Deep Learning-based Automated Recommendation Systems: A Systematic Review and Trends

Naveena Koneru[1] , Sanjana Rai[1], Guruprakash J[1], Sateesh kumar S[2], Venugopal P[2], Srinivas Koppu[1*]

[1] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore

[2] School of Electronic Engineering, Vellore Institute of Technology, Vellore

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Recommender systems are progressively playing larger roles in a business's transactions, profits, and overall success. This survey is intended to demonstrate the various aspects of recommender systems and their methodologies for implementation. As the purpose of a recommender system varies depending on the business's necessities, its components and attributes vary accordingly. In this survey, important attributes of recommender systems are discussed, along with design guidelines. Some widely recognised methodologies are examined. Lastly, movie recommenders of the three most relevant domains, movies, music, and e-commerce, are introduced. The survey aims to give readers an expansive view of the situations in which certain recommender systems would be appropriate.

Keywords: Recommender systems, Knowledge Representation, Artificial Intelligence in Recommender Systems, Collaborative Filtering, Ontology, Hybrid Systems

1. Introduction

Recommender systems have played an increasingly significant role in the success of e-businesses in the past 20 years. A plethora of information, and in e-commerce platforms, products are easily available to a user. However, this abundance of information frequently serves as a deterrent for the user to gain access to what they need. The role of recommenders is to offer a useful and reduced set of information that will meet the desires and needs of the user (Sinha & Swearingen, 2002). It makes sense that some filtering, grouping, and scoring methods are integrated into the system to output the most relevant information to the user. Figure 1 represents a generic recommendation system.

Recommender systems are most often seen in general e-commerce websites, such as Amazon, eBay, and Flipkart. These offer a limited, more relevant selection of items than those found in a manual search, and they are more likely to help a customer make a satisfactory buying decision. Users are well acquainted with entertainment recommenders, either in movie, book, or music domains. An entertainment recommender's success is crucial to maintain a user's interest in the platform, as the main purpose of the platform is to engage the user with attractive options (Swearingen & Sinha, 2002) continuously.

Netflix, an online movie streaming platform, relies greatly on its recommendation system, as approximately two-thirds of movies watched by viewers are recommended to them. Google News' recommendations prompt users to increase click-throughs by 38%. Amazon attributes 35% of its sales to recommendations, and Choice stream's survey showed that 28% of the customers had a willingness to purchase more music if they were appropriately recommended enjoyable options (Kumar, 2015).

The purpose of recommenders may vary. Some systems may simply seek to introduce various articles to widen a user's horizons by offering different suggestions to the user for an enjoyable experience. Some systems are geared towards recommending products that a customer is more likely to purchase and increasing the business's profitability. Studies show that users are less likely to purchase and commit to "new", unseen items upon recommendation. Thus, e-commerce businesses usually consider algorithms and attributes that help increase profitability, even if they must limit some other recommender system properties (Swearingen & Sinha, 2002).

Recommender systems are most commonly divided based on the information that is used in the filtering process. Collaborative filtering is widely used in real-time recommender systems. It relies on data about users' previous history and interactions with items or products offered. Using user ratings, feedback, and usage history, trends are found, which are then used to detect items of possible interest. A prerequisite to the success of this system is that there must already be a substantial amount of usage history present. New businesses will not have such information at their disposal. This is the cold-start problem (Vie, 2017). Another frequent setback of this method deals in data sparsity. Generally, a user interacts with and rates only a small proportion of items, while all others are left blank. This sparse data causes issues in making accurate recommendations that truly represent the user's preferences (Farokhi, 2016).

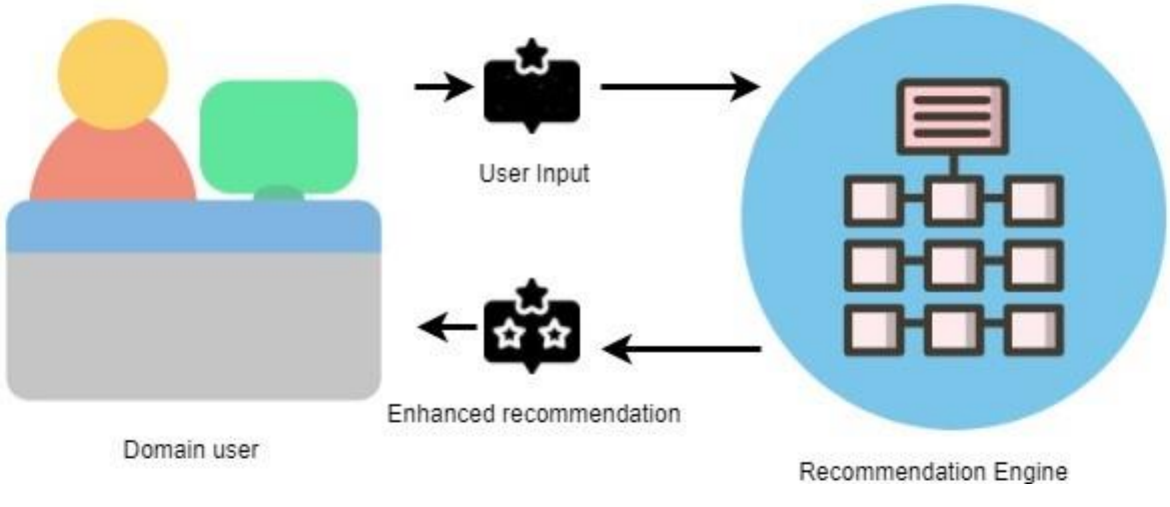
Content-based filtering focuses on the similarity between items that a customer has liked and other unseen items. The similarity is determined based on the items' descriptions and features.

Accuracy is often high in such algorithms, as the user only views items that are largely similar to those they have enjoyed in the past. However, the tradeoff for greater accuracy results in a limited diversity of products, resulting in over specialisation (Blanco-Fernandez, Pazos-arias, Gil-Solla, Ramos-Cabrer, & Lopez-Nores, 2008). Such systems have limited success, as they cannot further interest a user with novel or niche item recommendations that would still appeal to them (Zhou, 2010).

Hybrid systems combine various algorithms of content and collaborative methods to design a system that has the advantages of both individual methods. For example, an accurate, content- based recommender may be incapable of maintaining a user's interest if it repeatedly recommends too similar items. Methods to introduce serendipitous items can boost the recommender by allowing user to explore their interest (Iaquinta, 2008).

This survey was conducted by collecting 43 research papers related to the topic of recommender systems, with the goal of introducing and reviewing the various implementations of recommender systems. The survey is organised as follows - Section 2 reviews key desirable features of recommender systems. Section 3 introduces some design guidelines, and Section 4 reviews knowledge representation. Section 5 analyses common methodologies used by recommenders. Section 6 describes recommender systems of three domains: movies, music, and e-commerce. Lastly, Section 7 presents a conclusion and brief overview of the survey.

FIGURE.1 - Generic recommendation system



2.Attributes of Recommender Systems

2.1 Accuracy and Diversity. Accuracy in recommendations, the ability to predict a user's opinion correctly, while useful, is not the sole measure contributing to a recommender's success. Another attribute that must be considered is the system's ability to gather items that a user might show interest in but would not normally search for themselves. The recommendation of less conservative, novel items that can catch the user's attention has the added advantage of inspiring the user's enthusiasm in the business as a whole (Zhou, 2010).

(Zhou, 2010) creates a recommender system that balances the attributes of both accuracy and diversity. This paper proposes that recommending both niche, diverse items that specifically cater to an individual user's taste and similarity-based items (related by past purchases or similar users) enhances the system's performance. Thus, a hybrid recommender system that aims to find both an accurate and useful recommendation system is introduced by balancing both similarity and diversity. Five individual algorithms, including one hybrid algorithm, have been tested on three datasets of different domains: movie streaming data, music data, and internet bookmarking data. The algorithms used are: 1. HeatS is an algorithm working similarly to heat diffusion across a network utilizing average distribution of values. 2.ProbS use column normalized methods to evenly distribute some resource values to connected nodes at every level.

The hybrid algorithm uses both HeatS and ProbS to employ a process that is only as computationally complex as one of the two algorithms. If the hybrid parameter λ (in the matrix normalization) is 0, the HeatS algorithm is implemented. If λ is 1, the ProbS algorithm is implemented. Each proposed recommendation algorithm is evaluated based on performance metrics of accuracy, precision and recall, personalization, and novelty. ProbS are found to be the most accurate, whereas HeatS yields the most diverse results. The ProbS and HeatS processes are then combined into a single hybrid system that can manage both parameters. Results show a significant improvement among all datasets and in most all of the performance measurements, using the hybrid method over using only the ProbS method. Further extensions of the system can be made by combining the present hybrid system with the GRank algorithm (Zhou, 2010).

2.2 Stability. (Hou, 2017) focuses on addressing the concerns in (Zhou, 2010), with the additional concern of stability in a recommender. System stability refers to its ability to deal with noise. Specifically, there must be a steadiness between recommendations using the gathered user-item data and the recommendations generated using gathered user-item data and the original recommendations. In other words, assuming that users select some of the items recommended to them, the next round of recommendations must be relatively consistent with the previous ones. Stability with respect to item similarity is usually not satisfied, as item similarity is found to be vastly different among different samples within a period. In this case, the actual similarity between items is not represented within these systems. A heat conduction algorithm (HC) is used to determine item similarity. The starting item node begins with a temperature of 1, and the temperature is averagely distributed among the users it is connected. The temperature of each user is once again distributed among all of the items, where the final temperature of each item is equivalent to the similarity between the starting item and it. A TNS-HC algorithm divides a dataset into two randomly selected subsets, and item similarity using HC is calculated for each. The difference between subset similarities indicates the amount of similarity instability.

