values denoted by the symbols r_{ui} at each of its entries. N-tuples make up the review dataset D^t. Tuples (*u*, *i*, r_{ui}, w_{ui}) represent user reviews of an item with a rating and review text for an item *i*, and each represents one review of an item *i* by one person. There are multiple items in M^t, each of which represents a metadata document, and each of these documents is represented by one of the entries mi in M^t.

There are rating matrices R^s, review datasets D^s, and metadata sets M^s in every source domain. You should treat the users of each source domain as if they are part of your intended audience. For cross-domain recommendation, a problem definition like this must first be formulated once the data has been inserted into both the source and target domains.

For this analysis,

Input: we need the partially observable rating matrices $\{R^{s1}, R^{s2},\ldots,R^{sN}\}$, review datasets $\{D^{s1}, D^{s2},\ldots,D^{sN}\}$ and metadata sets $\{M^{s1}, M^{s2},\ldots,M^{sN}\}$ from the source domains as well as their respective review datasets and metadata sets.

Output: It is clear that in order to correctly predict *u* and *i* from only two parameters, the model must be capable of doing so. That is to say, the model must be capable of predicting *u* and *i* using only the inputs *u* and *i* as input.

$$\hat{r}^t_{u,i}=f(R^t, D^t, M^t, \{R^{s1}, R^{s2},\ldots,R^{sN}\},\{D^{s1}, D^{s2},\ldots,D^{sN}\},\{M^{s1}, M^{s2},\ldots,M^{sN}\}) \qquad (1)$$

It is possible to make recommendations to users with the help of this algorithm. Finally, All of the reviews written by a single person in our target domain are analyzed, and the results of our prediction scheme are used to suggest products that are most likely to be of interest based on those results. All of these suggestions are based on what other people have said about that particular reviewer. In Table 1, you will find a list of all the mathematical terms used in this article.

Symbols	Definitions and Descriptions
$d^u_{1:l}$	user or item u's text consisting of l words
$V^u_{1:l}$	word vectors of user or item u
c	the dimension of word embedding
h	the window size of convolutional kernel
$\mathbf{w}_c$	the kernel in the convolutional layer
g_i	the i-th feature map in the convolutional layer
b_c	the bias of the convolutional kernel
$\mathbf{w}_f$	the weight matrix of the fully connected layer
b_f	the bias of the fully connected kernel
$\mathbf{x}_u$	the output of Net_u
$\mathbf{y}_i$	the output of Net_i
n_c	the number of convolutional kernels
α	the learning rate

Table 1. Definitions and Descriptions of Symbols

4. Proposed Methodology

CD-DNN is the subject of this section of the document, which focuses on it. By analyzing user reviews and item metadata from a wide range of sources spanning a wide range of industries, CD-DNN can gather data on both users and items. In this way, CD-DNN can collect data on both users and items. To learn about users and items at the same time, CD-DNN utilizes this technique. Because of the lack of data, it combines data from various fields. As a result of this transfer and utilization of knowledge, the target domain may benefit. Multi-domain Deep Neural Networks will be discussed in Section IV-B and Section IV-C, while Deep Neural Networks will be covered in Section IV-A. (SD-DNN). The preferences of users and the properties of database items are modeled independently in SD-DNN, a single domain model. There is a significant amount of discussion in this book's Section IV-B about the CD-DNN model, which is a step up from SD-DNN model. Other methods can be used to gather information about individuals or things using CD-DNN software and other techniques. Use a variety of training methods to train the CD DNN model in Section IV-C of this guide.

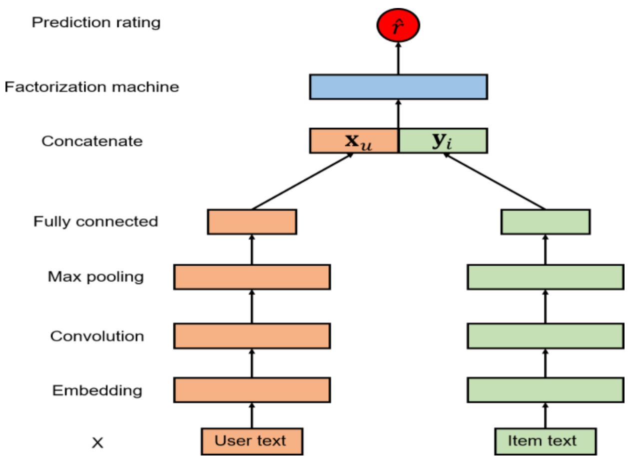

Fig 1. One parallel neural network can use the SD-DNN architecture.

