Research Article

Akanksha Bansal Chopra* and Veer Sain Dixit*

An adaptive RNN algorithm to detect shilling attacks for online products in hybrid recommender system

https://doi.org/10.1515/jisys-2022-1023
received January 10, 2022; accepted August 17, 2022

Abstract: Recommender system (RS) depends on the thoughts of numerous users to predict the favourites of potential consumers. RS is vulnerable to malicious information. Unsuitable products can be offered to the user by injecting a few unscrupulous "shilling" profiles like push and nuke attacks into the RS. Injection of these attacks results in the wrong recommendation for a product. The aim of this research is to develop a framework that can be widely utilized to make excellent recommendations for sales growth. This study uses the methodology that presents an enhanced clustering algorithm named as modified density peak clustering algorithm on the consumer review dataset to ensure a well-formed cluster. An improved recurrent neural network algorithm is proposed to detect these attacks in hybrid RS, which uses the content-based RS and collaborative filtering RS. The results are compared with other state of the art algorithms. The proposed method is more suitable for E-commerce applications where the number of customers and products grows rapidly.

Keywords: recommender system, shilling profile, modified density peak clustering, adaptive recurrent neural network

1 Introduction

1.1 Background of study

The recommender system (RS) gives recommendations to help potential buyers for choosing complex information and different products. RSs are becoming more and more indispensable with increasing number of choices available online [1]. Many e-commerce platforms such as JD.com and Amazon have launched many online services in recent years. These platforms are mostly used by consumers to obtain and share different products through online reviews [2]. To purchase products or services, the online reviews have become a significant data source for consumer. For example, among 2010–2016 the amount of customers reading online reviews enhanced to 91 from 71% according to the consumer market report [3]. The consumer read the online reviews to create the top purchase decision, when consumers decide to buy the products or services. However, it is very tough for buyers to read the entire online reviews within a limited time. From the online reviews, useful information are obtained which help to select the desirable products

* **Corresponding author: Akanksha Bansal Chopra,** Department of Computer Science, SPM College, University of Delhi, New Delhi, India, e-mail: akankshabansal.asm@gmail.com
* **Corresponding author: Veer Sain Dixit,** Department of Computer Science, ARSD College, University of Delhi, New Delhi, India, e-mail: veersaindixit@rediffmail.com

Open Access. © 2022 Akanksha Bansal Chopra and Veer Sain Dixit, published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

and rate these products [4]. Therefore, it is important for researchers to focus on techniques and methods for rating products depending upon online reviews to support purchase decisions of consumers [5].

Nowadays, depending upon the online reviews many schemes for rating products include the data fusion procedure from the existing research results. The sentiment orientation of each online review can be automatically identified and other products ranking can be explained by using this method [6–8]. Collaborative filtering (CF) and content based (CB) are the two algorithms mainly used in the RS. Nowadays, in the recommendation field the CF is very popular and it plays a vital role. In RSs there are number of hazard attacks. Depending on the purpose of attackers, the attacks can cause various damages to vulnerable systems. The attacks directly affect the performance of RSs [9]. In RSs, the biased user's representation not only makes the system vulnerable to attacks but also leads to improper recommendations to its users [10]. The large number of attacks results in system collapse under such conditions. It is very tough to stop unscrupulous users from injecting fake profile into a system [11]. The attack profiles must be detected and removed to ensure the trustworthiness of RS among its users. Two attacks such as push and nuke are presented in this article. The shilling attacks can be categorized as push and nuke attacks. To promote or demote the predictions is the main objective of these attacks which are made for targeted items. The rating of m-dimensional vector is considered by attack profile, where the total amount of products in the system is represented as "m." Figure 1 depicts the four parts of profile. The null partition "I_ϕ" are those items in the profile with no ratings. Depending on the attack type a rating designed will be given by single target item "I_t" to convert its recommendation; usually this will be both minimum (r_{min}) and maximum (r_{max}) possible rating. During the attack for special treatment a few attacks are needed to classify a group of products. This superior set "I_s" typically gets high ratings to make the profiles like user who desire these items. At last, to complete the profile, set of filler items "I_F" is included. This method is used for choosing the products in "I_F" and "I_S" and the ratings given to these things characterize an attack model.

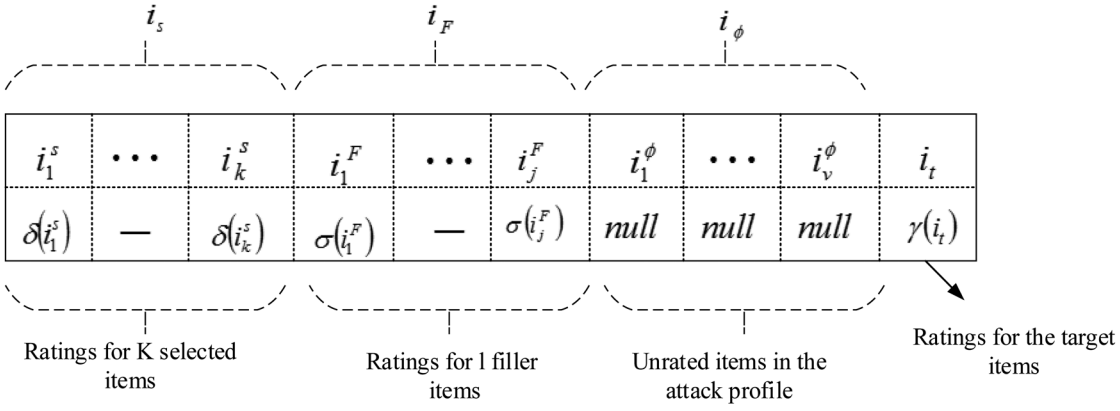

Figure 1: Shilling attack model.

In existing work, the researchers has studied the implementation of algorithm from an area of artificial intelligence (AI) and machine learning (ML) algorithm with RSs [12,13]. The ML algorithms such as Bayes network, clustering, ordinary least square regression, logistic regression, random forest, support vector regression, and k-nearest neighbour are the oldest ML techniques. These algorithms are used to give consumers good recommendations in RSs [14–16]. However, due to the number of methods and variations presented in the literature, the ML field does not have a clear classification method and also there is demand of strong ML algorithms and more robust to outperform. As a consequence, when developing an RS, it becomes difficult and confusing to choose an ML algorithm. One of the most robust ML algorithms is Ensemble model. It produces highly accurate results and improves the accuracy and performance [17].

Ensemble can give high accuracy with strong predictive power which may help to select some best products, however, selection of model is really hard and time consuming.

The proposed algorithm uses a hybrid approach in which an ML algorithm is integrated with an optimization algorithm to construct a more powerful and effective prediction model for RS. In this work, adaptive recurrent neural network model is developed by the integration of opinion mining approach with recurrent neural network (RNN) [18] and Bald Eagle Search (BES) optimization algorithm [19]. The two key approaches used – RNN and BES – are discussed in the subsequent subsections.

