Design of Personalized Recommender System based on Hybrid Filtering and Fog Computing Architecture

Survey of Recent Personalized Recommender Systems in Ubiquitous and IoT environment

Noura Abdaoui
National School of Computer Sciences (ENSI)
University of Manouba, Tunisia
E-mail: noura.abdaoui@gmail.com

Ismahene Hadj Khalifa, Sami Faiz
Higher Institute of Multimedia Arts of Manouba (ISAMM)
University of Manouba, Tunisia
E-mail: ismaheneawatef.hadjkhalifa@isamm.uma.tn, sami.faiz@isamm.uma.tn

***Abstract*—Ubiquitous recommendation systems aim to provide users personalized recommendations of online products or services. Various recommendations techniques have been developed to fulfill the needs in different scenarios. This paper presents a survey of recent Personalized Recommender Systems in Ubiquitous and Internet of Things (IoT) environment, followed by an in-depth analysis of groundbreaking advances in recommendation systems based on big data. Furthermore, this paper discusses the issues faced in modern recommendation systems, such as sparsity, scalability, and diversity and illustrates how these challenges can be transformed into personalized recommender model that is described in the final Section. The novel recommender model aims to integrate hybrid filtering technique and fog architecture in order to generate contextual and personalized recommendations. The Fog computing architecture aims to solve the ubiquitous recommendations issues related to IoT challenges. As result, the given model is a multi-layer fog structure which is implemented in Smart shopping and used multi sources big data in order to propose personalized offers according to the users' profiles and analyze their feedbacks to improve their experiences.**

Keywords-Ubiquitous Recommender System; personalized recommendation; hybrid recommender approach; fog architecture; IoT.

I. INTRODUCTION

Ubiquitous recommender systems assist the mobile user by providing him with personalized recommendations of items or services that are in his device while context is the most important aspect [3] context-awareness as defined by [4] includes parameters, such as Location, Time, Date and Weather. Thereby in order to provide accurate recommendations, we have to let the system use the context and at the same time have our privacy respected. Nowadays, technologies have become ubiquitous and users tend to use smart devices to use the internet. These devices used connected sensors coming under IoT [5]. This is a new paradigm-shift from traditional interactions between mobile user and devices, which provides the ubiquitous computing environment.

This new paradigm paves the path for huge applications on the mobile user level to improve the quality of being or service, and on the decision-makers' level to afford an enduring raise in revenue. This has created enormous potential for ubiquitous recommender systems in many industries and domains to provide tailored recommendations for mobile users. Since, thus enhancing the patient's quality in smart health, the customer's shopping experience in smart marketing as well as the enterprise goal, and improving the smart traveling plans, etc. As we know, most of the IoT applications are connected with cloud computing. And this cloud gives the services as on-demand and scalable storage, along with processing facilities according to IoT application needs. According to [1] for real-time applications, 30 million clients are transferring data up to 25 000 records every second, which is not efficient for the cloud. Together with the growth of data quantity and the emergence of different kinds of dynamic end-user and access smart IoT devices, the information overload problem is becoming serious. To address these limitations, fog computing technology was first introduced by Cisco in 2012, which integrated edge devices and cloud resources [2].

The main objective of this paper is to describe and discuss the existing recommend approaches in different fields. Then, we will give a brief description of our novel recommender system that aims to integrate hybrid filtering approach and fog architecture in order to generate personalized and ubiquitous recommendation collected from dynamic and heterogeneous big data. The paper is structured as follows: Section II gives an introductory of the studies of ubiquitous computing , the recent state of the art recommendation systems in ubiquitous and Cloud computing with similar discussion of given issues. Subsequently, the integration of fog architecture in the proposed recommender system will be presented in Section III. Section IV describes the system's implementation and highlights the results. In Section V, we wrap up the paper with conclusions and horizons of work that would improve the suggested ubiquitous fog-based recommender system.

II. RELATED WORK

In this Section, we present the background and studies of ubiquitous computing. Then, different ubiquitous and fog recommender systems are shown in many sectors using different filtering techniques that exist in literature. Finally, we will discuss the different limitations to improve our novel recommender model in the future work.