Each neural network has a unique set of parameters for both users and objects. User and item feature vectors are combined using the concatenation technique, and text features are extracted and concatenated into a single vector by CNN layers. Factorization machines take into account interactions between users and items in order to make accurate predictions about what will happen next.

a. Single-Domain Deep Neural Network (SD-DNN)

Using a single domain as an example, Figure 1 shows SD-DNN architecture for rating estimation. It is composed of two neural networks that work in tandem. Networks for users (*Net$_u$*) and networks for items (*Net$_i$*) are two examples (*Net$_i$*). An input of reviews and item metadata is used to generate prediction ratings for this configuration of (*Net$_u$*)& *Net$_i$*. The user text and the item text can both be stored in the first layer of word embedding matrices. As a result, semantic information from both types of input texts can be extracted. Users' and items' features are typically discovered using the convolutional, maximum pooling and fully-connected layers in CNN-based models. CNN layers generate feature vectors that can be used to represent an individual dataset item. It is possible to create a single feature vector by combining the feature vectors of various users and items. As a final step, data about user-product interactions and latent variables is extracted using a factorization machine layer. *Net$_u$* and *Net$_i$* generated latent feature vectors in the previous layer, which are used to calculate the prediction rating in this layer. Considering that both networks are essentially the same, we will only cover the process for the *Net$_u$* network here. Input is the only difference between the two networks.

i. Embedding Layer

A layer known as the embedding layer processes the input data into low-dimensional embeddings. Using the formula f: *W*→ *R^c*, word embedding is accomplished by transforming each word into a dense, In this case, a convoluted projection function is used to generate a c-dimensional vector. Using word embedding techniques, SD-DNN models are able to extract the semantics from both user and item text. To represent the matrices in the embedding layer, word embedding matrices can be used. For each matrix A $d^u_{1:l}$, embedding layer is responsible for creating the matrices so that reviews and item metadata can be represented as word embeddings. document with *l* words is created before any embedded layers are added. This document includes all user u reviews. Following the embedding layer, a word embedding matrix is generated for user u and constructed as follows.

$V^u_{1:l}$

$$V^u_{1:l}=\theta(d^u_1) \oplus \theta(d^u_2) \oplus \theta(d^u_3) \oplus \ldots \oplus \theta(d^u_l) \qquad (2)$$

Where $d^u_k$is the kth word in user *u's* document $d^u_{1:l}$ and find function $\theta(d^u_k)$precedes the equivalent c-dimensional word vector for the word $d^u_k \oplus$ is a symbol for combination two words together.

Layers of CNN

Some of the techniques used in the subsequent layers include fully connected layers, convolutional layers, and maximum pooling layers [39]. Deep features are generated using the convolutional layer's input window, which is composed of the filter $w_c \in R^{hc}$. As an illustration, a feature g_i can be created by generating and storing a window of words $V^u_{k:k+h-1}$

$$g_i = f(w_c * V^u_{k:k+h-1} + b_c) \qquad (3)$$

$b_c \in R$ represents the bias, and f, which can be either a sigmoid or a rectified linear units (*ReLU*) activation function, represents the nonlinear function. The filter $\{V^u_{1:h}, V^u_{2:h+1}, \ldots, V^u_{n-h+1:n}\}$ is used to generate a feature map for each possible sliding window in the sentence.

$$g = [g_1, g_2, \ldots, g_{n-h+1}] \quad (4)$$

The feature map is then subjected to a maximum pooling operation in order to obtain the highest possible value when $g \in R^{n-h+1}$.

$$\hat{g} = max\,\{g\} \quad (5)$$

It's made up of N+ 1 parallel neural network that work together. A single network represents the preferences of the network's members, while the remaining N networks represent each item's unique characteristics.

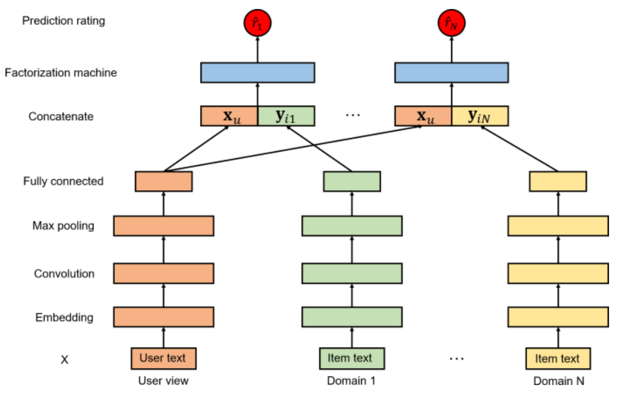

Fig 2. The architecture of CD-DNN for N+1 parallel neural network

Each feature map's most important feature must be found in the case where the current filter expects an attribute called $\hat{g}$ to be present. W_f is used to send the weight matrix to a layer that already has all of its connections established, as shown in the following illustration:

$$x_u = f(w_f * \hat{g} + b_f) \quad (6)$$

A user u would expect to see in a fully connected layer $x_u \in R^{n_2 \times 1}$ has thus far been defined by its output. User and item feature factors are calculated utilizing the methods described here. The x_u and y_i symbols represent these concepts, respectively.