The two algorithms are tested on Movielens, Netflix, Last.FM, and Epinions datasets. In all datasets, the diversity measures show that HC provides more unpopular recommendations than TNS-HC. Unpopular recommendations decrease system stability. TNS-HC maintains a reason- able diversity while improving system stability and accuracy.

2.3 Transparency. Transparency deals with how well the user understands the reasons for recommending a particular product. Similar to real-world scenarios, users are more likely to give credibility to a process when they have insight into how it works. Work in (Sinha & Swearingen, 2002) is a study performed on the significance of transparency in the success of a recommender system. It involves user preferences and interfaces rather than recommendation algorithms. The study aims to determine whether increased transparency in the system increases users' trust in that recommender system. The experiment studies five music recommender systems that allow user interaction and have different interfaces. Twelve participants, each regularly involved in music purchase transactions, gave their opinions on given items and rated their understanding of each music recommendation.

Results found that the mean enjoyment and confidence are somewhat higher in systems that are considered to be more transparent. One aspect to consider is users' previous exposure to the same items; they may have never been exposed to an item, or they have, and they have either a positive or negative impression of the item. Upon studying the three cases separately, positively judged items were well-liked, negatively judged items were not, and previously unseen items had mixed opinions. Users preferred transparent recommendations for unseen and previously seen, positively judged items but didn't seem to care much for negatively judged items. Future extensions of this work involve testing the competence of different transparency methods.

2.4 Trust. Many practical systems rely on collaborative filtering based on the similarity of user preferences. In (O'donovan & Smyth, 2005), O'Donovan and Smith claim that the effective- ness of systems focused solely on similar users is over emphasized. Other significant elements, such as the trustworthiness of a user, may be incorporated into a traditionally used CF system. The trustworthiness of similar use to the target user can be measured based on the reliability of past recommendations to the same user (O'donovan & Smyth, 2005).

The trust measuring mechanism is incorporated into the CF system using varying combi- nations using Resnick's formula, namely, trust-based weighting, trust-based filtering, or a com- bination of both. Furthermore, the study tests whether item-level trust or profile-level trust offers better results. All of the tested models have a lower error rate than a standard Resnick's prediction model. The combination, item-level trust methodology has a reduced error of 22% compared to the standard. The trust-based filtering, profile-level trust methodology has the highest percentage of wins over the standard benchmark process.

2.5 Serendipity. (Iaquinta, 2008) broaches the aspect of serendipity in recommender systems. Serendipity differs from novelty/diversity described in (Zhou, 2010). It can be used to find information that a user is very unlikely to encounter on their own but may still appeal to them. Novelty introduces niche, different items to a user, but there is a substantial likelihood that a user may discover those items by themselves. Thus, serendipity can expand a user's horizons by recommending items that they don't know are appealing to them and combats the issues of over specialization and limited diversity. The study tackles the relatively unpredictable nature of serendipity by filtering out recommendations that are too different and too similar.

The algorithm is implemented on a dataset of 45 paintings, each containing textual information about the title, artist, and description. First, a content analyzer uses semantic indexing to identify meaningful concepts among the textual information. An appropriate Bayesian text classifier model is built during the training phase that outputs either a positive or negative response to each painting. Certain heuristics are defined to find the serendipitous recommendations. Similarity comparisons will find the top-N recommendations with a higher classification score that is positive than negative. Serendipity comparisons deal in the recommendations that have less difference between positive and negative scores, thus leaning strongly in neither the positive nor negative scale. The regular serendipity function was tested against the same with randomness and different thresholds. The leading results came from the random over serendipity function with the highest threshold of 15%.

2.6 Scalability. Most e-commerce platforms need to process millions of transactions and provide millions of product recommendations. Scaling up a system to provide millions of recommendations in a shorter amount of time generally leads to a deterioration in the effectiveness of recommended items. (Sarwar, 2002) uses scalable clustering techniques to enhance a recommender system's ability to quickly provide large numbers of personalized recommendations while also maintaining the quality of each recommendation. The study focuses on finding the system that balances the quality of recommendations with the speed of the system's output to find a recommender that is functional in real-world e-commerce applications. The details of this study are further discussed in Sections 5.2 and 6.3.

2.7 Profitability. A business that employs a recommender system has a primary goal to in- crease its profits. The recommender system is not an essential feature for users to find what they need, nor does it play a role in the e-business transaction itself. It is a component that is used to improve profits by attempting to fulfil user needs. This is done by providing recommendations of interest that will encourage users to purchase more.

As investigated in (Azaria, 2013), an important challenge in recommendation is identifying the items that users are actively interested in, enough for them to complete a purchase. The paper proposes a movie recommender system to increase profits. The authors show that by providing recommendations that serve the businesses' needs but may not be the best from the users' point of view, the business may gain substantially. Two settings are considered while performing experiments: The Hidden Agenda setting, in which a business has items to promote in a manner not visible to the user, and the Revenue Maximizing setting, in which it is the recommender system's goal to maximise the revenue.

The proposed system 'mounts a black-boxed recommender system which supplies a ranked list of movies. In the Hidden Agenda setting, the proposed system outperforms the original recommender system considerably by increasing its promotion value by 57%. The Revenue Maximizing setting also showed a considerable increase in profits, up to USD 1.71 from USD 1.33. Interestingly, people are more willing to pay for a movie they have watched and are familiar with than one they have not watched and is new to them. This observation is consistent with that in (Swearingen & Sinha, 2002), where users show a greater willingness to invest resources in highly similar and familiar recommendations.

3. Design Guidelines for Recommender Systems

S. Sheth et al. (Sheth, 2010) introduce a reference architecture, which acts as a blueprint for creating social recommender systems. The architecture is aimed towards community-driven recommender systems, which means they automatically derive recommendations from the systems' user database, thereby taking into account community behaviours and interests. Furthermore, the architecture is not dependent on any particular domain or application. The social recommender systems developed using this architecture are not static, as they observe their users learn and refine the recommendations. The reference architecture consists of three main components/modules. These are 1. The Watcher, which monitors users' activities, 2. The learner, which infers patterns and recommendations and 3. The Advisor, which uses different UI options to provide recommendations. Other components include the database component used to store user activities and the Security and Privacy layer. The component structure of the reference architecture for social recommender systems offers a template for designing modularised recommender systems which in turn may lead to standard interfaces and interoperability among components of recommender systems. A limitation of the proposed reference architecture is that unless systems use any knowledge gained from usage data, they cannot be mapped to the proposed architecture. The authors aim to expand on and provide more concrete modules for the Security and Privacy layer for future work.

(Swearingen & Sinha, 2002) builds upon (Sinha & Swearingen, 2002), in which the importance of transparency for reliable recommendations is established. This paper aims to gain insight into elements other than accuracy that contribute to satisfaction with a recommender system and use the information gathered to offer design guidelines for an effective, user-interactive system. The input and output stages of systems in the music domain are analysed. Such systems include Amazon, Media Unbound, and CD Now. Most systems require some explicit entry of the user's preferences in ratings and genre filtration. The initial number of ratings that the system gathers may vary from one to thirty. The method of gathering reviews may vary from a like/dislike option, open-ended questions, or a numerical rating scale. The output stage must consider the number of items to recommend for each scenario, the optimal amount of information to display about the product. Additional factors to consider are the familiarity of recommendations and the transparency of the recommendation process.

The study recommends maintaining a balance on the number of ratings to obtain from the user to yield maximum accuracy and cause minimum inconvenience to the user. Mixing up the methods of gathering input is a favoured technique to maintain user interest. The recommendations provided must allow for flexibility in the number displayed and contain only concise information with an audio or visual aid. The analysis of Amazon and Media Unbound music systems reveals that the appropriate methods must be applied to the system based on its purpose. This is vital to determining its success. Amazon's recommender is successful with respect to e-commerce and making a sale, as it offers very similar items with high transparency and a detailed but simple interface. This observation corresponds to that made in (Azaria, 2013), where recommending somewhat familiar items result in increased purchases and profits. On the other hand, Music Unbound succeeds as a tool to learn more about one's taste and items of interest. It offers an engaging recommendation process and a music choice of less familiarity and transparency than Amazon. However, Music Unbounds direct purpose is not to make a sale, so its recommendations need not be as conservative as Amazon's recommendations.

(Pu, Chen, & Hu, 2011) details a unifying evaluation framework, intended to measure the 'qualities of the recommended items, the system's usability, usefulness, interface and interaction qualities, users' satisfaction with the systems, and the influence of these qualities on users' behavioural intentions, including their intention to purchase the products recommended to them and return to the system.' The study also applies psychometric methods to validate the combined criteria using data collected from a user survey (of 239 participants). The evaluation questionnaire has the question items structured into four layers to more clearly express how the users' perception of the physical features of a system influences their belief. The four layers are of higher-level constructs and are the perceived system qualities, users' beliefs resulting from these qualities, their subjective attitudes, and their behavioural intentions. The paper combines and validates existing criteria into a well-balanced user-centric evaluation framework for recommender systems. For future work, the authors aim to analyse further and collect user data, to understand cultural influences and influence users' attitudes and behaviours.

4.Knowledge Representation in Recommender Systems

4.1 Knowledge Base. Knowledge bases help a recommender system improve in quality by drawing semantic information from a structured or unstructured source.