1.2 Problem statement

In the previous research, some researchers have focussed on sentiment analysis to enhance the accuracy of collaborative recommender system (CFRS). On the other hand, other researchers have used clustering algorithms and used classification approach to detect the shilling profiles (or PN-Attacks) in CFRS. Also, some have used neural network approach to enhance the accuracy of CFRS. The existing research works either focus on enhancing accuracy or to detect the shilling attacks in CFRSs. There is no single algorithm that can do both, that is enhance and detect the attacks. Furthermore, the existing algorithms consider only product ratings to identify attacks. To bridge this gap, in this work, a hybrid approach is used in which RNN is applied on user reviews to detect the PN-Attacks. Also, modified density peak clustering algorithm (MDPCA) is used for clustering to find the most preferred user from each cluster and Gaussian kernel is used to enhance the performance of the proposed model, named as adaptive recurrent neural network (ARNN) model. The BES Optimization Algorithm is used to initialize and update the weights of RNN. The proposed ARNN model is developed for hybrid recommender system (HRS).

1.3 Contributions

The major contributions/highlights of this research are as follows:

(i) The pre-processing is used to find and eliminate the unwanted attacks in consumer reviews of Amazon product datasets.
(ii) The output of the pre-processing is given to the feature extraction by word embedding technique.
(iii) Developing an MDPCA on consumer reviews of Amazon product datasets to ensure a well-formed cluster.
(iv) The Gaussian kernel is used to enhance the performance of the density peak clustering algorithm (DPCA) to evaluate the local concentration.
(v) An efficient ARNN is used to obtain the maximum preferable user from each cluster. The RNN model parameter values (weights) are initialized through the fitness computation of BES “search agents.”

1.4 Organization of the article

The article is organized as follows. Section 2 discusses the related work. Section 3 discusses the methodology used. Section 4 gives the implementation of proposed model with discussion about the attacks, preprocessing stage, feature extraction by word2vect method, MDPCA technique for clustering, and ARNN method. The performance analysis depicting the results is presented in Section 4. Finally, Section 5 gives the conclusion and future direction.

2 Related work

For future prediction of online products, a method of Improved Adaptive Neuro-Fuzzy Inference System [20] performs a weighting factor for sentiment analysis of an online product review. At first, from the dataset the data values are divided into collaboration based (CLB), CB, and grade-based (GB). Next, by utilizing DLMNN every consequence goes through review analysis (RA) as positive, negative, and neutral. Supermarkets can help to obtain new customers, easy transactions, and easy buying. There are some issues related to user profiling and cold start problems. Textual reviews of users are used to design a cold start hotel recommendation system [21]. HRS is one of the main system modules, which help to overcome the issues of CB RS and traditional CFRS. Based on hybrid recommendation algorithm [22], a novel implementation of a product RS is used for product reviews using sentiment analysis [23]. Based on the original structure to deliver a visual data organization is the main advantage of this method. Sentiments, reviews, and ratings are evaluated and characterized as negative and positive sentiments. To avoid fake reviews, the media access control-based filtering method can be used.

To make the product's prediction, the recommendation algorithms mostly depend upon the user's rating. For RSs, they present a contextual information sentiment-based model. Based on contextual information of a product, sentiment analysis [24] is integrated with prospect theory-based method [25] to rank the products. In electronic product recommendation by using results of RMSE and MAE measurements their contextual information sentiment-based model illustrates better performance. Based on the automatic features, an ensemble detection method [26] is used for shilling profile detection. At first, the users' behaviour is analysed to collaboratively discover the shilling profiles from multiple views such as ratings, user graph, and product popularity. Second, the stacked denoising auto encoders are used based on the data pre-processed from multiple views to automatically extract the consumer features with various corruption rates. Moreover, based on principal component analysis (PCA) the features extracted from multiple views are effectively combined. Finally, the weak classifiers are generated according to the features extracted with various corruption rates and then combined to detect attacks.

A multi-criteria RS for CF techniques exploits the user given information in regard to their reviews and provides multi-faceted representation of user's interest [27]. Similarity with opinion of users [28] is combined to provide a better quality RS. Sentiment analysis [29] is a tool to rate an product in the RS. CB RS is combined with sentiment analysis [30] to improve prediction results. Textual features are extracted to learn about users and a CB recommendation system [31] is developed. Influence analysis [32] is another way to improve attack detection performance in CFRS. The method is applied only on influential users instead of the whole user set. A CFRS may be secured by integration of AI and block chain [12]. Use of intense sentiments, micro reviews, and trust relationships [33–35] boosts user preferences for a product in a social RS to enhance the accuracy. A user-product-based model using textual facts [36] and user-product latent factors can be used for sentiment analysis. Sentiment analysis approach is integrated with CF algorithms [37] and used for rating products. The polarity metric [38] of sentiment analysis gives an improved CFRS. The researchers [39] discussed a classification approach for detection of ratings and presented various features to detect ratings based on attack models. Unsupervised clustering [40] is used for attack detection using basic descriptive statistics in CFRS.

In the discussed related works, it may be noticed that some researchers have focussed on sentiment analysis to enhance the accuracy of CFRS. On the other hand, other researchers have used clustering algorithms and used classification approach to detect the shilling profiles (or PN-Attacks) in CFRS. Also, some have used neural network approach to enhance the accuracy of CFRS. The existing research works either focus on enhancing accuracy or to detect the shilling attacks in CFRSs. There is no single algorithm that can do both, that is enhance and detect the attacks. Furthermore, the existing algorithms consider only product ratings to identify attacks. To bridge this gap, in this work, a hybrid approach is used in which RNN is applied on user reviews to detect the PN-Attacks. Also, MDPCA is used for clustering to find the most preferred user from each cluster and Gaussian kernel is used to enhance the performance of the proposed model, known as ARNN model. The BES Optimization Algorithm is used to initialize and update the weights of RNN. The proposed ARNN model is developed for HRS.

3 Methodology

3.1 RNN

RNN is a type of neural network with feedback connections among nodes. It has an ability to model a dynamic system. RNN has three different layers, i.e. input, output, and hidden layers, as shown in Figure 2. RNN uses the following recursive formula:

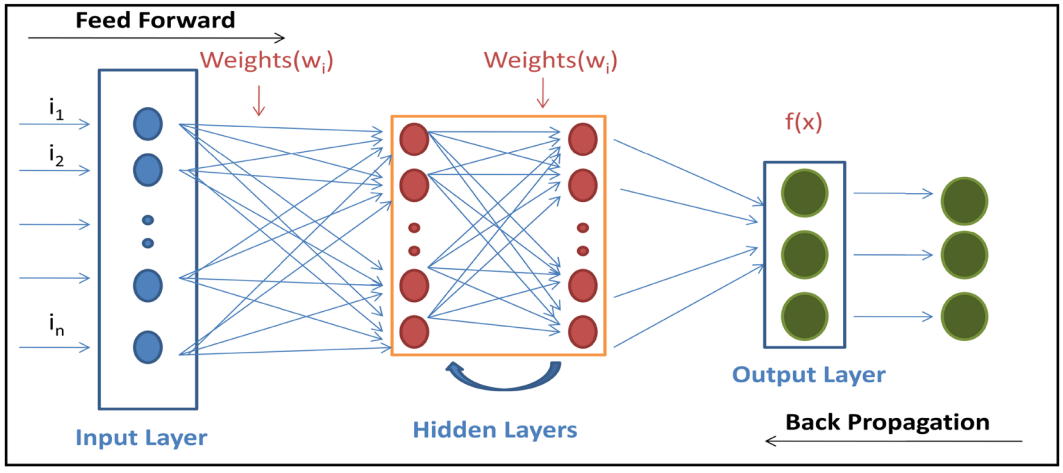

Figure 2: Structure of RNN.

$$h_t = \tanh(W_h h_{t-1} + W_x x_t), \tag{1}$$

$$y_t = W_y h_t, \tag{2}$$

where "h_t" is the hidden layer vector, "W_h" is the weighting matrix, "x_t" is the input vector, and "y_t" is the output vector.