iii. Factorization Machine Layer

However, the feature factors of users and items might be in special quality spaces, making it impossible to measure or compare their features using the outputs of the CNN layers previously discussed. As a result, we use the factorization machine technique to create a single feature space, which we call the factorization space. Net_u and Net_i networks are enhanced by a layer of factorization machines built on top of them. $z\hat{} = (x_u, y_i)$ must be combined into one feature vector in order to achieve this goal x_u and y_i. Accordingly, we use a factorization machine [40] in order to model all cross information in z^, and this is done as follows:

$$\hat{r} = \sum_{i=1}^{|\widehat{z}|} x \sum_{j=i+1}^{|\widehat{z}|} y < \widehat{v_i}\ , \widehat{v_j} > \widehat{z_i}\widehat{z_j} + \sum_{i=1}^{|\widehat{z}|} x\, \widehat{w_i}\widehat{z_i} + \widehat{w_0}$$
(7)

Where $\widehat{v_i}\ , \widehat{v_j}$ represents the annoyed information, i.e., second-order connections, and $\widehat{v_i}, \widehat{v_j} = \sum_{i=1}^{|\widehat{z}|} x \sum_{j=i+1}^{|\widehat{z}|} y\, vi$ is a unit of measurement variable representing the significance of the i^{th} variable in z^. $w\hat{}_0$ is a reasonable overall adjustment bias.

iv. Cross-Domain Deep Neural Network

As a result of our extensive data collection, we will now present CD-DNN, an extension of SD-DNN. Figure 2 depicts an architectural representation of the CD-DNN. N source domains and N + 1 parallel neural networks can be seen in the figure below. IV-A goes into great detail about the design of each network. The remaining N networks represent the properties of the network's items, while the first network represents the preferences of the network's users. The CD-user DNN inputs are mapped to latent factors in a feature space using a single mapping. As a result of this mapping, the user factors are optimized in conjunction with factors from other domains. A user's preferences in other domains can be matched to the user's preferences in a hidden space created by mapping this project. To learn about a proper mapping, you could try using another source domain with more historical records than the current target domain. The target domain can also benefit from the knowledge gained by users in other source domains, which suggests that users with interests similar to those of users in other source domains can be used to learn. As a result, the accuracy of recommendation generation can be significantly improved through the creation of a much better mapping of user and item profiles.

CD-DNN has a few unique features that are worth noting. To begin with, it's a simple game to pick up. As a second benefit, it allows the company to expand into new areas while still benefiting from its extensive experience in those areas. Auxiliary data from multiple sources and data from multiple source domains can be combined at the same time by this tool.

v. CD-DNN Training

Samples are used to create training data. There are two inputs for the *k-th* sample: a user network $X_{U,k}$ and an index of the used domain $X_{a,k}$. User network $X_{U,k}$ can be found on the second sample. There are no other domain inputs $X_{i:i6 \neq a}$ because they are all empty vectors of length 0. At a high level, we describe the CD-DNN training procedure on the first page of Algorithm 1.

Algorithm 1 *CD-DNN Training*

Input: N = Number of domains
M = Number of training cycles
K = Number of sample training pairs
A_U = Architecture of user network
$A_I = \{A_{I1}, \ldots A_{IN}\}$ = architecture of item network
W_U = weight matrix of user network
$W_I = \{W_{I1}, \ldots W_{IN}\}$ = weight matrices of item network
$X_U = \{X_{U1}, \ldots X_{UK}\}$ = inputs for user network
$X_I = \{X_{I1}, \ldots X_{IK}\}$ = inputs for item network
Output: W_U = user network of weight matrix
W_I = a group of item network weight matrices
1: ***Initialization***
Initialize A_U and A_I using W_U and W_I
2: ***for*** $m = 1$ to M ***do***
3: ***for*** $k = 1$ to K ***do***
4: train X_{Uk} and X_{Ik} using W_U and W_I
5: ***end for***
6: ***end for***