(F. Zhang, 2016) In a paper detailing Collaborative Knowledge Base Embedding for Recommender Systems, the authors investigate how to leverage the heterogeneous information in a knowledge base to improve the quality of recommender systems. They have designed three components to extract items' semantic representations from three types of content: structural, textual, and visual. Their recommendation system combines collaborative filtering with semantic representations of items from the knowledge base. The first step is knowledge base embedding: structural embedding, textual embedding and visual embedding, and the second step is collaborative joint learning. Then, the proposed framework is assessed on two real- world scenarios. After studying textual knowledge usage, structural knowledge usage and visual knowledge usage, the whole framework is studied. The result shows the proposed recommendation system to be very effective and outperform various extensively adopted recommendation methods.

(Hsu, Hwang, & Chang, 2010) Hsu, Chang, and Hwang implement a recommender system that suggests appropriate reading resources to increase student's proficiency in the English language. In this situation, each student is at a different level of reading proficiency, even if they are within the same class. Assigning the same reading material to all students may even be counterproductive to their learning progress and interest. The recommender system emulates expert-level recommendations and offers options to the student's abilities, areas of interest, and the characteristics of each of the available articles. Such rules for recommendation are stored in a knowledge base. The development process involves gathering the knowledge of experts by having them rate 92 articles based on the relevant topics present. Twenty-five students were also given ratings based on their affinity for each topic listed. This previously gathered data is arranged into two repertory grid matrices, one of the student preferences and the other articles characteristics. A fitness analysis formula is created to best fit student preferences with the article topics. If the distance between the student and article is within bounds, the fitness degree is calculated. The articles with the highest fitness degree will be recommended to the particular student. After implementing the system, 94.59% of the students felt encouraged to learn, as the system recommended appropriate articles corresponding to their level of proficiency. The results also represent the real-world situation of English classes in Taiwan, where the study was conducted, as students of a class tend to be much more advanced or much less proficient than the level of teaching in class. By comparing the recommended articles with the expert teacher's opinion of suitable articles for each student, the overlap in recommended articles proves that the system can provide recommendations similar to experts' opinions.

4.2 Ontology. An ontology aids in the hierarchical structure of items and detects underlying relationships between them (Middleton, Shadbolt, & Roure, 2004). In (Middleton et al., 2004), ontological studies are employed to build better user profiles for a research paper recommender. Rather than collecting ratings or usage of binary class models to build a user's profile, unobtrusive data collection is preferred by academics who are the primary users of the system. Two-hybrid recommender systems with a user profile visualised in ontological methods are tested. Quickstep collects browsing activity and behaviour to classify browsed articles based on their topic. Recommendations are generated based on the correlation between the researcher's interest and the classes of articles. The use of an ontology resulted in a substantial increase in the acceptance of the topics of articles. Foxtrot extends Quickstep by allowing for search and profile visualisation.

(Blanco-Fernandez et al., 2008) contrasts the benefits of semantic versus syntactic-based similarity and their performance in relation to recommendation diversity and over specialisation, a common result of content-based filtering. The area of focus is a content-based recommendation from semantic connections and a defined ontology in the domain of Interactive Digital Television. Ontologies allow for a structure of domain-relevant terms and their relationships to be organised hierarchically. The Semantic Web provides a platform to develop an ontology with various available languages. To create semantic ontology information of each TV program from the TV-Anytime metadata and an appropriate language is used. The ontology is utilised to model users' profiles with descriptions of the TV programs the user has watched. In the filtering phase, the related nodes of appealing shows are found in the ontology and selected. TV shows with common ancestors are compiled into a possible list of TV show recommendations. Semantic as- sociations further filter the shows to determine if they will meet the preferences of the user. TV shows with the greatest relevance scored are then recommended to the user. The user benefits from an increased variety of TV shows that have some underlying related themes to previously enjoyed shows that may not be visible at first glance. Thus, an appropriately diversified selection of TV shows is made available while maintaining accuracy and avoiding over specialisation.

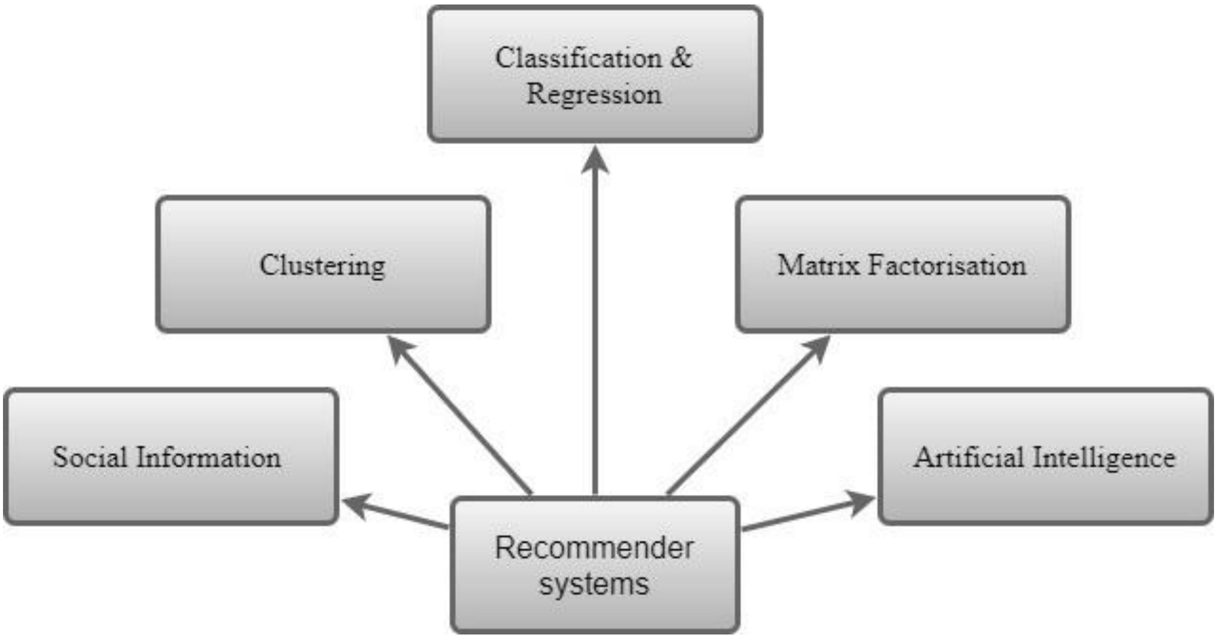
(Bahramian & Abbaspour, 2015) Recommender systems in the tourism industry allow tourists and business to benefit if they are pointed in the correct direction, as transactions comprise a significant amount of money, time, and effort. Points of interest (POI) for a user are determined by using ontological information. The ontology aids in making connections about the users' enjoyments and attributes of POIs. A spreading activation strategy is used to learn user profiles by iteratively advancing scores on preference and confidence throughout the ontology. The system created has three components, the first being an ontology-based content analyser. The second component learns from user profiles, and the spreading activation process assigns preference values to each node in the user profile. The third component considers semantic similarities to create a final list of recommendations. The tested method returns a list of diverse recommendations to eradicate over specialisation within the recommender system.

In (Guia, Silva, & Bernardino, 2019), the authors propose a hybrid recommender system that combines the KNN algorithm and ontology. They create and use an ontology 'to implement a data model that can describe and represent all the concepts that concern the active user, his neighbours and the products purchased by the latter, which are the ones that can be recommended to the active user'. The proposed recommender system is discussed in detail in section 6.3.

5. Methods for Recommender Systems

There are various methods in which recommendation systems are modelled. Domain usage and attributes present in them play a key role in determining the method of choice. A representation of our review is based on the Figure 2.

Figure **2** Recommender system methods



5.1 Social Information-Based Recommender Systems. In (Basu, Hirsh, & Cohen, 1998), the authors discuss the recommendations based on two types of information: social-based information and content-based information. Their goal is to create a recommendation system that uses both social and content-based information. They frame the problem as one classification instead of artefact rating, thus departing from the conventional social-filtering approach. When it comes to content-based rating, they use social information in the inductive learning process. They describe the collaborative, content and hybrid features that serve as a framework for their recommendation system. After different recommendation approaches, they obtained a precision of 78% and a recall of 33%, a precision of 77% and a recall of 27% (for no content), a precision of 73% and a recall of 33% (for simple content), and a precision of 83% and a recall of 34% (for hybrid features). The proposed inductive approach to recommendation has an advantage over other social-filtering methods in that it is more flexible.

In (Ma & Hao, 2011), a user's social information is adopted into a matrix factorisation recommender model. It compares and contrasts between social-based and trust-aware models. Trust relationships in recommender models are based upon one-sided ratings about other user recommendations. In contrast, social relationships rely on mutual contact and communication. In the experiment, social data is used to limit the results of the matrix factorisation process by means of regularisation. Two types of regularisation, average-based and individual-based, are implemented upon Douban's social networking and recommendation website, as well as Epinions, a customer reviews website that maintains a "trust list". By comparing with a com- mon trust-aware system RSTE using evaluation metrics of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), both proposed social regularisation methods performed better with a lower MAE and RMSE value. Of the proposed methods, the individual-based social regularisation consistently has a higher performance. Furthermore, the response of the two datasets differ with respect to the algorithm: using the item means to predict missing values achieved better results in Douban, whereas user mean achieves a better result on the Opinions data.

For future research, the study proposes using only a choice group of friend recommendations, not all of them, to model the real world more accurately.

In (Kazai, Yusof, & Clarke, 2016), the authors propose a prototype that offers personalised content recommendations by combining various factors like the users' location, Facebook and Twitter feeds, and their in-app actions so as to deduce their interests. They aim to build individual models for each user and location. The user's feed comprises items related to the user's interests, combined with locally popular and trending stories, and content derived from other sources like the user's Twitter feed and other social network profiles. The system first processes various input streams of content from Twitter, Facebook and RSS. Once the user signs up and creates an account, their Twitter and Facebook feeds are analysed, and an early user model is built. Then, social media is monitored to identify trending content, which is location-specific. Each users' model is used to generate relevance scores for each page consumed. A user's feed consists of content from the model-based recommendations. The proposed system is the first to provide personalised feeds by combining model recommendations and various sources.