Long-term dependency issue is presented in RNN, and the issues of weight matrix and long interval time keep multiplying recurrently with earlier outcomes. This may cause exploding gradient and vanishing gradient issues. To avoid this issue, long short-term memory (LSTM) is used, which will enhance the performance.

In the LSTM, every neuron is a memory cell. Every neuron includes three gates, namely, input gate, forget gate, and output gate. The three gates are discussed as follows:

1. The forget gate $F(t)$ defines unwanted data. At the previous unit "$t-1$," by entering the output "h_{t-1}" and adding the input "x_t" with current time "t," softmax function "$s(t)$" is given as follows:

$$F(t) = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3}$$

$$s(t) = \sum_j \frac{\exp(z_i)}{\exp(z_j)}. \tag{4}$$

2. The input gate explains which new data to be recollected in cell state.

$$i(t) = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{5}$$

$$\tilde{C}(t) = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{6}$$

To obtain the updated information the values of "$i(t)$" and $\tilde{C}(t)$ are multiplied by the sigmoid function that we want to add to the cell state.

$$C(t) = F(t) \times C(t-1) + i(t) \times \tilde{C}(t). \tag{7}$$

3. The output gate explains which data will be output in the cell state. The cell state is first triggered in the "tanh" layer before being multiplied by "$o(t)$." At time "t" the multiplication result is the output data "$h(t)$" in the block of LSTM.

$$o(t) = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{8}$$

$$h(t) = o(t) \times \tanh(C_t). \tag{9}$$

The available data are categorized into three non-overlapping sets for the purpose of training, testing, and validation. At first, the LSTM model in consumer reviews of Amazon products dataset is trained. To choose the best parameters as well as the performance in the proposed model the validation is employed. Finally, the same dataset is utilized for testing purpose and the performance and accuracy are verified. The weight and bias value of all the three gates is updated by BES Optimization Algorithm [19]. To validate the co-sequences of every phase of hunting is the main behaviour of bald eagle.

3.2 BES optimization

The main behaviour of the bald eagle is to validate the consequences of every phase of hunting. The fitness function of BES is calculated by:

$$\text{Fitness } f(t) = \max \sum \frac{W_i(t)}{W_{\text{best}}}. \tag{10}$$

The hunting behaviour of BES can be classified into three stages, i.e. select, search, and swooping.

1. In the select stage, the BES finds and picks the best area as best bias within the chosen search space.

$$b_{\text{new},i} = b_{\text{best}} + \alpha \times r(b_{\text{mean}} - b_i), \tag{11}$$

where random number is denoted by "r."

2. In the search stage, the best position as best weight value for the swoop is mathematically calculated by

$$W_{i,\text{ new}} = W_i + y(i) \times (W_i - W_{i+1}) + x(i) \times (W_i - W_{\text{mean}}). \tag{12}$$

3. In the swooping stage, the bald eagle blows from best weight in the search space and best bias in the best area. Both these are calculated and mathematically illustrated as follows:

$$\begin{aligned} W, b_{i,\text{ new}} = \text{rand} \times W, b_{\text{best}} + x_1(i) \times (b_i - c_1 \times P_{\text{mean}}) \\ + y_1(i) \times (W_i - c_2 \times W_{\text{best}}). \end{aligned} \tag{13}$$

3.3 Proposed framework

The proposed framework for HRS is implemented in four phases. First of all, the user data are collected which contain ratings and reviews of various products from dissimilar users. The user data include both genuine and attack profiles. The collected data then enter into Phase 1 and are pre-processed. Dissimilar products are eliminated and feature extraction is applied using Word2Vec technique. In Phase 2, MDPCA is applied to form clusters and the most preferable user from every cluster is obtained. The proposed ARNN model is applied on the reviews and ratings of this most preferable user, in phase 3. Finally, the classification is done in Phase 4. If the rating is found genuine, then it is stored in the database to generate the recommendation otherwise it is discarded considering it to be an attack. The higher rating "5" is labelled as "PUSH ATTACK" (P-Attack) and the lower rating "1" is labelled as "NUKE ATTACK" (N-Attack). The rating is labelled "NORMAL" otherwise (i.e. for ratings "2," "3," and "4"). Figure 3 presents the proposed framework for the HRS.

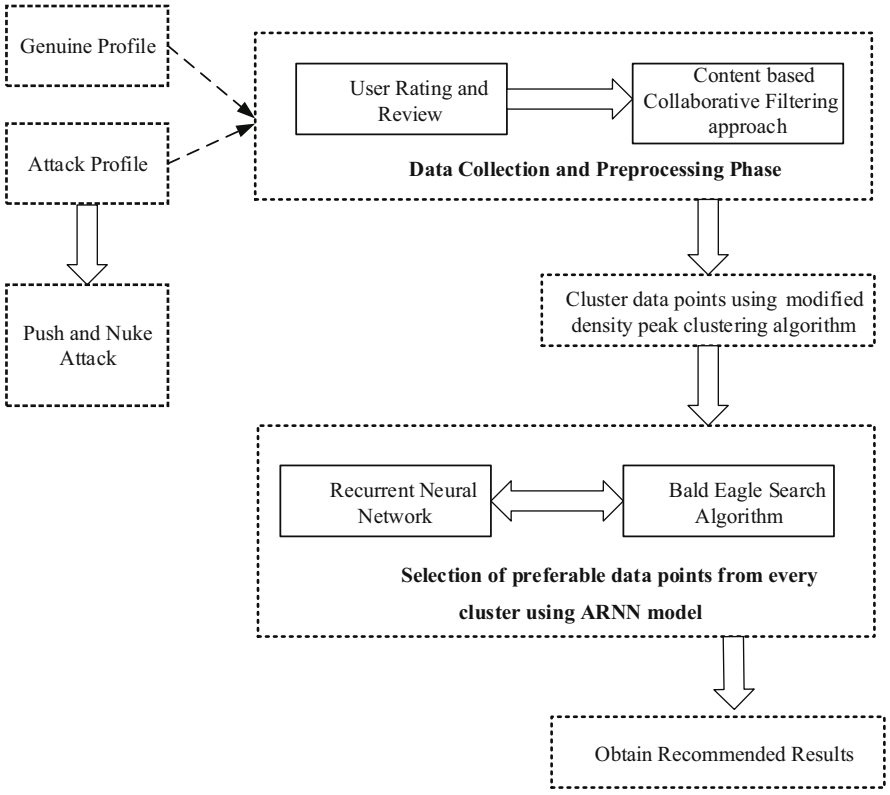

Figure 3: Proposed framework for HRS using the ARNN model.