An example of regression can be found in this article's recommendations, which are based on the ranking of ratings. Problem The common squared loss will be used as an example in this section because of this. The training goal function L is denoted by the symbol

$$L = \sum_{u,i \in S}^{k} W\,(r_{u,i} - \hat{r}_{u,i})^2 \qquad (8)$$

Training set is S, When a user u has given a rating to an item i, a symbol appears to represent their rating by the letter $r_{u,i}$, and the prediction rating is also denoted by the letter $r_{u,i}$. L's objective function is optimized using the root mean square prop optimizer (RMSProp). Improved gradient descent (RMSProp) is used to further optimize problems where the loss function update has a large swing and speed up the convergence speed of problems with a large loss function update swing. A weighted average of the differential squared weights is used by the RMSProp algorithm to calculate weights and gradients. Over fitting can be avoided by employing a dropout strategy when training layers that are fully connected. When it comes to dropout strategies, they fall somewhere in the middle of the model's range of possibilities. In the algorithm, some hidden nodes are omitted at random. Each batch of training ignores a random number of nodes in the hidden layers. As a result, each training session has its own customized network. Since the hidden nodes' interactions are no longer relied upon, a random dynamic difference in the weight update can occur (but only on a subset of them). However, the weight update was able to maintain some degree of efficacy despite these difficulties.

By using a larger number of smaller neural networks, the model's performance can be improved after the test period [42]. If the model is averaged by using fewer neural networks, the results may be better; another side effect of dropout will result in better results than using the larger neural networks during the testing period.

5. Experimental Results

In this step, the accuracy of the proposed CD-DNN network will be assessed in order to show its utility. Our efforts will succeed or fail based on how well the datasets and evaluation metrics are evaluated. Materials were discussed in Section V-A of this chapter. According to Section V-B, there are four competing methods for baselines in this document. Conditions under which a study is conducted its being investigated by sections V-C. In this section, we look at how our CD-DNN stacks up against the competition. A word embedding method and a factorization machine are discussed in the final section of our model.

i. Evaluation Metric and Datasets

We used four datasets from the Amazon dataset that we selected for our experiments. Customer reviews and item metadata totaling 1428 million were collected by Amazon between May 1996 and July 2020. It was the most complete option because it contained both review text and item metadata, and it was the largest rating dataset that was publicly available. There are many datasets to choose from, as shown in Table 2. Customers of Digital Music leave at least ten reviews and ratings for each transaction, making it the most populous data set for researchers to work with. It is clear that the Toys and Games category is still in its infancy because the average number of reviews and ratings per item is less than twice that amount. A dearth of data in the recommender system's database severely reduces the accuracy of its recommendations. Ratings range from 1 to 5 and are represented numerically in the table below for each of the datasets in the collection. In order to predict how people will act, we can look at user reviews. The item's title and description can be used to model the item's properties in the same way. Reviews of user and metadata items are capped at p percent of the population and p percent of items to account for the wide distribution of reviews and metadata in the long tails of the population. The default value of p is 0.92. CD-DNN and its competitors are evaluated using the mean square error (MSE) metric, which measures the accuracy of CD-rating DNN's prediction accuracy (MSE).

When the MSE value is greater than 0, it is assumed that the quality of the recommendations is lower. Ground-truth rating $(r_{u,i})$ is multiplied by an item I to get the mean square error (MSE) and a prediction rating $\widehat{r_{u,\iota}}$.

$$MSE = \frac{1}{N} \sum_{u,i}^{p} W\,(r_{u,i} - \widehat{r_{u,\iota}})^2 \qquad (9)$$

The current dataset's number of instances is denoted by N.

Classification	Users	Items	Reviews	Reviews per User	Reviews per Item
Digital Music	6,652	4,679	75,817	22.8	29.2
Musical Instruments	2,530	1000	11,372	8.3	22.5
Video Games	17,127	9,439	45,864	3.3	5.3
Toys and Games	12,985	9,942	23,882	3.0	3.7

Table 2. Dataset statistics

ii. Benchmarks

Our CD-DNN is compared to three representative methods that use a rating prediction recommendation model to evaluate its effectiveness.

(1) The model is based on a matrix factorization. Our comparison tool for this section is SVD++ because it is an excellent place to start.

(2) There are also models with topic modeling capabilities. Topic modeling is a popular technique for extracting semantic information from review text. The Hidden Factor was the subject of our comparison between our CD-DNN and another model (HFT).

(3) A neural network is used to build the model. A neural attention regression model and a Deep Cooperative Neural Networks (DeepCoNN) model with Review-level Explanations were used to compare the two (NARRE).