 ISBN: 978-1-68558-077-3

A. Ubiquitous Computing

The term ubiquitous computing, was conceived by Mark Weiser in 1988 at Xerox PARC. It is an embedded computational technology in the form of a microprocessor in every object [3]. Ubiquitous computing can occur with any device, any time, any place and in any data format across the network technologies, such as RFID, Wi-Fi and Bluetooth. Moreover, in order for ubiquitous to be achieved, common objects that humans use daily are appeared and afford computational services without expecting from users to explicitly interact with them. Apart of ubiquitous research deals with Location-Based Information Systems. Such systems utilize the user's location as context to provide users with the ability to produce and access information that is related to a location. Thus, context is an information that can be used to characterize the situation of a user in his interaction. Consequently, Ubiquitous recommender systems facilitate users on location by providing them with personalized recommendations of items in the proximity via mobile devices.

A number of ubiquitous recommender systems (RS) challenges exist, ranging from technological, such as wireless technology limitations and storage limitations, to challenges related to context-awareness, tracking user intentions and privacy concerns [6]. Without forgetting the amount of information, a user has to access is often lost and left with a feeling of disappointment and frustration.

In the next subsections, we will present the several approaches used to make the RS.

B. Collaborative Recommender System

The Collaborative Filtering (CF) technique recommends items based on the similarity measures between users and items [7]. The system recommends those items that are preferred by similar category of users. The CF approach has been used for recommendations in the IoT. There are two main CF techniques: memory-based CF and model-based CF. In memory-based CF, user recommendations or predictions of ratings on future items are based on the users' rating behavior by using correlations between items or between users. Model-based CF is more scalable in cases where only the training set is used to make the pattern. It uses this pattern to recommend future ratings. Compared with memory-based CF, model-based CF is considered less accurate because of the large fraction among the item-user values in the training part of the dense dataset [10]. In [11], the authors proposed a unified CF model based on a probabilistic matrix factorization recommender system that exploits three kinds of relations in order to extract the latent factors among these relations: user–user, thing–user and thing–thing. The author in [12] exploited the CF method to design an IoT trust and reputation model that investigated trust and reputation among IoT nodes and Probabilistic Neural Networks (PNN). The Quality of Recommendation (QR) is defined as a score of trustworthiness. In [13], the CF approach was adapted to the weather and location data that were collected by the sensors to provide effective recommendations for the residents of that geographical region. It is the weather and location-aware recommendation system. A Ubiquitous Context Aware Recommender Systems for Ubiquitous Learning (UbiCARsUL) is proposed by [14]. The system enables students to scan QR Code tag attached to the corresponding plant in order to display related multimedia materials on the screen of mobile phones. This new paradigm uses Collaborative Filtering Approach in recommender systems, Clustering Algorithm for grouping students and Association Rules Algorithm in Data Mining for discovering interesting relations between variables. Smart devices are far connected in IoT to allow ubiquitous service access. This can produce heavy service redundant. Therefore, a recommender searching mechanism for trust-aware recommender system (TARS) is proposed to enhance CF in IoT environment by [17], named S_Searching: based on the scale-freeness of trust networks, selecting the global highest-degree nodes to construct a Skeleton, and looking for the recommenders via this Skeleton. Benefiting from the higher outdegrees of the nodes in the Skeleton, S_Searching can find the recommenders efficiently. S_Searching can find almost the same number of recommenders as that of conducting full search, which is much more than that of applying the classical searching system in the scale-free network, while the computational complexity and cost is much less. But, the research on the recommender searching mechanism of TARS is still always at the beginning stage. It is not easy to find the most reliable recommenders for the active users in the scale-free network.

However, Collaborative Filtering (CF) suffers from many problems, such as :

- Scalability RSIoT deals with a large amount of data that need computation power to conduct the recommendations, as well as fast response to online user requirements.
- Cold start problem particularly when a new user joins the system.
- Data sparsity may affect the accuracy of collaborative RS.