4 Implementation

The implementation starts with the collection of user ratings and reviews of dissimilar users on numerous products from a given dataset. The collected user data contain both genuine and attack profiles. The dataset is divided into training set and test set. The training set is used to label the data, and the test set is used to make prediction. The user rating in the training set is labelled as "NORMAL" for ratings "2," "3," and "4;" "PUSH ATTACK" for rating "5" and "NUKE ATTACK" for rating "1." The ARNN model is applied to both training set and test set. The PN-Attacks are predicted by comparing the final values of test data (obtained at the Output Layer) with the training set (which has labelled rating data). The proposed framework is implemented on the data in four phases as follows.

4.1 Phase 1A: Pre-processing

In the pre-processing [41] stage, products with dissimilar features are eliminated. These dissimilar products are measured as outliers and are eliminated because if the products are rated through significantly smaller number of users, then considering such products into subsequent phases of RS will affect its whole performance. The user reviews of filtered products are then given to the pre-processing unit. The steps involved in pre-processing are removal of stop words, lower case, punctuation, and lemmatization.

(1) *Stop word removal*: Stop words such as articles (a, an, the), prepositions (on, at, in, there, this etc.), conjunctions (for, but, and, or, etc.) are removed from the given reviews. These words are removed because, generally, they do not contribute any significant value to the review.

(2) *Lower case*: All words of the review are converted into lower case. Similar words are then combined and the dimensionality is reduced. For example, word Star will be converted as star and if it appears twice in the sentence, it will be combined as one.
(3) *Punctuation removal*: This is a classic method in data mining and information retrieval. The punctuations are removed from the text. However, the presence of punctuation marks often denotes the existence of some sentiment but it does not contribute mathematical value to the text.
(4) *Lemmatization*: The process of merging many words to one is known as lemmatization. This lemmatization analyses a word morphologically and removes its inflectional ending.

Table 1 shows an example of pre-processing a review from the given dataset.

Table 1: Pre-processing example

Pre-processing	Review
Original	Coffee at Starbucks! Awesome coffee. I had never drank such a coffee.
Stop word removal	Coffee Starbucks! Awesome coffee never drank coffee.
Lower case	Coffee starbucks! awesome coffee never drank.
Punctuation removal	Coffee starbucks awesome coffee never drank
Lemmatization	Coffee starbucks awesome coffee never drank

4.2 Phase 1B: Feature extraction

A machine cannot understand the text directly. To make the text understandable, each word of text is represented in a real-valued vector. This procedure of representation of a word into real-valued vector is called word-embedding. In word-embedding technique, the words with similar meanings have similar representation. TF-IDF is one such technique of word-embedding. This technique is discussed and used in the proposed hybrid deep neural network model for CFRS in the previous chapter. The main drawback of this technique is that it cannot maintain the sequence of words. This problem is handled by another word-embedding technique known as Word2Vec [42]. This technique enables similar words to have similar dimensions and, consequently, helps to bring context. Each sentence in this technique is known as a "Vector." This means that all individual words of a sentence are treated as one – one feature of a vector. For the given vectors, a vocabulary is created by taking only non-repeated vectors followed by creating the one-hot encoding of the vectors. Consider reviews as follows:

R1: This dish is very spicy and hot
R2: This dish is not spicy and not hot
R3: This dish is yummy and good

Then, vocabulary is created as: **Vocab:** {**This, dish, is, very, spicy, and, hot, not, yummy, good**}. One-hot coding of the Vocab is created as shown in Table 2.

Table 2: Word2Vec one-hot encoding

One-hot encoding with vector length 10	
This	[1 0 0 0 0 0 0 0 0 0]
Dish	[0 1 0 0 0 0 0 0 0 0]
Is	[0 0 1 0 0 0 0 0 0 0]
Very	[0 0 0 1 0 0 0 0 0 0]
Spicy	[0 0 0 0 1 0 0 0 0 0]
And	[0 0 0 0 0 1 0 0 0 0]
Hot	[0 0 0 0 0 0 1 0 0 0]
Not	[0 0 0 0 0 0 0 1 0 0]
Yummy	[0 0 0 0 0 0 0 0 1 0]
Good	[0 0 0 0 0 0 0 0 0 1]

In the above example, each word occupies one of the ten dimensions. This means that there is no similarity among the words irrespective of their literal meanings. Word2Vec technique establishes the association between the similar meaning words. Word2Vec is an unsupervised model where the corpus can be passed without giving any label. But, since it is basically an internal neural network, it leverages a supervised classification model to get the embeddings from the corpus. There are two methods in word2vec for word embedding, known ashpen Continuous Bag of Word (cBOW) and Skip-Gram (SGram).

cBOW technique tries to find the target word using context word. The hidden layer is eliminated and the words with the same position are analysed. This way of procedure is known as bag of word. The continuously distributed words are thus named as cBOW. In this method, the context word is given as input "X" and the target words are predicted. Figure 4 represents the cBOW structure.

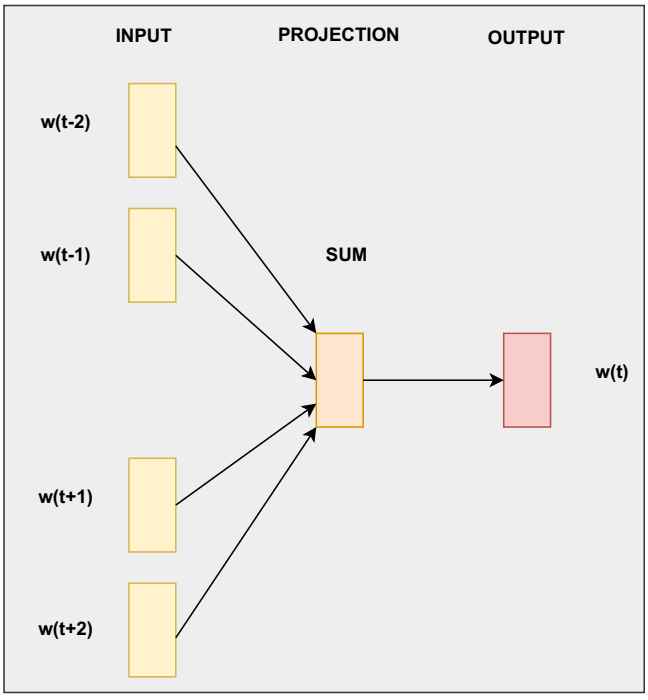

Figure 4: cBOW structure (Source: https://arxiv.org/pdf/1301.3781.pdf Mikolov et al., 2013).