- SVD++ [44] is a model extension that incorporates implicit feedback to improve accuracy. Based on the latent factor model and the SVD++ algorithm, it is built on the SVD model.
- A topic model that incorporates both ratings and customer reviews is used by HFT [9]. The goal of high-frequency trading (HFT) is to improve rating prediction accuracy by identifying previously unnoticed topics in customer reviews.
- DeepCoNN [28] uses a deep model to learn about an item's features and preferences from review text before applying that knowledge to the item itself. Neural networks that are linked to one another form the basis of this system the first method makes use of reviews written by current users to learn about user preferences, while the second method makes use of review text assigned to an item to learn about its features.
- In accordance with NARRE, [29] a new neural attention model has been developed that predicts appropriate ratings for items as well as selecting automatically explainable reviews that provide convincing recommendations in conjunction with these ratings.

iii. Settings for experiments

Table 2 contains the statistical data for the datasets in question. A 2.80% split is used for the validation and testing sets of each dataset, with 80% of each dataset being used for testing, and the remaining 20% being used for adjusting hyper-parameters in the system. The training sets are randomly selected from these remaining 20%. Using both the training set and the validation set, we can fine-tune our CD-hyper DNN's parameters and baseline values based on the MSE. All recommender algorithms are put to the test on this dataset.

So, we looked at what happened when we increased the number of latent factors from 10 to 50 in the HFT model by setting K to 10.

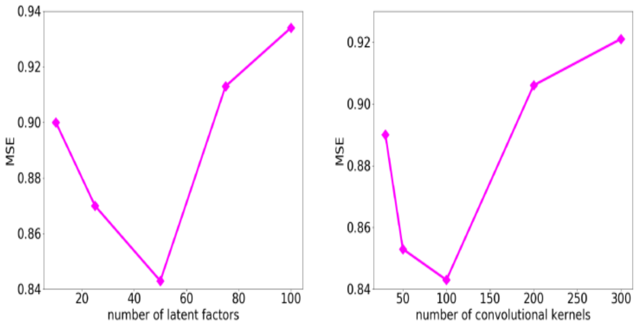

Fig 3. A direct correlation exists between the CD-performance DNN's and the number of latent factors and convolutional kernels. Toys and games are the subject of both the target and source domains.

For our CD-DNN, we conducted an experiment to evaluate the performance of the two critical parameters. ($|x_u|$ and $|y_i|$) are both latent factors, and the number of convolutional kernels is also important in this equation (n_c). Figure 3 depicts the MSE of CD-DNN while varying the former $|x_u|$ and $|y_i|$ from 10 to 100 and the latter (n_c) from 30 to 300 as a test set to investigate its performance using the Toys and Games validation set. The MSE decreases until it reaches its minimum when the number of latent factors and convolutional kernels are both increased to 50 and 100, respectively. To put it another way, (n_c) is 100 and $|x_u|$ equals $|y_i|$= 50. Other hyper parameters were also tested using a grid search method to see what impact they had. As a result, the learning rate α is set to 0.0102, and the batch size is set to 50. Setting the dropout ratio to 0.5, convolution kernel h is window size is set to 3, and the word embeddings dimension is set to 300. A word embedding that had been trained on more than 100 billion words [45] was also directly adopted by our CD-DNN to continue training the word vectors. In order to eliminate the possibility of error, each experiment was repeated three times.

iv. Evaluation of Performance

MSE quality is shown in Table 3 as a comparison between our proposed baseline and four publicly available baselines. In bold are the lines with the most promising results. The two rightmost columns in our CD-DNN vs. SVD ++ and NARRE comparison show the percentage gains our CD-DNN made. In the table, the following conclusions can be drawn. On all four datasets, SVD++ and DeepCoNN perform better than HFT. In terms of performance, DeepCoNN does not outperform NARRE. Furthermore, SVD++ and NARRE are both outperformed by it by an average of 10.4% and 18.0% in comparison to the best baseline when all datasets are considered together. For example, CD-DNN uses both the source and target domains, while SVD++ and HFT only use one of the source or target domains. There are many ways in which cross-domain recommendation can improve the accuracy of recommendations in the target domain if done correctly. In many ways, this information can be put to good use. During the experiments, the CD-DNN method proved successful.

CD-DNN outperforms SVD++ and NARRE on the most dense dataset available, Digital Music, by 3.8% and 2.2%, respectively (see Table 2). Furthermore, CD-DNN outperforms both SVD++ and NARRE on the datasets with the most sparse data: Toy- and Game-related dataset (by 16.6% and 32.2%, respectively). Consequently, this conclusion holds if the dataset is incomplete, the more accurate the recommendations made will be. Since the CD-DNN can significantly improve rating prediction in sparse datasets, it is possible to conclude that it is capable of relieving recommender systems of the issue of data scarcity.