5.2 Clustering-Based Recommender Systems. (Sarwar, 2002) uses clustering to produce outputs more quickly to meet the real-world demands of large-scale e-commerce. A basic collaborative filtering algorithm of nearest neighbours using Pearson's correlation is used as the standard measure to compare against a clustering neighbourhood formation algorithm, namely, the bisecting K-means algorithm. The bisecting K-means algorithm continuously divides customers into several uniformly-sized neighbourhood clusters until the desired clusters are reached. The Mean Absolute Error measure, which calculates the deviation of a predicted recommendation from a customer's true preferences, is found to be higher in the bisecting K-means algorithm by a negligible amount. The bisecting K-means algorithm yields a greater number of recommendations per second than the standard collaborative filtering algorithm. Consequently, the proposed algorithm returns a much higher throughput of recommendations with similar quality when compared to a standardly used CF algorithm.

The research work in (Kumar, 2015) is a movie recommender system, further examined in Section 6.1, that implements a K-means algorithm to generate the top 20 movie recommendations. (Ahuja, Solanki, & Nayyar, 2019) implements a K-means algorithm that derives the distance measure from multiple attributes, namely the movie's genre, actor, director, year, and rating, each of which are weighted accordingly. In (Q. Li, Myaeng, & Kim, 2007), a music recommender system, a k-medoids algorithm, is used to build a fixed number of movie com- munities. Using an existing object as the representation of a cluster, rather than the centroid used in K-means, the k-medoids method helps mitigate the effect of outliers that can occur in the traditional K-means recommenders of (Kumar, 2015) and (Ahuja et al., 2019).

(Farokhi, 2016) incorporates not just a single user-item rating into its system but rather multiple ratings of different criteria to combat data sparsity and increase accuracy and scalability in a tourism recommender system. Fuzzy C-means clustering is adept at dealing with sparsity and, when used with collaborative filtering, allows a user to be placed into more than one cluster of similar users. The use of multiple criteria in rating an item allows users to be more specific about the attributes they like and dislike, whereas a single rating acts as an overall opinion of an item but gives no insight into what aspects of the item is preferred. The experiment is performed using 5000 users and their various ratings in TripAdvisor for the hotels they have visited. The data is organised into a user-hotel matrix for each criterion. Data pre-processing removes repetitive ratings and users or hotels that are "insignificant" to the matrix while also filling in missing values. The offline phase of the system performs training and testing of the data to learn the best possible model. During training, several fuzzy clusters of users are inputted into a neural network of 10 layers, which develops an approximator. During testing, the user's cluster is identified by the degree of cluster involvement, and the appropriate approximator is used to generate recommendations. The online phase then uses a rating given to the system to recommend another item to the user in real-time. Evaluation with MAE shows that the chosen method of similarity measure is a correlation, not cosine similarity. Using correlation in both K-means and C-means clustering methods yields a lower MAE for C-means clustering.

(B. Li, Liao, & Qin, 2014) incorporates a pre-clustering system that forms cluster divisions of movies preceding the actual recommendation system component. The movies are clustered using four different methods: affinity propagation, density-based clustering, hierarchical clustering, and random clustering. It generates a pertinent, consolidated list of movies and corresponding cluster labels. The affinity propagation method allows for a variable number of clusters based on the value in the diagonal part of the similarity matrix between movies. Density-based clustering tends to form clusters around areas of higher density. Each cluster comprises central movies that compose highly dense areas and other movies that, while not central movies, have some closeness with them. Hierarchical clusters gradually form through merging existing clusters, and random clustering assigns movies randomly to any of a predetermined number of clusters. The density-based pre-clustering recommendation system is found to yield the greatest accuracy. The complete recommendation system of this paper is discussed in Section 6.1.

5.3 Classification and Regression-Based Recommender Systems. (Vie, 2017) uses a linear regression component combined with a matrix factorisation component to identify a final rating for each item pertaining to the user. The linear regression block LASSO uses the tags derived from the item poster and parameters corresponding to that item to estimate the user's rating value of that item.

(Chapphannarungsri & Maneeroj, 2009) Multiple regression can be used to incorporate the effects of multiple features by weighting them accordingly. This implemented recommender system can provide better recommendations compared to an existing Thai-Music recommender that considers only the weights of the most significant parts in a feature while other components are lost.

(Basu et al., 1998) uses classification to learn whether or not a user would like or dislike a particular movie. Each movie is recommended to the user if the movie-user combination is classified as “liked”. The estimated movie rating is higher than the rating threshold that divides the training data ratings into the top 25% of ratings.

In (Ahuja et al., 2019), the k-nearest neighbour classifier involves a specified user and the most similar users to predict movie ratings.

5.4 Matrix Factorisation. Matrix factorisation in recommender systems allows for the capability of recognising implicit relationships and features among users and items by decomposing the user-item matrix with rating values. This can help predict ratings for other items.

As seen in (Shi, Larson, & Hanjalic, 2010), joint matrix factorisation can be used to combine a regular matrix factorisation method with movie similarity within a certain mood context. An objective function concurrently factorises the two matrices (user-item and movie similarity matrices), and a gradient descent optimisation can be applied to the objective function to minimise it.

A complete component for matrix factorisation, ALS, which outputs an estimated item rating, is merged with the linear regression LASSO component, which also outputs an item rating, to produce a rating value that has advantages of both the methods in (Vie, 2017). It is also possible to incorporate additional information, such as social information, into a matrix factorisation model by means of a minimising objective function, as seen in (Ma & Hao, 2011).

As mentioned above, matrix factorisation is an effective means for the recommendation. The ways of incorporating it are many; it can be used on its own, may be implemented as a separate component of the system, or may combine with other information with the use of an objective function.

5.5 Artificial Intelligence and Recommender Systems. (Cheng & Heng-Tze, 2016) Cheng et al. propose a solution to tackle over-generalisation by deep neural networks with embeddings, which results in less relevant items using wide and deep learning. They present a model to amalgamate the advantages of memorisation and generalisation with jointly trained wide linear models and deep neural networks. Memorisation is the learning of frequent co-occurrence of items or features in the past and available in the data. In contrast, generalisation explores new combinations that have never or rarely occurred in the past and are not already available in the data. In this paper, the authors evaluated and focused on the app’s recommendation problem for the Google Play store. For such large-scale systems, linear models such as logistic regression are widely used. In their Wide and Deep Learning Recommender System, the wide component is a generalised linear model, and the deep component is a feed-forward neural network. They combine the two components “using a weighted sum of their output log odds as the prediction, which is then fed to one common logistic loss function for joint training. “To test the efficiency of their model, they ran live experiments and assessed the system on app acquisitions and serving performance. The results demonstrate that the Wide and Deep framework substantially betters the app acquisition rate compared to wide-only and deep-only models.

(Wang, Wang, & Yeung, 2015) In this study, deep learning algorithms are implemented on item content information and joined with a collaborative filtering algorithm conducted on user ratings. It is hypothesised that the system will be capable of combating the sparsity problem that plagues many CF recommender systems, where the lack of information of most movie ratings between users can cause the system performance to degrade. The proposed algorithm, CDL, uses tightly coupled processes to ensure a two-way information flow between the supplementary content information and the traditional CF model. Bayesian SDAE is the neural network with a hidden layer that filters out corrupted input values and outputs a cleaner subset of the inputs. The results of the system implemented on three datasets, one from Netflix and two from CiteULike, are used to emulate real-world situations where data may be of different sparsity levels. All three datasets with the CDL algorithm yield a higher mAP and recall compared to state-of-the-art systems. The number of layers to add to the system vary in accordance with the highest recall value among all the layers. The recommender performs well in the second CiteULike dataset, which has greater data sparsity than the first. Thus, the recommender is capable of functioning accurately with data of varying sparsity levels.

(J. Zhang, 2019) Zhang et al. propose a Stacked and Reconstructed Graph Convolutional Networks architecture to boost the performance of recommender systems, especially in the cold- start scenario, by means of learning node representations. The proposed model learns low- dimensional user and item latent factors as the input to the network in an end-to-end model and restrains the model space complexity. Node embeddings are produced for new nodes by reconstructing masked input node embeddings. A block of graph encoder-decoder is used to reconstruct masked embeddings in the training phase. These reconstructed masked embeddings are used to generalise the model and improve the learned embeddings to predict embeddings of unseen nodes to tackle the cold start problem. Experiments are conducted over two tasks: transudative rating prediction, which is used in matrix completion tasks, and inductive rating prediction, which is used to evaluate the ability of different models on the cold start problem. The model's architecture is a multi-block graph encoder-decoder that enables the revaluation of initial estimates and features. Each block contains one encoder, which generates node representations, and one decoder, which recovers the input node embeddings. In the encoder, the node representations are generated by encoding semantic graph structures and input content features. After graph encoders, a task-specific loss is imposed, and after decoders, a node reconstruction loss is imposed. For the transudative and inductive rating prediction tasks, experiments are conducted on five recommendation benchmarks. The results show substantial improvements in predicting ratings in the cold start scenario. For the future, the study proposes improving the model to enable it to handle heterogeneous graphs with varied node types and incorporate ranking algorithms.