Another technique is known as SGram. In this technique, context word is searched using the target word. This method tries to find the words in the same sentence. We utilize the present word with hidden layer projection as an input that may calculate the words within the range. This method predicts multiple words from a single given word. In the SGram model, target word is given and the context words are predicted. Consider a review: "The dish is very spicy and hot;" and a context window size of 2, the target word is "spicy," then the model tries to predict ["very," "and"], and so on. The words are fed in pairs of (X,Y) where "X" is the input and "Y" is the label. This is done by creating "Positive Input Samples" (PIS) and "Negative Input Samples" (NIS). PIS have training data in the form [(TARGET, CONTEXT), 1], where "TARGET" is the centre word, "CONTEXT" is the surrounding word, and "1" is the label which indicates the relevant pair. NIS has training data in the form [(TARGET, RANDWORD), 0], where "TARGET" is the centre word, "RANDWORD" is the randomly selected word, and "0" is the label which indicates the irrelevant pair. Figure 5 represents the structure of SGram method.

In the proposed framework, SGram model of Word2Vec technique is used. The model is implemented in the python platform using "genism" and "pickle" modules. For the given dataset, the total number of epochs is 20. The total number of embeddings is 5,000 with a vocabulary size of 6,408 and a window size of 4.

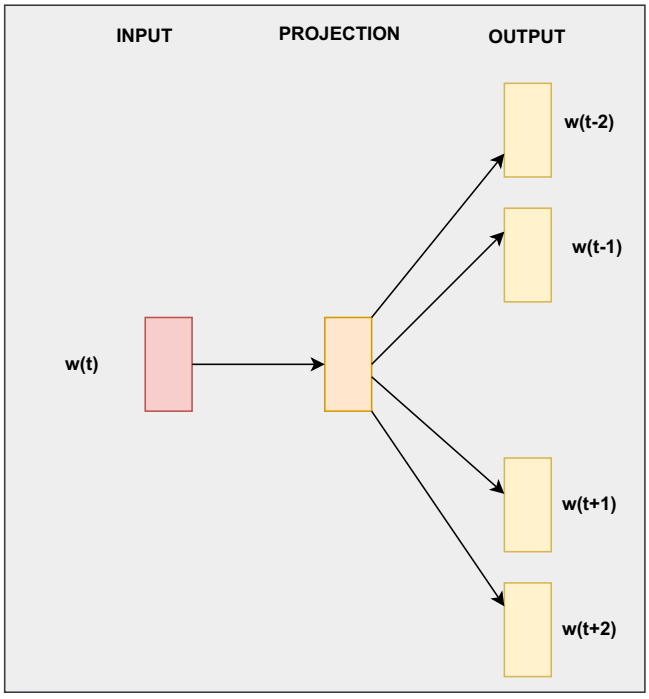

Figure 5: SGram structure (Source: https://arxiv.org/pdf/1301.3781.pdf Mikolov et al., 2013).

4.3 Phase 2: MDPCA for clustering

The objective of clustering is to sort the data into groups depending on characteristics similarity. Thus, information from dissimilar clusters is different from each other and information from similar clusters have the same properties. To find the cluster centres is the main idea of DPCA that fulfil two significant assumptions which are given as: (i) by lower density neighbour the cluster centre is bounded, (ii) the centre of cluster is away with a higher local concentration from any other cluster centre point.

To calculate the cluster centres the DPCA uses two measures such as local concentration and distance. Distance among two data points should have been determined by the two measures. For computing the distance between two data points, first DPCA accepts the Euclidean distance. However, the distance can cause several misclassifications if the dataset is inseparable and complex. Thus, we enhance the DPCA which is known as MDPCA. The Gaussian kernel is one of the modified algorithms which is developed to calculate the distance. Moreover, for calculating the local concentration the Gaussian kernel is used in place of cut-off kernel to improve the performance of the proposed method. Thus, the MDPCA allocates each and every residual point to the similar cluster as its nearby neighbour with high density after that cluster centre has been found (Figure 6).

4.4 Phase 3: ARNN model (using experimental modelling)

The most preferred user from every cluster is passed on to the ARNN model. The dataset is split into training dataset and test dataset. In the training dataset, the user ratings “2,” “3,” and “4” are labelled as “NORMAL”; higher ratings “5” are labelled as “PUSH ATTACK,” and lower ratings “1” are labelled as “NUKE ATTACK.” The ARNN model is applied to both training dataset and test dataset. The results obtained for the test dataset are then compared with the training dataset and attacks are predicted. The implementation of ARNN model uses a hybrid approach by integrating RNN with BES Optimization and works with two procedures. One is Forward Feed and the other is Backward Feed. The implementation is done in the Python

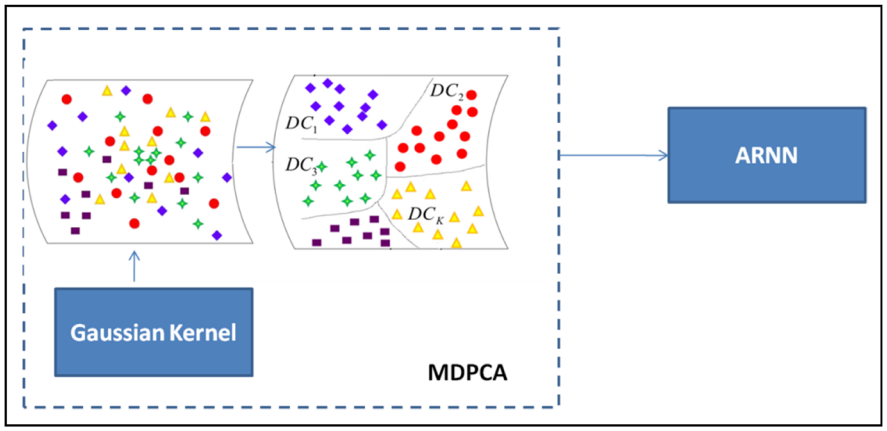

Figure 6: MDPCA.

platform using the "keras" module. Because of the gradient issue, LSTM RNN is used. During the Forward Feed, using the BES algorithm, the fitness value is computed.

The algorithm enters into the select stage and the best bias is found to be 25–75%. The value of alpha (α) is taken as 0.25. Now, the algorithm enters into the search stage and the best weight is picked. The random value required to pick the best weight is computed using random.uniform() function, that is available in python. Finally, both best bias and best weight are computed. After the weights are initialized using the BES Optimization Algorithm, the hidden layers are calculated. At this point, the long-term dependency issue of vanishing gradient and exploding gradient is generated.

LSTM is used to resolve the issue. Forget gate, input gate, and output gate are calculated, and hidden vectors are updated. The back propagation is done using the "Adam" optimizer (available in the "keras" module of python). The stop criterion is set to 0.0001, so as to achieve minimum error. If the stop criterion is reached, then the execution stops and resultant values are passed on to classification phase, else the procedure is iterated again. During the ARNN model implementation, the LSTM dropout is 0.2. Dropout is a technique used to prevent a model from over fitting. Dropout works by randomly setting the outgoing edges of hidden vectors to 0 at each update of the training phase. The activation function used is softmax (given by equation (4)). Figure 7 explains the implementation of ARNN model briefly.