Source Domain	**Target Domain**	i: HFT-10	ii: SVD++	iii: HFT-50	iv: NARRE	v: DeepCoNN	vi: CD-DNN	Performance vi vs i	Performance vi vs v
Musical Instruments	Digital Music	1.966	0.934	1.417	0.906	0.907	0.937	11.3%	8.5%
Digital Music	Musical Instruments	1.284	1.255	1.297	1.237	1.264	1.212	4.6%	3.4%
Toys and Games	Video Games	1.428	1.122	1.329	1.382	1.382	0.954	17.5%	34.4%
Video Games	Toys and Games	1.683	1.524	1.665	1.761	1.761	1.364	12.6%	25.9%
Average on all Datasets		1.590	1.208	1.427	1.321	1.328	1.116	12%	19%

TABLE 3. Analyzing the MSE in Light of Historical Data and data relevant to the target domain can be used in SVD++, hft, DeepCoNN, and NARRE.

v. Model Analysis

If our proposed CD-DNN uses word embedding to extract semantic meaning from text, as we have in the past, one might wonder if this would be beneficial. That's what you meant to say. To ensure that ratings are as accurate as possible, a factorization machine layer is used. Is our model used to make predictions? As a result, we're going to investigate these other options. The proposed CD-DNN is compared to four other variants in a study based on the Toys and Games dataset. CD-DNN-TFIDF is the abbreviation. Each of the three types of CD-DNN CDs has its own unique set of characteristics, which are summarized here. The following distinctions must be made in order to distinguish between these four options: An overview of the CD-DNN system is provided in the following:

• CD-DNN-TFIDF: CD-text DNN's input is modeled using the TF-IDF mechanism rather than word embedding, as previously described.

• CD-DNN-Trigram: CD-DNN models text instead of words using letter tri-grams [32] instead of word embedding. This makes it possible for us to more accurately model text.

• CD-DNN-Random: Input text is arbitrarily initialized into vectors of a specified length using a randomization algorithm.

• CD-DNN-DP: Instead of employing a factorization machine, a common dot product between xu and yiis used by the CD-DNN architecture to successfully resolve the problem.

In Figure 4, Playing with children's toys as a basis for comparing CD-DNN and its variants and other networks is most effective.

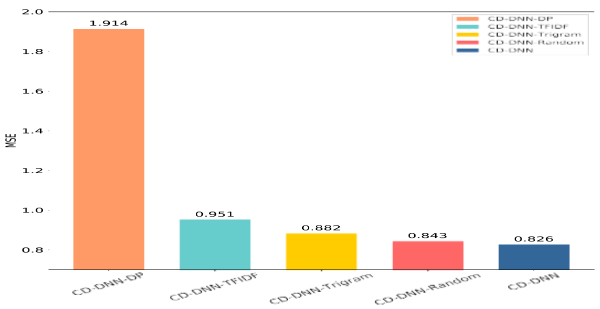

Fig 4. Variants of the proposed model are derived from video games, toys and games, as well as other sources of data.

There are three different word embedding techniques used in this demonstration: CD-DNN, TFIDF, and CD-DNN Trigram. Three word embedding models are being trained to see if and how they can mine the semantics of review text and item metadata. DNNs with CD superiority have been shown to represent latent representations of items and users using dense vector mapping to convert text into dense vectors. The MSE method is used to compare CD-DNN and CD-DNN-factorization classification accuracy. CD-DNN outperforms CD-DNN-DP when demonstrating the efficiency of the machine layer DP, as shown in Figure 4. Because the factorization machine can model xu and yi interactions in both first- and second-order interactions at the same time, this is an important consideration when using the machine.

6. Conclusion

Using all of the available user interests from multiple domains to create more complete user profiles and better recommendations has the potential to be extremely effective. The authors of this paper propose a new neural network, which they call CD-DNN, for making recommendations that span multiple domains. Users and products from a wide range of industries can be used as input when training the CD-DNN system. In order to optimise the user network in conjunction with product characteristics from various domains, it transforms user characteristics into latent spaces. Researchers found that CD-DNN outperformed all baseline recommender systems in terms of accuracy and that it alleviated the data sparsity problem by incorporating more data from different domains. They used four publicly accessible datasets to test their findings. By comparing our CD-DNN to others that only use TF-IDF, letter trigrams, and random text representation, we discovered the word embedding technique. Using this comparison, we were able to conclude that word embedding helps extract semantic information from text. The model's factorization machine layer was assessed to see if it was a success or not.

References

[1] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, ''A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks,'' *IEEE Access*, vol. 6, pp. 64301–64320, 2018.

[2] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, ''Location-aware deep collaborative filtering for service recommendation,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published.

[3] L. Wu, P. Sun, R. Hong, Y. Ge, and M. Wang, ''Collaborative neural social recommendation,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published.

[4] C.-Y. Lin, L.-C. Wang, and K.-H. Tsai, ''Hybrid real-time matrix factorization for implicit feedback recommendation systems,'' *IEEE Access*, vol. 6, pp. 21369–21380, 2018.