In (Mooney & Roy, 2000), a content-based book recommender uses machine learning to classify and recommend books with similar textual content. The system must first extract in- formation from book titles from an Amazon search. Information extracted includes book title, authors, content descriptions, related books, etc. The textual content in each slot is separated into word tokens and stored in a "bag" in that particular category. The user allows the system to learn their preferences by providing ratings of a chosen list of books. The Bayesian text classifier returns the probability of the book being positively received by a user. The system is capable of the significant categories, or reason recommendation, for each item recommended. Several measures, precision, recall, accuracy, and f-measure, were used to evaluate the system. For many of the tested datasets, a precision of over 80% is present in the top 3 and top 10 recommendations, even with a smaller number of training samples. The study must be tested against other similar systems to compare its efficiency better and must also be tested on larger datasets to test its efficacy in real situations.

5.6 Linear Methods for Data Sparsity, as presented in the research work (Ning & Karypis, 2011), is a study that focuses on producing fast and quality recommendations over sparse matrices using the newly introduced SLIM (sparse linear methods) algorithm. A coefficient matrix is learned for each item with regularisation and optimisation. The final recommendation score for each user is generated as a sparse aggregation of rated items. The user-item matrix is combined with the coefficient matrix. The SLIM algorithm is also extended into another algorithm, including feature selection with two sets—the first consists of four datasets of customer card transactions and the second with four datasets of multi-valued ratings. Upon comparing the hit ratio (HR) and average reciprocal hit-rank (ARHR) of the SLIM methods with other existing methods, the HR and ARHR on SLIM methods surpass other methods in almost all of the datasets used. The feature selection SLIM algorithm, fsSLIM, solve the speed issue of SLIM while providing equivalent performance as SLIM.

(Lin, Pu, Li, & Lian, 2018) Course recommender systems for technologically-based smart education have an increasing responsibility to recommend courses that align with the learner's curriculum, abilities, goals, and interests. The study utilises a modified approach to the SLIM (sparse linear method) for smart education. A student-course matrix is built in this approach, where a student-course pair value is marked one if fulfilled and 0 otherwise. A sparse matrix of aggregation coefficients that are iteratively determined is used to determine the score of courses not taken. The first N courses with the highest score are recommended to the student. Results evaluated using hit rate and average reciprocal hit-rank show that compared to other standardly used methodologies, such as K Nearest Neighbour (KNN), P-SVD, and regular SLIM, the proposed methodology has a greater hit rate and average reciprocal hit-rank. It is noted that the method works more accurately as the number of course recommendations needed increases.

6. Domains of Recommenders

Recommender systems are both domain and company-specific. Users are provided with different recommendations even between different companies in the same domain. Oftentimes, the recommender used by the company affects the purchases made by a customer. From the customer's point of view, a helpful recommender offers a balance between what the customer recognises and what the customer is likely to enjoy being introduced. From pursuing entertainment in movie streaming platforms and music purchasing businesses, customers prefer companies that employ reliable recommenders to purchasing necessities.

An issue in recommender systems that affect recommendations is item cold-start. Vie et al. (Vie, 2017) aim to alleviate this issue in anime and manga recommendations by using a deep learning technique to extract tag information from anime and manga posters, including a lot of character information. They propose a new model for collaborative filtering and a convolutional neural network for extracting semantic information from images. The proposed model relies on the rating matrix as far as possible, and in cases when rating information hardly exists, it relies on the posters. The model is composed of several blocks. The first block is a convolutional neural network, and it takes a poster as input and gives tag predictions as the output. The second block uses alternate least squares with λ-weighted regularisation to perform a matrix factorisation. The third block uses tag predictions to perform a regularised linear regression of every row of the rating matrix to deduce explainable user preferences. There is also a Steins gate, which blends the outputs of the Alternate Least Square (block two) and Least Absolute Shrinkage and Selection Operator (block three) models to provide a final rating value by overcoming their limitations. After conducting experiments on datasets, it is found that the proposed model shows substantial improvement in the recommendation of little-known items but provides comparable performance to ALS. The model performs better than most models in the item cold-start scenario and provides better rating predictions than pre-existing techniques, and gives an accountable understanding of the users' taste.

6.1 Movie Domain. K. Chapphannarungsri and S. Maneeroj (Chapphannarungsri & Maneeroj, 2009) propose an advanced recommender system to offer better quality recommendations by merging two types of ratings; multiple criteria rating and multidimensional approaches. The proposed recommender system focuses on providing recommendations of a higher quality. The prototype uses Multiple Linear Regression analysis in place of a Reduction-Based approach. It has three steps: 1) obtaining the users' opinion, 2) Finding Neighbour Process and finally 3) Generating the recommendations. The authors then experiment and evaluate the data. The objective of doing so is to show that the proposed prototype provides more accurate recommendations than the hybrid systems used currently, which are based only on multiple criteria ratings. In the experimental evaluation result, the proposed system is a success and provides more accurate recommendations than the Thai-Music system. The approach proposed in this pa- per provides a higher quality of recommendations because the method weights all components with their weight value instead of weighting only the biggest value of the feature. Furthermore, it also increases the weight of multi-criteria rating by using the frequency of the selected movie features. The system can overcome the unsuitable weight problem and incorporate contextual information without losing a huge amount of data.

Kumar et al. (Kumar, 2015) propose a movie recommendation system based on collaborative filtering, which obtains and analyses the information provided by the users and finally recommends the best-suited movie for that user. The K-means algorithm is used to generate the recommended move list, which is sorted based on other users' ratings given to the movie. The proposed model pre-filters before applying the K Means Algorithm and uses five attributes to calculate the distance of each point from the centroid, namely, genre, actor, director, year, and rating. They have prioritised the rating attribute over other attributes. Their research discovered that the most fitting recommendations that can be generated are to be based on the ratings given by the previous users. The authors faced a couple of challenges and arrived at solutions. These include: Giving weights to different attributes, solved by conducting a survey of people and prioritising the attributes based on the survey results. They make the system diverse to satisfy users in various geographical locations, solved by including movies irrespective of language and location. And to have the most fitting movie recommendation list, solved by applying the K Means clustering algorithm along with a pre-filter.

In (Cui, 2017), the author aims to create a prototype of a movie recommendation system to enable users to obtain a movie they will be interested in from the database. The KNN collaborative filtering algorithm is used, which is a combination of the KNN algorithm and collaborative filtering algorithm. The KNN algorithm is used to select neighbours, and the algorithm has a few basic steps. The steps are: user similarity calculation, K nearest neighbour selection and predict score calculation. The user similarity is computed by assessing the value of the items evaluated by two different users. This is done by using the methods of Cosine Similarity and Pearson Correlation Similarity. The predicted score calculation is done according to the score of the item's neighbour. The author also creates a personalised recommendation system design which comprises an architecture design and database design. The architecture design system uses JavaEE architecture and is based on B/S mode. For the database, MYSQL is used. Through experiments and tests, the system is found to be highly efficient (Cui, 2017).

S.-M. Choi et al. (Choi, Ko, & Han, 2012) aim to improve and refine a previously proposed movie recommender system based on genre correlations (Choi & Han, 2010) and improve the genre correlation algorithm. They aim to solve the problems of sparsity and cold-start, associated with the collaborative filtering approach. They analyse the characteristics of genres to improve genre correlations. They consider genre correlations according to the number of genres and the decade in which the movie was released. By considering genre correlations according to the number of genres, changes in genre correlations where movie data is limited can be revealed. And by considering genre correlations according to the decade, changes in genre correlation according to periods can be indicated. As a result, accurate genre correlations can be formed based on a limited amount of movie data. Precise ratings for users who prefer movies from certain decades can also be provided. After conducting the experiments and analysing genre correlation using specific data, the results indicate that dependable genre correlations can be formed using the improved algorithms. Another inference was that using decade-based genre correlations results in more accurate recommendations.

(B. Li et al., 2014) Systems that make use of user-profiles, their social connections, and past usage history are prone to make inaccurate recommendations in some commonly occurring scenarios. The main issues arise from users that refrain from filling out their profile as well as from "similar" users that may be socially connected to a user but may not necessarily have matching movie preferences. Hence, the authors propose the use of clustering techniques to analyse the 1. Movie features, and 2. Users' voting logs to make correct and accurate recommendations. Using clustering techniques, they classify movies into various different areas and compute a distance matrix for the features. For each user, the movies are recommended based on a combination of the clusters and the users' majority voting result. Clustering techniques like density-based clustering, affinity propagation, hierarchical clustering, and random clustering as a baseline are implemented to compare efficiency and accuracy. They proposed a recommendation system with three parts: 1. Distance computation system, 2. Pre-clustering system and 3. Real-time online recommendation system. The distance computation system computes the distance between different movies based on the different properties of the movies. The movies are separated into different clusters by the pre-clustering system before being given to the online recommendation system. The authors tried five different kinds of clustering algorithms. Upon obtaining the pre-clustering results, the website's history of users' votes on movies was read. The majority of the voting was used to generate recommendations. After the evaluation and experiment, the proposed model obtained 84.71% accuracy as the best result. This pre-clustering approach allows for less storage to be used and less time needed to generate an output. The system does not compromise accuracy by performing constant, real-time updations.