4.5 Phase 4: Classification

At output layer, when the stop criterion is met, finally the computed activation value for the test set is compared with the training set and the ratings are classified as genuine or an attack. The genuine rating is labelled as "NORMAL," higher rating "5" is labelled as "PUSH ATTACK," and the lower rating "1" is labelled as "NUKE ATTACK." The "NORMAL" ratings are stored in the database to generate recommendations, while "PUSH" and "NUKE" attacks are discarded.

The Pseudo code for ARNN model is given as follows:

Pseudo code 1: Implement ARNN

1. Collect user ratings and user reviews of various products from dissimilar users.
2. Apply pre-processing steps (stop word removal, lower case, punctuation removal, and lemmatization) to eliminate dissimilar products.
3. Apply feature extraction using Word2Vec–SGram technique
4. Apply MDPCA to form clusters and find most preferable user from every cluster.
5. Apply procedure RNN_BES (call pseudo code 2).
6. Classify the user ratings – "NORMAL," "PUSH," "NUKE."

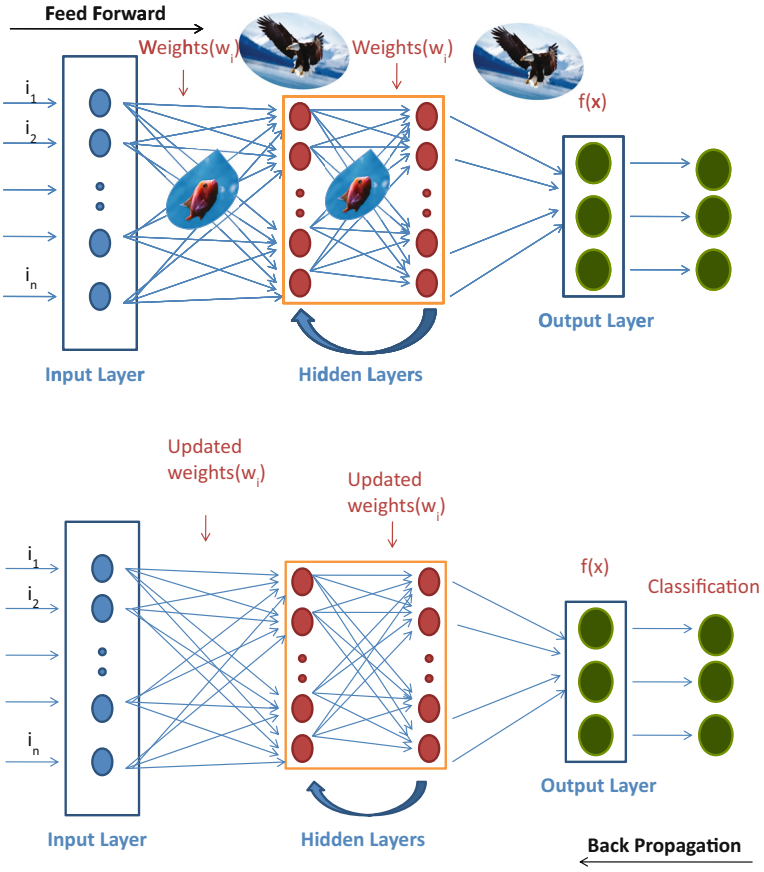

Figure 7: Proposed ARNN model for HRS.

Pseudo code 2: Forward and Back Propagation of ARNN

Step 1: Initialize: Input, Output, and Hidden Vector
Step 2: Update Weight and Bias by BES
\\ set the fitness value
\\ Pick the best bias using equation (11)
\\ Pick the best weight using equation (12)
\\ Both bias and weight are calculated using equation (13)
\\ Check the fitness using equation (10), if the condition is satisfied or not
Step 3: Calculate hidden layer by equation (1)
\\ Long-term dependency issue is generated
\\ Exploding gradient and vanishing gradient
Step 4: Enhance the performance by LSTM
\\ Calculate the forget gate using equation (3)
\\ Calculate the input gate using equation (6)
\\ Calculate the output gate using equation (8)
\\ Update the hidden using equation (9)
\\ If satisfied it will stop, or else it will continue
\\ else
Stop

5 Performance analysis

The proposed model has been implemented in the python platform and performance is evaluated with respect to Precision, Recall, *F*-measure, and Accuracy measures of the proposed methodology. The existing methods [26] deep neural network (DNN), multiview ensembled detection algorithm (MV-EDM), convolutional neural network (CNN), and stacked denoising autoencoder principal component analysis (SDAE-PCA) are compared with our proposed ARNN model.

5.1 Dataset

Dataset in https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products is used. There are about 35,000 consumer reviews for Amazon electronic products such as Kindle, Fire TV Stick, and so on. The dataset includes basic product data, review text, rating, and more for each product. The user ratings are collected in the rating scale of 1–5. There are 22 attributes in the dataset. Out of the available 22 attributes, userId, userReview, and userRating are used for attack detection. The Review data type is text and user rating data type is integer. The platform used for the experiment is Python.

The algorithm enters into the select stage and the best bias is found as 25–75%. The model was created with 75% of the data and the test data use 25% for the best fit model.

5.2 Experiment and results

First, necessary packages "keras," "pandas," and "genism" for the proposed model are imported. "Pandas" package is used to read the dataset. Pre-processing is done by replacing NAN cells into 0 and removing unwanted data. Feature extraction by word embedding technique Word2Vec is done by using "genism" package. After that, clustering of data using MDPCA is done. Finally, the clustered data are fed into ARNN for classification. After the classification, Push and Nuke ratings are discarded from the dataset and recommendations are generated from genuine ratings. Performance measures are calculated for existing algorithms and their results are compared with the proposed ARNN algorithm. Table 3 shows the result for various performance measures for both proposed (ARNN) and existing techniques (DNN, RNN, MV-EDM, CNN, and SDAE-PCA). It depicts the accuracy performance for both proposed and existing techniques. As a measure of recommendation quality, accuracy measures the degree to which recommendations are in alignment with the user's preferences. The accuracy performance of the proposed method is 98.66% and for the existing methods it is as follows: SDAE-PCA is 82.60%, MV-EDM is 85.28%, CNN is 88.94%, DNN is 92.75%, and RNN is 94.28%. The proposed method ARNN gives better accuracy results (98.66%) than other existing methods.

Table 3: Performance evaluation

Performance	Accuracy (%)	*F*-Measure (%)	Precision (%)	Recall (%)	FAR (%)
SDAE-PCA*	82.60	87.76	83.67	89.10	26.78
MV-EDM*	85.28	90.54	87.01	90.12	21.43
CNN*	88.45	92.34	89.64	93.67	18.90
DNN*	92.75	92.83	91.85	93.84	7.24
RNN	94.28	94.20	93.89	94.67	5.74
ARNN (Proposed)	98.66	98.67	98.50	98.83	1.33

*Reference: [26].

As a concept *F*-measure is calculated by taking the harmonic mean of recall and precision measures. The significance of using *F*-measure is that the harmonic mean penalizes extreme values as compared to average means. The *F*-measure performance of the proposed method is 98.67% and that of the existing methods – SDAE-PCA is 87.76%, MV-EDM is 90.54%, CNN is 92.34%, DNN is 92.83%, and RNN is 94.20%. The value of proposed method ARNN is 98.67%, which is the highest among other existing methods. This implies that the proposed method ARNN is better than the existing methods.