[5] X. Deng and F. Huangfu, ''Collaborative variational deep learning for healthcare recommendation,'' *IEEE Access*, vol. 7, pp. 55679–55688, 2019.

[6] R. J. Mooney and L. Roy, ''Content-based book recommending using learning for text categorization,'' in *Proc. 5th ACM Conf. Digit. Libraries (DL)*, 2000, pp. 195–204.

[7] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, ''A content-based recommendation algorithm for learning resources,'' *Multimedia Syst.*, vol. 24, no. 2, pp. 163–173, Mar. 2017.

[8] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, ''A content-based recommender system for computer science publications,'' *Knowl.-Based Syst.*, vol. 157, pp. 1–9, Oct. 2018.

[9] J. McAuley and J. Leskovec, ''Hidden factors and

hidden topics: Under- standing rating dimensions with review text,'' in *Proc. 7th ACM Conf. Recommender Syst. (RecSys)*, 2013, pp. 165–172.

[10] G.-N. Hu, X.-Y. Dai, Y. Song, S.-J. Huang, and J.-J. Chen, ''A synthetic approach for recommendation: Combining ratings, social relations, and reviews,'' in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1756–1762.

[11] J. Tang and K. Wang, ''Personalized Top-N sequential recommendation via convolutional sequence embedding,'' in *Proc. 11th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2018, pp. 565–573.

[12] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, ''Deep matrix factorization models for recommender systems,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3203–3209.

[13] I. Cantador, I. Fernández-Tobías, S. Berkovsky, and P. Cremonesi, ''Cross- domain recommender systems,'' in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 919–959.

[14] Y. Koren, R. Bell, and C. Volinsky, ''Matrix factorization techniques for recommender systems,'' *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[15] N. Jakob, S. H. Weber, M. C. Müller, and I. Gurevych, ''Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations,'' in *Proc. 1st Int. CIKM Workshop Topic-Sentiment Anal. Mass Opinion*, 2009, pp. 57–64.

[16] Y. Bao, H. Fang, and J. Zhang, ''TopicMF: Simultaneously exploiting ratings and reviews for recommendation,'' in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2–8.

[17] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, ''Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS),'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 193–202.

[18] G. Ling, M. R. Lyu, and I. King, ''Ratings meet reviews, a combined approach to recommend,'' in *Proc. 8th ACM Conf. Recommender Syst. (RecSys)*, 2014, pp. 105–112.

[19] S. Feng, J. Cao, J. Wang, and S. Qian, ''Recommendations based on comprehensively exploiting the latent factors hidden in items' ratings and content,'' *ACM Trans. Knowl. Discovery from Data*, vol. 11, no. 3, p. 35, 2017.

[20] R. Salakhutdinov, A. Mnih, and G. Hinton, ''Restricted Boltzmann machines for collaborative filtering,'' in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 791–798.

[21] S. Li, J. Kawale, and Y. Fu, ''Deep collaborative filtering via marginalized denoising auto-encoder,'' in *Proc. 24th ACM Int. Conf. Inf. Knowl. Man- age. (CIKM)*, 2015, pp. 811–820.

[22] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, ''Collaborative denoising auto-encoders for Top-N recommender systems,'' in *Proc. 9th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2016, pp. 153–162.

[23] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, and M. Ispir, ''Wide & deep learning for recommender systems,'' in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[24] R. Wang, B. Fu, G. Fu, and M. Wang, ''Deep & cross network for ad click predictions,'' in *Proc. ADKDD*, 2017, p. 12.

[25] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, ''Learning deep structured semantic models for Web search using clickthrough data,'' in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 2333–2338.

[26] Z. Xu, C. Chen, T. Lukasiewicz, Y. Miao, and X. Meng, ''Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling,'' in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2016, pp. 1921–1924.

[27] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, ''Hashtag recommendation for multimodal microblog using co-attention network,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3420–3426.

[28] L. Zheng, V. Noroozi, and P. S. Yu, ''Joint deep modeling of users and items using reviews for recommendation,'' in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017, pp. 425–434.

[29] C. Chen, M. Zhang, Y. Liu, and S. Ma, ''Neural attentional rating regression with review-level explanations,'' in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 1583–1592.

[30] J. Tang, S. Wu, J. Sun, and H. Su, ''Cross-domain collaboration recommendation,'' in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 1285–1293.

[31] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo, ‘‘Cross-domain recommendation via cluster-level latent factor model,’’ in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2013, pp. 161–176.

[32] A. M. Elkahky, Y. Song, and X. He, ‘‘A multi-view deep learning approach for cross domain user modeling in recommendation systems,’’ in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 278–288.