In section 9 of (Santos, 2019), the authors propose a model for a movie recommender system inspired by the LIDA model (Franklin, 2016). The proposed model consists of nine modules that have short- and long-term memories. The work cycle of the model is that it recognises input data, processes the movie data, mounts the processed information based on attention concepts that are to be resent to the external environment. There are twelve linear steps in the model, as follows. 1. Stimulus Identification, where the user ID is identified. 2. User ID Sending, here the user number identification is captured and sent. 3. User Data Request, where the user ID is requested for and then sent to the declarative memory module, which stores all movies related to users. 4. User Data Sending, where the corresponding user data is retrieved after searching for the user ID. 5. User Data Routing, 6. Similar Users Request 7. Similar Users Sending 8. Histogram Generation, here a histogram is generated of all the movies that similar users watched. Every genre is split and assigned a binary value; 1 if the genre is present and 0 if not. Then the K Means algorithm is applied to identify which cluster each movie belongs. 9. Cluster Identification 10. Movie Competition 11. Movie Preparation 12. Recommendation Commit. Several experiments and comparisons are conducted with a traditional model. The first is with clusters and similar user numbers. This resulted in the proposed model performing better when K=8, regardless of similar user numbers. This is because it demonstrates a greater precision index. Then the proposed model was evaluated by users, where each volunteer rated every movie they had watched. The results were positive, and the volunteers showed a high satisfaction rate with the movies recommended to them. The average precision among the volunteers who evaluated the model's recommendations based on their watched movies was 32.7%. The proposed cognitive model is closer to cognitive behaviour than a traditional model and could become an alternative to understand neural models better and to simulate said, models.

In (Ahuja et al., 2019), Ahuja et al. propose a movie recommender system using K-Means Clustering and K-Nearest Neighbour algorithms. The users' preference for a movie is predicted on the basis of a number of parameters, and the model's underlying concept is that people can have a common preference. The architecture of the system has three modules: the input module, the processing module and the output module. The input module takes in the personal and profile details of the user. In the processing module, the data is

TABLE 1 – Summary of the reviewed movie recommendation systems and their methods.

Ref. No.	Method	Observations
(Chapphannarungsri & Maneeroj, 2009)	Multiple criteria rating and multidimensional approaches are merged, and Multiple Linear Regression analysis (in place of a Reduction Based approach) is used. This proves beneficial over systems that are based only on multiple criteria ratings.	The system weights all components with their weight value, thus providing higher quality recommendations than those provided when only the biggest value of the feature is weighted.
(Basu et al., 1998)	The system is based on collaborative filtering and applies a pre-filter before using the K-Means algorithm. Five at- tributes are used to calculate the distance of each point from the centroid.	The 'rating' attribute is prioritised over other attributes, and it is observed that doing so generates the most fitting recommendations.

(Cui, 2017)	The KNN collaborative filtering algorithm is used, a combination of the K Nearest Neighbour algorithm and collaborative filtering algorithm. Cosine Similarity and Pearson Correlation Similarity methods are used to compute the user similarity.	The system is proven to be highly efficient and stable, with good recommendations.
(Choi et al., 2012)	Based on genre correlations, the model aims to solve the problems of sparsity and cold-start associated with collaborative filtering. This is done by improving genre correlations.	Accurate genre correlations are formed, which results in precise recommendations for users. Using decade-based genre correlations is shown to have more accuracy.
(Q.Li et al., 2007)	Clustering techniques like density-based clustering, affinity propagation, hierarchical clustering, and random clustering as a baseline are implemented. A distance matrix is computed for the features.	The result is a real-time updating model that does not compromise accuracy. The model obtained 84.71% ac- curacy as the best result.
(Santos, 2019)	The proposed cognitive model has nine modules and works in twelve linear steps. It uses the K Means algorithm to identify which cluster every movie be- longs to.	The model was efficient and got satisfactory reviews from volunteers who tested it. The average precision rating among volunteers was 32.7%.
(Ma &Hao, 2011)	The model uses K-Means Clustering and KNN algorithms, and utility and similarity matrices are built.	The model produces the same value of RMSE as existing systems but with a lesser number of clusters.
(Shi et al., 2010)	The system recommends movies based on 'mood' by using joint matrix factorisation and conventional collaborative filtering techniques. User-item and mood similarity matrices are created, which are then combined into a movie-mood matrix	The recommendations show significant performance improvement in comparison to regular matrix factorisation algorithms.

processed, and a utility matrix is built, KNN and M-Means algorithms are used, and a similarity matrix is built. In the output module, the recommendations are shown. It is observed that the proposed system achieves the same value of root mean square error as the existing systems but with a lesser number of clusters. The data is obtained from the Movie Lens dataset, and the algorithm is implemented in python. For future work, the authors propose using sentimental analysis to increase the efficiency of the system and accommodate more situations.

(Shi et al., 2010) explores the usage of context-aware information in movie recommendation systems is explored with respect to movie mood. The main purpose of this proposed recommender is to incorporate surrounding, contextual data into conventional collaborative filtering techniques while also being able to consider the "mood" or specific event for which the system is being used. The "mood" may be indicative of a specific event, holiday, genre, style, etc., and only movies with similarity to a specified mood must be recommended. The system is achieved using joint matrix factorisation, which goes beyond the user-item matrix of regular matrix factorisation by integrating any additional contextual information, such as user occupation, social media relationships among users, and movie mood tags. Specifically, the user-item matrix based on ratings is jointly factorised alongside a mood similarity matrix. The mood similarity matrix is applied with respect to a defined mood and is then combined into a movie- mood matrix. The final matrix contains similarities between movies based on both ratings and mood. Results produce recommendations of movies with similar mood tags by gathering movies with the highest values in the final matrix. Compared against other common algorithms using precision evaluation metrics, the new algorithm shows considerable performance improvement, especially against a regular matrix factorisation algorithm.

6.2 Music Domain. As reported by TechCrunch, based on the Recording Industry Association of America data, music streaming applications have been gaining popularity and traction, accounting for 79.5% of all recorded music revenues in 2019. A staple feature of these applications is music recommendations, which introduces users to new music.

In (Wu, 2019), a personalised music recommendation system is proposed using a hybrid recommendation algorithm based on collaborative filtering and music genes. The proposed model consists of three parts: 1. User preference model, 2. Music resource description, which involves the organisation and management of music resources, and 3. The recommendation algorithm. The two main technologies used are collaborative filtering, in which users are divided into groups on the basis of their interests, and music genes, which are the characteristics of the music and are divided into internal genes (lyrics, melody, rhythm, speed) and social genes (composer, singer, name, styles). The system's architecture comprises three layers: the presentation layer, business logic layer, and data access layer. Every user's preference for each feature is analysed to recommend music accurately. The system calculates a recommendation list through the collaborative filtering algorithm by first obtaining 'the self-rating data generated by user operation or automatically filing data based on user preferences from the database'. Another recommendation list is created based on the gene information of the users' favourite songs. The two lists are used to provide the final recommendations. The proposed system is tested using a dataset from the Pandora Rado network. The experiments show that the model has the smallest Mean Absolute Error and Root Mean Square Error (Wu, 2019).

A. Vall et al. (Vall et al., 2019) propose two feature-combination hybrid recommender systems for music playlists. On its own, collaborative filtering applied to collections of curated mu- sic playlists is accurate, but the music collections display a long-tailed distribution. The authors introduce a 'two feature-combination hybrid recommender systems that extend collaborative filtering by integrating the collaborative information encoded in curated music playlists with any type of song feature vector representation.' The first system is a profiles-based playlist continuation, and the second is a membership-based playlist continuation. In profiles-based playlist continuation, the system is specialised to a playlist collection, using a song-to-playlist classifier and can deal with out-of-set songs, but can't extend out-of-set playlists. The membership- based playlist continuation system checks whether a particularly given song and a given playlist fit together. Specialisation towards specific playlists is discouraged, and the system can deal with both out-of-set songs and out-of-set playlists. Offline evaluation is conducted to evaluate the quality of the proposed model and to compare it with pre-existing and widely used pure and hybrid collaborative systems matrix-factorisation-based playlist continuation and hybrid matrix-factorisation-based playlist continuation. Through the results obtained, it is concluded that profiles-based playlist continuation and membership-based playlist continuation outer- form matrix-factorisation-based playlist continuation for sparse and out-of-set songs but are at a similar level when there is an availability of sufficient training data. The performance of the proposed systems can be improved by combining different song feature vectors due to their flexibility of being able to exploit any type of song feature vector.

Hadash et al. (Hadash, Liang, & Willemsen, 2019) present a study based on music recommender systems to examine how individual track evaluation compares to playlist evaluation. They further examine details like how playlist attractiveness relates to individual track rating and other subjective and objective measures. The study uses three algorithms to generate playlists: 1. Track and artist similarity algorithm 2. Genre similarity algorithm 3. Audio feature algorithm. In the track and artist similarity algorithm, a common algorithm is applied to a list of tracks and artists a user likes. The genre similarity algorithm utilises an ordered list of genres that a particular user likes and a similarity metric to produce genre likeability scores for various other genres. The audio feature algorithm clusters track with similar audio features. As a result, the first two algorithms generate recommendations that are close to the users' known music taste and liking. The third algorithm recommends music and tracks that have similar audio features, which leads to the recommendation of genres, artists, and tracks that the user may not be familiar. The relation between track and playlist evaluations is evaluated based on three conditions in a randomised order. The conditions and track and artist algorithm, track and artist algorithm combined with a genre similarity algorithm, and track and artist algorithm combined with the audio feature algorithm. The results show that there is no impact of personalisation rating on perceived attractiveness. However, likeability rating can in part predict perceived attractiveness. Moreover, playlist attractiveness is more strongly related to the recommendation algorithm. Playlists in the track and artist algorithm combined with the audio feature algorithm condition are more negatively evaluated compared to playlists in the other conditions. This is despite the track evaluations being similar on average. The results demonstrate that playlist attractiveness is not mainly related to the likeability of its tracks and that other factors (for in- stance, diversity) can play a role too. For future work, the authors aim to address why certain algorithms generate more attractive playlists compared to others, and they propose to include a subjective measure for track familiarity (Hadash et al., 2019).