In recommendation systems, it is desired to recommend only top N products to the user. To achieve this, precision metric is computed by considering topmost N results. The precision performance of the proposed method is 98.50% and for the existing methods it is as follows: SDAE-PCA is 83.67%, MV-EDM is 87.01%, CNN is 89.64%, DNN is 91.85%, and RNN is 93.89%. The proposed method ARNN got 98.50%, which is the better precision result than other existing methods.

Similarly, Recall is the percentage of true documents that are successfully retrieved. The recall performance of the proposed method is 98.83% and for the existing methods it is as follows: SDAE-PCA is 89.10%, MV-EDM is 90.12%, CNN is 93.67%, DNN is 93.84%, and RNN is 94.67%. The recall value for the proposed method ARNN is 98.83%, which is best among the existing methods.

Another metric, false alarm rate (FAR) is defined as a proportion of number of wrong alarms to total number of warnings. Table 3 shows that the FAR performance measure of the proposed method ARNN is 98.67% and for the existing methods it is as follows: SDAE-PCA is 26.78%, MV-EDM is 21.43%, CNN is 18.90%, DNN is 7.24%, and RNN is 5.74%. It can be concluded that the proposed method ARNN has least FAR than other existing methods and thus is best of all.

Figure 8 shows the comparative analysis of various performance measures – accuracy, *F*-measure, precision, recall, and FAR for existing methods – DNN, RNN, MV-EDM, CNN, and SDAE-PCA and the proposed method ARNN. Figure 9 represents the receiver operating characteristic curve for the proposed ARNN model. It is clear from the graph that the proposed method ARNN has highest accuracy, precision, recall, and *F*-measure values and least FAR value as compared to other existing methods DNN, RNN, MV-EDM, CNN, and SDAE-PCA.

5.3 Computational complexity

The computational complexity of the proposed ARNN model is $O(n)$, where n is the length of the input sequence. Table 4 shows the computational complexity of ARNN model. The result shows that the ARNN model has less computational complexity because of high convergence speed of the BES algorithm.

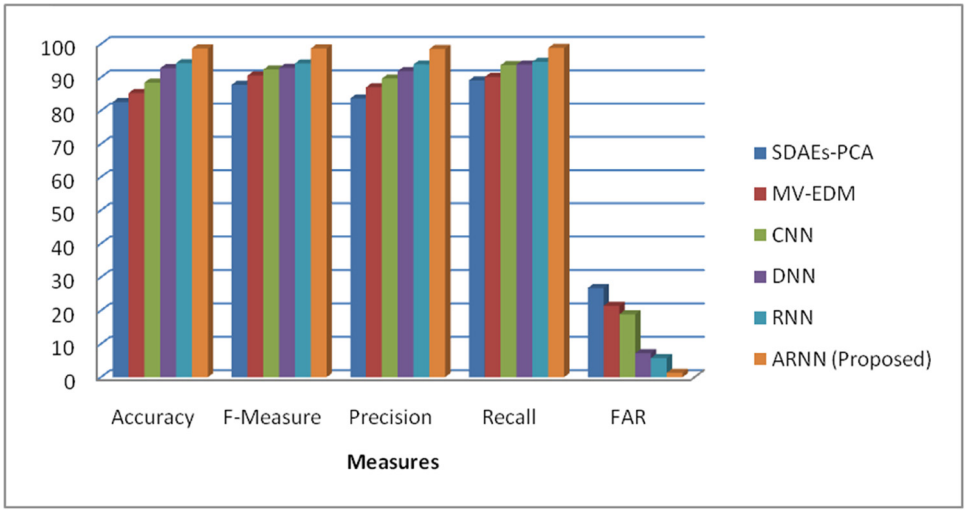

Figure 8: Performance analysis of HRS.

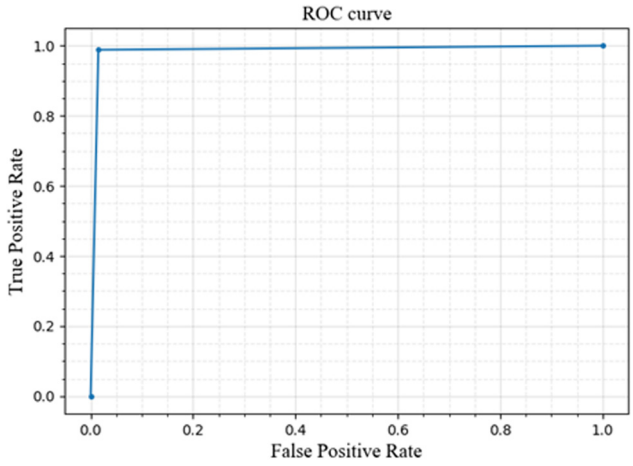

Figure 9: Receiver operating characteristic curve.

Table 4: Computational complexity of ARNN model

Technique	Input size (No. of runs)				
	5 (%)	**10** (%)	**15** (%)	**20** (%)	**25** (%)
ARNN	5.1	6	7.5	9	11.2

5.4 Limitation of research work

This article confines the work to detect shilling profiles (push attacks and nuke attacks) in HRS but not on multi-criteria CFRSs. The multi-criteria CFRSs take into account multiple views collected on different features of a product. Thus, wrong recommendations are viable for the fraud reviews. The proposed ARNN algorithm can be extended to deal with such system.

6 Conclusion and future direction

An RS predicts recommendation for an product to its users. The injected shilling attack, that is push attack or nuke attack either aggravates or degrades the performance of the RS. This results in false recommendations of products to the users. In this article, ARNN algorithm is proposed to detect push and nuke attacks for online product in CB Collaborative Filtering RS. The performance is evaluated using various measures such as precision, recall, *F*-measure, FAR, and accuracy with values 98.50, 98.83, 98.67, 1.33, and 98.66%, respectively. It is also concluded that the proposed algorithm outperforms all other existing methods (DNN, RNN, MV-EDM, CNN, and SDAE-PCA).

The practical advantage is to use this model in ecommerce industry. This helps to find the reason that why the user likes the product, additional to how much the user likes the product. When the proposed work would be implemented on multiple-views, it is expected that stronger multi-criteria CFRS would be achieved. A more strong system would definitely yield a more secured system. Our future work will try to address these issues.

The research can be further used in future to remove the fraud recommendations and it can be used as more secure system. Also, the data will be more accurate and will provide the accurate results

Conflict of interest: There is no conflict of interest.