C. Contetnt-Based Recommender System

In Content-Based Filtering CBF [23] methods, the algorithm recommends items or similar to those items that were liked in past. It examines previously rated items and recommends best matching item. The CBF approach also has been used for recommendations in the IoT. SOMAR is a recommender system proposed by [20], which aims to suggest activities to user and the data used are based on Facebook and sensor data. In [21], the authors adapted CBF approach in medicine to recommend a suitable activity plan for the patient's data through their medical sensors that can be worn and a virtual nurse explains this. In [22], the authors adapted a CBF approach to building a recommendation module for their smart restaurant, which aims to provide dish recommendations based on the customers' tracking history.

A natural limitation of Content-Based Filtering is the need to have a generic and rich representation of the content of the items. Moreover, this type of system generally suffers from the problem of overspecialization; for example, when a user likes an event (e.g. ads of discount pricing) during shopping, it does not mean that he will want to see it again. However, using a CBF approach, the system will suggest him to come-back a second time to the same place with the same type of event (even if it is not organized). When he might be more interested in events, he did not discover on the last shopping.

D. Knowledge-based Recommender System

Knowledge-based approach suggests products based on inferences about user's needs and preferences. It is based on the identified relationship between a user's need and a possible recommendation [24]. Ontology is a formal method of representing knowledge that is central to building RSIoT. For example, the authors in [25] proposed a method for generating automatic rules and recommending the best rule. The user has the opportunity to add new rules for newly connected devices. Three ontology models were created: (i) Things, which provides all information for things; (ii) Context, which provides contextual information about people, environment and things and (iii) Functionality, which links the functionality of context and things. For smart health applications, ontology plays a significant role in building a recommender system. [26] Adapted a fuzzy ontology to build a system that monitors diabetes patients and recommends specific food and drugs.

However, such models still face major issues, which cannot treat decisions that have no rules allied to the system, and needs knowledge engineering in the building, and can be expensive.

E. Hybrid Recommender System

Hybrid recommender approach [27] is the one that combines multiple recommendation techniques together to produce the output with higher accuracy. A lot of research work has been done to peruse the healthy life style, by employing machine learning algorithms on past user activity data, heart rate data, and accelerometer data to identify the type of activity, and/or estimate caloric consumption. PROFIT [28] is a hybrid recommendation system that matches the user's profile to available activities according to his geo-location and time availability. It is a personalized fitness assistant framework that integrates activity data collected by the user's smart phone, their preferences and fitness goals, their availability and their social network. Then, it automatically generates fitness schedules and socially enhanced recommendations of new activities, as well as fitness buddies by using the collaborative filtering. FOBA [5] is a fog computing system aims to recommend products of a banking entity. The solution developed by a hybrid method of recommendation: Collaborative Filtering combined with Content-Based Filtering.

F. Recommendations with machine learning (ML)

ML algorithms can be divided into three (supervised, unsupervised and semi supervised), based on the nature of the data involved. Many studies of RS have investigated this approach. In [16], the authors designed the Optimal State-based Recommender System by exploiting ML algorithms as Distributed Kalman Filters, Distributed mini-batch SGD and Distributed Alternating Least Square based classifier and some ML platforms. In [18], a recommendation engine that provides personalized wearable technologies recommendations for proactive monitoring. The framework consists of three main models: (i) the classifier model; (ii) the optimization model, (iii) the Monitoring Framework. Rasch in [19] adapted unsupervised learning to build an RS for smart homes. The system learns user patterns and conducts recommendations based on user contexts. A decision tree [8] is used to build a system that provides lifecare recommendations. A personalized recommendation system for e-commerce based on big data analysis is improved by [9]. The system is divided into four levels, which are data layer, management layer, business layer and presentation layer, and each layer is closely related to big data. The results improve that the text matching algorithm used in the system can define the membership of goods with the search keywords input by customers. But When the method is applied to full goods search, due to the variety of goods, the proposed method is difficult to understand when selecting the representative features and defining membership, so it needs further study. With the increase of IoT connected devices, the amount of data has increased. Many research contributions have employed ontology-based semantic approaches to improve the access and integration of heterogeneous information from various sources in many areas. ProTrip, a health-centric tourism recommender system has been proposed by [29]. It based on hybrid filtering, which is capable of suggesting the food availability through considering climate attributes based on user's personal choice and nutritive value.