[33] G. Pandey, D. Kotkov, and A. Semenov, ‘‘Recommending serendipitous items using transfer learning,’’ in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2018, pp. 1771–1774.

[34] G. Hu, Y. Zhang, and Q. Yang, ‘‘Conet: Collaborative cross networks for cross-domain recommendation,’’ in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 667–676.

[35] B. Liu, Y. Wei, Y. Zhang, Z. Yan, and Q. Yang, ‘‘Transferable contextual bandit for cross-domain recommendation,’’ in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3619–3626.

[36] A. Taneja and A. Arora, ‘‘Cross domain recommendation using multidimensional tensor factorization,’’ *Expert Syst. Appl.*, vol. 92, pp. 304–316, Feb. 2018.

[37] C. Gao, X. Chen, F. Feng, K. Zhao, X. He, Y. Li, and D. Jin, ‘‘Cross- domain recommendation without sharing user-relevant data,’’ in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 491–502.

[38] G. Hu, Y. Zhang, and Q. Yang, ‘‘Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text,’’ in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2822–2829.

[39] Y. Kim, ‘‘Convolutional neural networks for sentence classification,’’ 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882

[40] S. Rendle, ‘‘Factorization machines with libFM,’’ *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–22, May 2012.

[41] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and

R. Salakhutdinov, ‘‘Dropout: A simple way to prevent neural networks from overfitting,’’ *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[42] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, ‘‘Attentional factorization machines: Learning the weight of feature interactions via attention networks,’’ in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3119–3125.

[43] R. He and J. McAuley, ‘‘Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering,’’ in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 507–517.

[44] Y. Koren, ‘‘Factorization meets the neighborhood: A multifaceted collaborative filtering model,’’ in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ‘‘Distributed representations of words and phrases and their compositionality,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[46] Dattatraya, K.N., Rao, K.R. “Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN”, Journal of King Saud University - Computer and Information Sciences, 2022, 34(3), pp. 716–726

[47] Dattatraya, K.N., Raghava Rao, K, “Maximising network lifetime and energy efficiency of wireless sensor network using group search Ant lion with Levy flight”, IET Communications, 2020, 14(6), pp. 914–922.

[48] Ramkumar, J., Karthikeyan, C., Vamsidhar, E., Dattatraya, K.N.,” Automated pill dispenser application based on IoT for patient medication”, EAI/Springer Innovations in Communication and Computing, 2020, pp. 231–253.

[49] Dattatraya, K.N., Raghava Rao, K., Satish Kumar, D., “Architectural analysis for lifetime maximization and energy efficiency in hybridized WSN model”, International Journal of Engineering and Technology(UAE), 2018, 7, pp. 494–501.

[50] Dattatraya, K.N., Ananthakumaran, S., “Energy and Trust Efficient Cluster Head Selection in Wireless Sensor Networks Under Meta-Heuristic Model”, Lecture Notes in Networks and Systems, 2022, 444, pp. 715–735.

[51] Dattatraya, K.N., Ananthakumaran, S., Kiran, K.V.D., “Optimal cluster head selection in wireless sensor network via improved moth search algorithm”, Artificial Intelligence in Information and Communication Technologies, Healthcare and Education: A Roadmap Ahead, 2022, pp. 95–108.

[52] Manikandan, G. ., Hung, B. T. ., S, S. S. ., & Chakrabarti, P. . (2023). Enhanced Ai-Based Machine Learning Model for an Accurate Segmentation and Classification Methods.

International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 11–18. https://doi.org/10.17762/ijritcc.v11i3s.6150

[53] Christopher Davies, Matthew Martinez, Catalina Fernández, Ana Flores, Anders Pedersen. Using Machine Learning for Early Detection of Learning Disabilities. Kuwait Journal of Machine Learning, 2(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/172

[54] Sharma, R., Dhabliya, D. A review of automatic irrigation system through IoT (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 24-29.

[55] Motghare, S. M. ., & Nair, P. S. . (2023). Empirical Analysis of Privacy Preservation Models for Cyber Physical Deployments from a Pragmatic Perspective. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 19–29. https://doi.org/10.17762/ijritcc.v11i3s.6151

[56] Christopher Davies, Matthew Martine, Catalina Fernández, Ana Flores, Anders Pedersen. Improving Automated Essay Scoring with Machine Learning Techniques. Kuwait Journal of Machine Learning, 2(1). Retrieved from http://kuwaitjournalss.com/index.php/kjml/article/view/173

[57] Timande, S., Dhabliya, D. Designing multi-cloud server for scalable and secure sharing over web (2019) International Journal of Psychosocial Rehabilitation, 23 (5), pp. 835-841.