Q. Li et al. (Q. Li et al., 2007) detail a model for a collaborative music recommender system, which is based on the authors' proposed item-based probabilistic model (Li & Kim, 2005; Li, Kim, Kim, & Kim, 2004). In the item-based probabilistic model, items are categorised into groups and predictions are made for users taking into account the Gaussian distribution of user ratings. The model uses audio features which help ease three common problems associated with data sparseness in collaborative recommender systems: user bias, non-association, and cold start problems. Both user rating data and content information of items are considered for parameter estimation to alleviate the problems caused by the sparseness of data. A simple k- medoids clustering method is used, by which each cluster is represented by one of the objects in the cluster. In the k-means algorithm, however, the centroid of all the member objects represents the cluster. As a result, the k-medoids algorithm is less influenced by outliers as compared to the k-means algorithm. After experimental evaluation, it is observed that the proposed model outperforms the other two standard models. In addition, even though the model was originally developed for music recommendations, it can be applied to movies and other domains (Q. Li et al., 2007).

TABLE 2 - Summary of the reviewed music recommendation systems and their methods.

Ref. No.	**Method**	**Observations**
(Wu, 2019)	User preference-based song genre and music source-based filtering. Grouping based on generic interest was the key to this system and rating help to archive results.	User and groups rate help to achieve results with smaller mean absolute error in the system
(Vall et al., 2019)	Collaborative filtering hybrid method was experimented. In this method feature are combined to obtain collaborative filtering.	This approach gave more flexibility and opportunity to combine feature and train the system.
(Hadash et al., 2019)	Similarity indexing was the key method used in this system the authors considered track, genre and audio features	The work was able to demonstrate the playlist and the track in the playlist are independent attributes.
(Kazai et al., 2016)	The system was model using item based probabilistic model in which item are categorised and predication are based on the distribution.	This model outperformed standard models and an- other interesting feature was this model was able to accommodate domain out music.

6.3. E-commerce Domain**.** There has been a rapid growth of e-commerce in the past two decades, most notably in online shopping sites like Amazon, Alibaba, and eBay. The platforms need to learn from the habits of users individually to make customised recommendations to users. The idea of recommendation in online shopping links one product to other products and one user to other users, as clusters, and then creates connections between them (Hung, Duy, & Huynh, 2019). E-commerce recommenders must be able to keep up with the growing customer base and a large selection of products. One of the most important factors is scalability- the system to provide significantly more recommendations in the same time period (Sarwar, 2002).

L. Jiang et al. (Jiang et al., 2019) proposed a slope one algorithm for e-commerce based on the combination of user similarity and trusted data to increase the accuracy of the predictions when dealing with personalised recommendation tasks. To solve the problems faced by slope one algorithms, such as untrustworthy ratings by fraudulent users and low prediction accuracy, the authors present a slope one algorithm. The proposed algorithm has three steps. The first is the selection of trusted data. Then the similarity of users should be calculated. Lastly, this similarity is added to the weight of the improved slope one algorithm to obtain the final recommendation expression. The ratings by users in the dataset are classified as useful or not. If the scores a rating as 'YES', it is considered to be trusted data, else it isn't. It was then observed that the recommended results were fairly accurate and good when the trusted ratio is greater than

0.8. The authors then present three slopes one algorithms: the basic slope one algorithm, the weighted slope one algorithm, and the bi-polar slope one algorithm. Put simply, the basic idea of the algorithm is to 'use the average instead of the rating difference between two different individuals.' The evaluation is done in four steps. First, the slope one algorithm is compared with the traditional algorithm, based on trusted data. Then the difference between the slope one algorithm based on the combination of trusted data and similarity and the algorithm based on trusted data using mean absolute error is researched. After comparison using mean absolute error, the comparison is made using root mean square error. Finally, there is a comparison be- tween the proposed algorithm based on user similarity under different dataset sizes. Based on experimental results, it is determined that the proposed slope one algorithm has a significantly improved prediction accuracy as compared to the traditional slope one algorithm.

The authors of (Asghar et al., 2020; Gadekallu, Soni, Sarkar, & Kuruva, 2019; Gu, 2020; Hakak et al., 2021; Maddikunta, Gadekallu, Al-Ahmari, & Abidi, 2020; Rm & Priya, 2020) pro- pose Hierarchical User Profiling (HUP) framework is used to resolve the problem of hierarchical user profiling in e-commerce systems. HUP collects the users' micro-behaviours for modelling like clicks on pictures or comments of an item, searches with navigation of the search engines or recommendations. This differs from conventional methods where only item-level behaviours like ratings and clicks are utilised. The paper also proposes Behaviour-LSTM. A behaviour gate is used to model the types and dwell time of behaviours to formulate the real-time interest of users effectively'. The proposed HUP method is evaluated on the basis of item recommendations and category recommendations. The 'JD Micro Behaviours Dataset' is used to evaluate its effectiveness, which contains users' micro-behaviour in the 'Computers' and 'Appliances' product categories. The results obtained demonstrate that the proposed model substantially outperforms state-of-the-art methods. For item recommendation, it has a 3.4% improvement. For category recommendation, the performance gains are comparatively subtle. The method is proven to be effective for both item and category recommendation tasks. As a result, the framework can be widely used to recommend items and their corresponding categories.

In (Guia et al., 2019) present a hybrid recommender system that combines the simplicity of the KNN collaborative filtering algorithm with the efficiency of ontology-based recommenders. They create user profiles and the users' neighbours' profiles because to be able to recommend products to a user. It is necessary to find other users that have bought at least one common product. The KNN algorithm is applied to find the user's nearest neighbour. Doing so enables the extraction of products and items that the user has not yet purchased. Then, the KNN algorithm is applied to find the nearest products. The metrics for finding the nearest product include the overall rating given by the neighbour, the price of the product, and text reviews of each product. They use a dataset containing 82.63 million reviews from Amazon between May 1996 and July 2014, divided into two parts. The first part contains attributes, and the second part contains metadata. They create and use an ontology 'to implement a data model that can describe and represent all the concepts that concern the active user, his neighbours and the products purchased by the latter, which are the ones that can be recommended to the active user'. Four main classes are created to develop the ontology-based recommender system: person, user, neighbour, and product. 'Person' is a class that represents the persons that have reviewed the product. 'User' is a subclass of 'person' and represents the user who will be recommended new products. 'Neighbour' is a subclass of 'person' and represents a reviewer that has purchased at least one product in common with the user and other different products as well. Lastly, 'product' is a class that represents a product purchased by a 'neighbour'. While conducting experiments, the authors compare the proposed hybrid ontology-based recommendation system with a collaborative filtering algorithm. The results show that the proposed system displays improved product recommendations. It increases the number of products that are recommended to the user, even if they belong in unknown categories. The recommended products also better match the user's preferences compared to the collaborative filtering recommendation system. A drawback of the proposed model is that it takes more time to apply the KNN algorithm to find the nearest products, while the time taken for the same is very less in the collaborative filtering recommender system. The authors intend to use a larger dataset for future work and improve the model by adding extra features.

TABLE 3 - Summary of the reviewed music recommendation systems and their methods.

Ref. No.	**Method**	**Observations**
(Jiang et al., 2019)	This method banked on the used similarity index dataset of trusted user pattern was used to train the system.	The system was able to achieve fair result compared to the conventional methods.
(Gu, 2020)	This utilized hierarchical user profile and trained more on the user behaviors in the system.	Achieved strong category recommendation and should good result inside category choices.
(Hsu et al., 2010)	Hybrid KNN based with collaborative filtering with ontology support. Training was on per- formed attribute and metadata. The methods was able to support millions of attributes training data.	The system was able to provide superiors recommendation and the value added distinction was they system was able to recommend products in different new categories.

7. Conclusion

Recommendations comprise a considerable proportion of the purchases a user makes. This influence of the recommender system varies depending on its purpose and effectiveness. In this survey, various research papers about recommender systems were analysed in order to survey the various existing recommender systems and the purposes for which they were created.

Systems must be designed by considering relevant and necessary attributes. Such attributes include accuracy, diversity, stability, trust, scalability, transparency, serendipity, and profitability. We have noticed that among the papers surveyed, recommender systems of the entertainment domain find greater success by using methods that more provide diverse, niche, and serendipitous recommendations. Generally, by offering a very diverse or niche recommendation suggestion, there is a greater benefit if it appeals to the consumer than there is a risk that they will be dissatisfied with the inaccuracy of the suggestion. A balance between diversity and accuracy is necessary. One method of doing this can be using a hybrid system of two components, one that yields higher accuracy and another that yields higher diversity. For example, (Zhou, 2010) uses a heat spreading component for diversity and a probability spreading component for accuracy.

On the contrary, ecommerce domains have a sales objective and consequently prefer systems with a considerable amount of accuracy and stability in order to ensure that they are properly matching the user's preferences. Stability is important in order to offer proper suggestions despite background noise. In most real-world systems, transparency must be maintained to increase the user's understanding of the recommendation. Both scalability and profitability are desired characteristics throughout all systems. However, the degree to which they are desired varies depending on the size of the user platform and the company's profit requirements.

This paper reviewed several common methods of implementation for recommenders. Often times, social information pertaining to each user is collected from external sources and integrated with content or collaborative filtering in order to improve recommendation relevance. Clustering methods are implemented when the system wishes to group items or users together based on the distances between them, either to generate the final list of closest item recommendations or to create neighbourhoods of similar and trustworthy user's classification algorithms for the system to learn whether or not a user will like a particular item. Regression algorithms can be used to predict a numerical value of the user's enjoyment. Sparse linear methods specifically combat the data sparsity problem found in many collaborative filtering systems. Many systems using machine and deep learning techniques were found to hybridise the systems by consolidating the best recommendations of various methodologies into a superior selection of recommendations.