G. Discussion of recommender system's challenges

Nevertheless, despite the success of ubiquitous recommender systems in IoT research there is still room for more studies to resolve many constraints as:

- Scalability. Traditional algorithms will face scalability issues as the number of users and items increases. However, recommendation systems must respond to the user's needs immediately, regardless of the user's rating history and purchase situation, which requires high scalability [30].
- Sparsity. Many commercial recommendation systems use big data, and the user-item matrix used for filtering may be very large and sparse. Therefore, the performance of the recommendation process may be degraded due to the cold start problems caused by data sparsity [31].
- Diversity. Unfortunately, some traditional algorithms may accidentally do the opposite because they always recommend popular and highly-rated items that some specific users love. Therefore, new hybrid methods need to be developed to improve the performance of the recommendation systems [32].

III. INTEGRATION OF FOG COMPUTING ARCHITECTURE FOR IMPLEMENTING THE PROPOSED RECOMMENDER SYSTEM

Fog computing expands the traditional cloud by adding an intermediate layer between mobile users and the cloud, where the new layer involves fog servers directly used near mobile users and IoT devices. The integration of fog architecture in

 ISBN: 978-1-68558-077-3

the proposed recommender system with the influx of huge amounts of data referred to big data is a crucial axis of the proposed model. This solution aims to provide intelligent tools to target and recommend the personalized offers according to the mobile users' profile, and to track and analyze their feedbacks to improve the customers experiences and predict their demands. The proposed architecture consists of three main layers as presented in Figure 1, which are IoT layer , Fog layer and cloud layer.

- Things Layer: is the layer closest to the end mobile user and ubiquitous environment where data is generated. This layer contains the mobile user sensors that operate to feed the system with data.
- Fog Layer: contains a number of decentralized fog nodes in each given location. Fog nodes have the potential to reduce the amount of data transmitted to the cloud layer and minimize the request response time for ubiquitous recommendations. Moreover, the fog nodes are also connected with cloud data center by IP core network.
- Cloud data center Layer: is the top layer of the architecture consists of multiple high-performance servers, storage devices and network access to shared resources over the IoT network. Thus, Cloud performs the "heavy services" of data analysis and processing that fog cannot perform, such as big data processing.

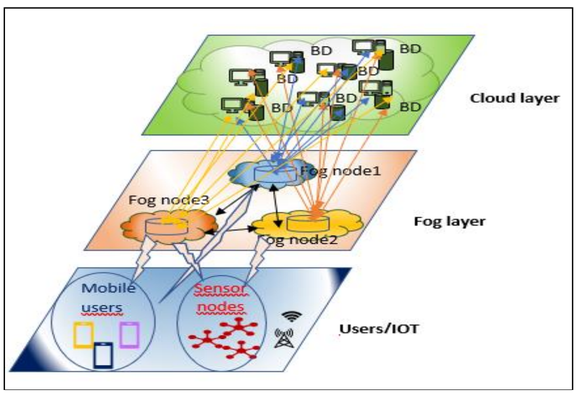

Figure 1. Fog architecture for implementing the recommender system

The principle of the designed model is that each mobile user should be connected with different fog nodes (fog server) in different floors by wireless access technologies Wi-Fi. Each fog server is linked into the cloud by IP core network. This architecture provides efficient data processing and storage services because each fog node represents RS that has a number of mobile users interfaces connected to both layers: IoT 's sensors, mobile users, other intelligent devices and the cloud layer. The RS provides personalized and contextual preferences recommendations to mobile user. Many resident modules are parts of the Context-A ware RS, including big data resources, new users, recommendations, list of personalized offers. Therefore, hybrid algorithm is implemented combined Content-Based Filtering and Collaborative Filtering to build the recommendations list. The goal is to rank most suitable content from contextual preference and user's profile. Then according required information and his interaction with fog nodes, we provide personalized recommendations.