References

1 Pennacchiotti M, Gurumurthy S. U.S. Patent No. 10,552,488. Washington, DC: U.S. Patent and Trademark Office; 2020.
2 Shaw PA, Heiland P, Tillmann RW, Maas HJ. U.S. Patent No. 10,555,045. Washington, DC: U.S. Patent and Trademark Office; 2020.
3 Chakrabarty S, Banik S, Islam MR, Sarma HKD. Context-aware song recommendation system. Trends Commun Cloud Big Data. 2020;157–65.
4 Shen J, Zhou T, Chen L. Collaborative filtering-based recommendation system for big data. Int J Comput Sci Eng. 2020;21(2):219–25.
5 Hu P, Gerard JP, Bernard S, Chauhan R. U.S. Patent No. 10,592,956. Washington, DC: U.S. Patent and Trademark Office; 2020.
6 Oswari T, Yusnitasari T, Kusumawati RD, Mittal S. Design and test music recommendation system for online music websites using collaborative filtering approach. Int J Digital Signals Smart Syst. 2020;4(1–3):64–79.
7 Erkek M, Çayırlı K, Taş H, Hepsen A, Aytekin T. Recommendation systems applied in Turkish real estate market. J Comput Model. 2020;10(1):1–10.
8 Pradhan R, Khandelwal V, Chaturvedi A, Sharma DK. Recommendation system using lexicon based sentimental analysis with collaborative filtering. International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC). IEEE; 2020.
9 Chopra AB, Dixit VS. Performance analysis of collaborative recommender system: a heuristic approach. International Conference on Information, Communication and Computing Technology. Singapore: Springer; 2019.
10 Davagdorj K, Park KH, Ryu KH. A collaborative filtering recommendation system for rating prediction. Advances in intelligent information hiding and multimedia signal processing. Singapore: Springer; 2020. p. 265–71.
11 Zhang G, Cao Q. An empirical study of E-commerce users' positive behavioural intention to personalized recommendation system. 6th International Conference on Humanities and Social Science Research (ICHSSR 2020). Atlantis Press; 2020.
12 Arora M, Chopra AB, Dixit VS. An approach to secure collaborative recommender system using artificial intelligence, deep learning, and blockchain. Intelligent communication, control and devices. Singapore: Springer; 2020. p. 483–95.
13 Piletskiy P, Chumachenko D, Meniailov I. Development and analysis of intelligent recommender system using machine learning approach. In Integrated Computer Technologies in Mechanical Engineering. Springer, Cham; 2020. p. 186–97.
14 Joshi S, Dubey J. Restaurant recommendation system based on novel approach using k-means and Naïve Bayes classifiers. Proceedings of the Global AI Congress 2019. Singapore: Springer; 2020.
15 Cui Z, Xu X, Fei XU, Cai X, Cao Y, Zhang W, et al. Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Trans Serv Comput. 2020;13(4):685–95.
16 Aggarwal S, Goswami D, Hooda M, Chakravarty A, Kar A. Recommender systems for Interactive Multimedia Entertainment. Data Visualization and Knowledge Engineering. Cham: Springer; 2020. p. 23–48.
17 Brundage M, Miikkulainen R. U.S. Patent No. 10,606,885. Washington, DC: U.S. Patent and Trademark Office; 2020.
18 Pan J, Pottimurthy Y, Wang D, Hwang S, Patil S, Fan LS. Recurrent neural network based detection of faults caused by particle attrition in chemical looping systems. Powder Technol. 2020;367:266–76.
19 Alsattar HA, Zaidan AA, Zaidan BB. Novel meta-heuristic bald eagle search optimisation algorithm. Artif Intell Rev. 2020;53(3):2237–64.
20 Sasikala P, Mary Immaculate Sheela L. Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS. J Big Data. 2020;7(1):1–20.
21 Levi A, Mokryn O, Diot C, Taft N. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. Proceedings of the Sixth ACM Conference on Recommender Systems; 2012. p. 115–22.
22 Mehta H, Bhatia SK, Bedi P, Dixit VS. Collaborative personalized web recommender system using entropy based similarity measure. arXiv preprint arXiv:1201.4210; 2012.
23 Revathy R. A hybrid approach for product reviews using sentiment analysis. Adalya J. 2020;9(2):340–3.
24 Osman NA, Noah SAM, Darwich M. Contextual sentiment based recommender system to provide recommendation in the electronic products domain. Int J Mach Learn Comput. 2019;9(4):425–31.
25 Song Y, Li G, Ergu D. Recommending products by fusing online product scores and objective information based on prospect theory. IEEE Access. 2020;8:58995–9006.
26 Hao Y, Zhang F, Wang J, Zhao Q, Cao J. Detecting shilling attacks with automatic features from multiple views. Secur Commun Netw. 2019;2019:2019–113.
27 Musto C, de Gemmis M, Semeraro G, Lops P. A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. Proceedings of the Eleventh ACM Conference on Recommender Systems; 2017. p. 321–5
28 Dong R, O'Mahony MP, Schaal M, McCarthy K, Smyth B. Sentimental product recommendation. Proceedings of the 7th ACM Conference on Recommender Systems; 2013. p. 411–4.
29 Koukourikos A, Stoitsis G, Karampiperis P. Sentiment analysis: A tool for rating attribution to content in recommender systems. RecSysTEL@ EC-TEL; 2012. p. 61–70.

30 Singh VK, Mukherjee M, Mehta GK. Combining a content filtering heuristic and sentiment analysis for movie recommendations. International Conference on Information Processing. Berlin, Heidelberg: Springer; 2011. p. 659–64.
31 Musto C, Semeraro G, Gemmis MD, Lops P. Learning word embeddings from Wikipedia for content-based recommender systems. European Conference on Information Retrieval. Cham: Springer; 2016. p. 729–34.
32 Morid MA, Shajari M, Hashemi AR. Defending recommender systems by influence analysis. Inf Retr. 2014;17(2):137–52.
33 Kumar A, Teeja MS. Sentiment analysis: A perspective on its past, present and future. Int J Intell Syst Appl. 2012;4(10):1.
34 Arazy O, Kumar N, Shapira B. Improving social recommender systems. IT Prof. 2009;11(4):38–44.
35 Alahmadi DH, Zeng ISTSXiao-Jun. ISTS: Implicit social trust and sentiment based approach to recommender systems. Expert Syst Appl An Int J. 2015;42(4222):8840–9.
36 Li F, Wang S, Liu S, Zhang M. Suit: A supervised user-item based topic model for sentiment analysis. Twenty-eighth AAAI Conference on Artificial Intelligence; 2014.
37 Leung CW, Chan SC, Chung FL. Integrating collaborative filtering and sentiment analysis: A rating inference approach. Proceedings of the ECAI 2006 Workshop on Recommender Systems; 2006. p. 62–6.
38 García-Cumbreras MÁ, Montejo-Ráez A, Díaz-Galiano MC. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. Expert Syst Appl. 2013;40(17):6758–65.
39 Burke R, Mobasher B, Williams C, Bhaumik R. Classification features for attack detection in collaborative recommender systems. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006. p. 542–7.
40 Bhaumik R, Mobasher B, Burke R. A clustering approach to unsupervised attack detection in collaborative recommender systems. Proceedings of the International Conference on Data Science (ICDATA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp); 2011. p. 1.
41 Serrano-Guerrero J, Olivas JA, Romero FP, Herrera-Viedma E. Sentiment analysis: A review and comparative analysis of web services. Inf Sci. 2015;311:18–38.
42 Abujar S, Masum AK, Mohibullah M, Hossain SA. An approach for Bengali text summarization using word2vector. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE; 2019. p. 1–5.