Furthermore, we analysed recommendation systems used in three domains movies, music and ecommerce. The recommendation systems were designed to alleviate problems like the cold start scenario, data sparsity, and lack of accuracy and efficiency. K means algorithm, KNN algorithm, collaborative filtering and clustering techniques were found to be frequently used.

As such, the different kinds of recommenders generate varying results, and these results can only be considered accurate and proper recommendations based on their popularity with users. Well-developed recommender systems have the ability to attract and maintain the interest of users consistently. In this survey, we have combined differing recommender systems to illustrate their functions, uses, and appropriate implementations in differing circumstances.

References

[1]. Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie recommender system using KMeans clustering and KNearest Neighbor. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 263268). IEEE.

[2]. Asghar, M. Z., Subhan, F., Ahmad, H., Khan, W. Z., Hakak, S., Gadekallu, T. R., & Alazab, M. (2021). Senti-eSystem: A sentiment-based eSystem-using hybridized fuzzy and deep neural network for measuring customer satisfaction. Software: Practice and Experience

[3]. Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., & Netanely, I. (2013, October). Movie recommender system for profit maximization. In Proceedings of the 7th ACM conference on Recommender systems (pp. 121128).

[4]. Bahramian, Z., & Abbaspour, R. A. (2015). AN ONTOLOGYBASED TOURISM RECOMMENDER SYSTEM BASED ON SPREADING ACTIVATION MODEL. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 40.

[5]. Basu, C. (1998). Recommendation as Classification: Using Social and ContentBased Information in Recommendation Chumki Basu.

[6]. BlancoFernandez, Y., PazosArias, J. J., GilSolla, A., RamosCabrer, M., & LopezNores, M. (2008). Providing entertainment by contentbased filtering and semantic reasoning in intelligent recommender systems. IEEE Transactions on Consumer Electronics, 54(2), 727735.

[7]. Chapphannarungsri, K., & Maneeroj, S. (2009). Combining multiple criteria and multidimension for movie recommender system. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1).

[8]. Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Shah, H. (2016, September). Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems (pp. 710).

[9]. Choi, S. M., Ko, S. K., & Han, Y. S. (2012). A movie recommendation algorithm based on genre correlations. Expert Systems with Applications, 39(9), 80798085.

[10]. Cui, B. B. (2017). Design and implementation of movie recommendation system based on Knn collaborative filtering algorithm. In ITM web of conferences (Vol. 12, p. 04008). EDP Sciences.

[11]. Farokhi, N., Vahid, M., Nilashi, M., & Ibrahim, O. (2016). A multicriteria recommender system for tourism using fuzzy approach. Journal of Soft Computing and Decision Support Systems, 3(4), 1929.

[12]. Franklin, S., Madl, T., Strain, S., Faghihi, U., Dong, D., Kugele, S., ... & Chen, S. (2016). A LIDA cognitive model tutorial. Biologically Inspired Cognitive Architectures, 16, 105130.

[13]. Gadekallu, T., Soni, A., Sarkar, D., & Kuruva, L. (2019). Application of sentiment analysis in movie reviews. In Sentiment Analysis and Knowledge Discovery in Contemporary Business (pp. 7790). IGI Global.

[14]. Gu, Y., Ding, Z., Wang, S., & Yin, D. (2020, January). Hierarchical User Profiling for Ecommerce Recommender Systems. In Proceedings of the 13th International Conference on Web Search and Data Mining (pp. 223231).

[15]. Guia, M., Silva, R. R., & Bernardino, J. (2019). A hybrid ontologybased recommendation system in ecommerce. Algorithms, 12(11), 239.

[16]. Hadash, S., Liang, Y., & Willemsen, M. C. (2019, January). How playlist evaluation compares to track evaluations in music recommender systems. In 6th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2019 (pp. 19). CEURWS. org.

[17]. Hakak, S., Alazab, M., Khan, S., Gadekallu, T. R., Maddikunta, P. K. R., & Khan, W. Z. (2021). An ensemble machine learning approach through effective feature extraction to classify fake news. Future Generation Computer Systems, 117, 4758.

[18]. Hou, L., Liu, K., Liu, J., & Zhang, R. (2017). Solving the stability–accuracy–diversity dilemma of recommender systems. Physica A: Statistical Mechanics and its Applications, 468, 415424.

[19]. Hsu, C. K., Hwang, G. J., & Chang, C. K. (2010). Development of a reading material recommendation system based on a knowledge engineering approach. Computers & Education, 55(1), 7683.

[20]. Hung, P. D., & Le Huynh, D. (2019, February). ECommerce Recommendation System Using Mahout. In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (pp. 8690). IEEE.

[21]. Iaquinta, L., De Gemmis, M., Lops, P., Semeraro, G., Filannino, M., & Molino, P. (2008, September). Introducing serendipity in a contentbased recommender system. In 2008 Eighth International Conference on Hybrid Intelligent Systems (pp. 168173). IEEE.

[22]. Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H., & Wang, X. (2019). A trustbased collaborative filtering algorithm for Ecommerce recommendation system. Journal of Ambient Intelligence and Humanized Computing, 10(8), 30233034.

[23]. Kazai, G., Yusof, I., & Clarke, D. (2016, July). Personalised news and blog recommendations based on user location, Facebook and Twitter user profiling. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 11291132).

[24]. Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. (2015). A movie recommender system: Movrec. International Journal of Computer Applications, 124(3).

[25]. Li, B., Liao, Y., & Qin, Z. (2014). Precomputed clustering for movie recommendation system in real time. Journal of Applied Mathematics, 2014.

[26]. Li, Q., Myaeng, S. H., & Kim, B. M. (2007). A probabilistic music recommender considering user opinions and audio features. Information processing & management, 43(2), 473487.

[27]. Lin, J., Pu, H., Li, Y., & Lian, J. (2018). Intelligent recommendation system for course selection in smart education. Procedia Computer Science, 129, 449453.

[28]. Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011, February). Recommender systems with social regularization. In Proceedings of the fourth ACM international conference on Web search and data mining (pp. 287296).

[29]. Maddikunta, P. K. R., Gadekallu, T. R., AlAhmari, A., & Abidi, M. H. (2020). Location based business recommendation using spatial demand. Sustainability, 12(10), 4124.

[30]. Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5488.

[31]. Mooney, R. J., & Roy, L. (2000, June). Contentbased book recommending using learning for text categorization. In Proceedings of the fifth ACM conference on Digital libraries (pp. 195204).

[32]. Ning, X., & Karypis, G. (2011, December). Slim: Sparse linear methods for topn recommender systems. In 2011 IEEE 11th International Conference on Data Mining (pp. 497506). IEEE.

[33]. O'Donovan, J., & Smyth, B. (2005, January). Trust in recommender systems. In Proceedings of the 10th international conference on Intelligent user interfaces (pp. 167174).

[34]. Pu, P., Chen, L., & Hu, R. (2011, October). A usercentric evaluation framework for recommender systems. In Proceedings of the fifth ACM conference on Recommender systems (pp. 157164).

[35]. RM, S. P., Maddikunta, P. K. R., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCAGWO for intrusion detection in IoMT architecture. Computer Communications, 160, 139149.

[36]. Santos, M. B., Lima, A. M., Silva, L. A., Vargas, F. S., WachsLopes, G. A., & Rodrigues, P. S. (2019). MIRA: A Computational NeuroBased Cognitive Architecture Applied to Movie Recommender Systems. arXiv preprint arXiv:1902.09291.

[37]. Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002, December). Recommender systems for largescale ecommerce: Scalable neighborhood formation using clustering. In Proceedings of the fifth international conference on computer and information technology (Vol. 1, pp. 291324).

[38]. Sheth, S., Arora, N., Murphy, C., & Kaiser, G. (2010). weHelp: a reference architecture for social recommender systems. In IEEE/ACM International Conference on Automated Software Engineering workshops. IEEE/ACM International Conference on Automated Software Engineering (p. 461). NIH Public Access.

[39]. Shi, Y., Larson, M., & Hanjalic, A. (2010, September). Mining moodspecific movie similarity with matrix factorization for contextaware recommendation. In Proceedings of the workshop on contextaware movie recommendation (pp. 3440).

[40]. Sinha, R., & Swearingen, K. (2002, April). The role of transparency in recommender systems. In CHI'02 extended abstracts on Human factors in computing systems (pp. 830831).

[41]. Swearingen, K., & Sinha, R. (2002, June). Interaction design for recommender systems. In Designing Interactive Systems (Vol. 6, No. 12, pp. 312334).

[42]. Vall, A., Dorfer, M., EghbalZadeh, H., Schedl, M., Burjorjee, K., & Widmer, G. (2019). Featurecombination hybrid recommender systems for automated music playlist continuation. User Modeling and UserAdapted Interaction, 29(2), 527572.

[43]. Vie, J. J., Yger, F., Lahfa, R., Clement, B., Cocchi, K., Chalumeau, T., & Kashima, H. (2017, November). Using posters to recommend anime and mangas in a coldstart scenario. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 3, pp. 2126). IEEE.

[44]. Wang, H., Wang, N., & Yeung, D. Y. (2015, August). Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 12351244).

[45]. Wu, D. (2019, January). Music personalized recommendation system based on hybrid filtration. In 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS) (pp. 430433). IEEE.

[46]. Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016, August). Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 353362)..

[47]. Zhang, J., Shi, X., Zhao, S., & King, I. (2019). STARGCN: stacked and reconstructed graph convolutional networks for recommender systems. arXiv preprint arXiv:1905.13129.

[48]. Zhou, T., Kuscsik, Z., Liu, J. G., Medo, M., Wakeling, J. R., & Zhang, Y. C. (2010). Solving the apparent diversityaccuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences, 107(10), 45114515.