As Figure 2 shows, there are four principal components in the proposed Personalized Recommender System Architecture namely; mobile user profile, Process of ubiquitous fog recommendations, mobile user interface and the fog data processing.

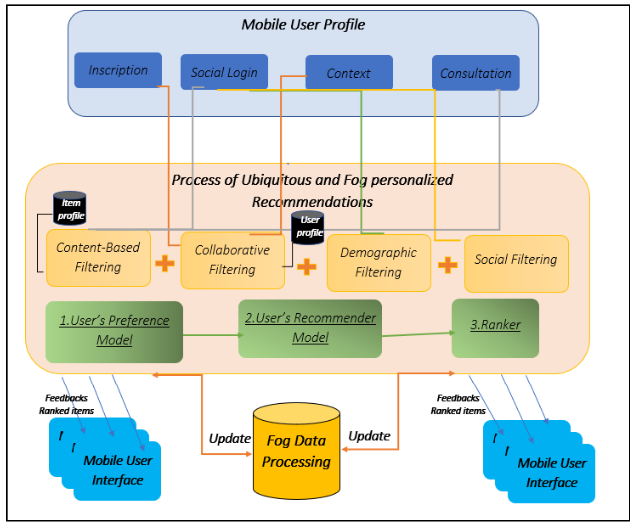

Figure 2. Personalized RS Architecture

A. *Mobile User Profile*

The user's profile can be extracted from many sources. Like inscription or by social login, even without log-in: such as frequency and duration of browsing, number of clicks, scrolling, etc. Also, through integrating contextual information (location, time, physical environment, ...). The data collected about the mobile user are then selected, analyzed, and saved as independent modules. These modules are combined to build the "user profile". A profile will contain information that can be used to determine mobile user's preferences in terms of items.

B. *Process of ubiquitous and fog personalized recommendations*

The process takes as input all the modules that constitute the target user's profile:

- The Content-Based Filtering describes the characteristics of center activities that the mobile user has consulted in the past in the form of key-word vectors. These key words are generally extracted automatically during the consultation or manually assigned during the inscription.
- The Collaborative Filtering contains the rating data of the consulted items and the user's context.
- Demographic filtering contains the user's demographic attributes. These attributes can either be entered by the user himself by filling in the registration form or extracted from his social login.

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-077-3

Once the mobile user profile modules are detected, recommendation approaches and the appropriate hybridization technique are selected, this process returns a list of items with the degrees of appreciation that the target user can give to each item. This process is detailed over several modules as shown below.

1) Update user's personal preference Model : it takes as input the historical data of a single user and outputs the user's contextual preference. Concerning the newly registered users for those the system has no historical data, a clustering block is designed using K-means *[*33] and Density-based spatial clustering of applications with noise (DBSCAN) [34] algorithms. These algorithms are used easily with any data type, various distance functions, and efficient indexing approaches facilitating the analysis of large datasets. According to the mobile user profiles of all the users, the system identifies clusters of users with similar profile information. Then, for each cluster, it collects the historical data of all the users belonging to that group. Next, the classification module used to learn a group-level contextual preference model. Having a clustering model and group-level contextual preference models for groups in hand, as soon as new user registers to the system. Finally, the group-level contextual preference model will be used as the user's personal preference model. The model will be in use until the system collects enough historical data from the new user for building a pure personal contextual preference model.

2) User's recommender Model : it receives as input the historical data of a single user or all users that belong to the cluster. To capture the user context, each historical data of user is made of the most recent user profile information, the offer (purchased or not). To create the training dataset, the system takes the received user data. Then, we use the data to find the best algorithm and best parameters and then save the final model. For each new offer, the system predicts the probability that the user will buy it and considers this value as the score of the offer; finally, offers are ranked according to their scores (i.e., the probability that they will be bought). To compute the prediction, and in particular to build the personal recommender models, we use popular classification algorithms is Bayes Search Cross-Validation (BSCV) [15], because it updates the current best model during each iteration and changes parameter settings according to the search ranges. Finally, we store the best model to be used in the future.

3) Ranker : for each personnalized offer, the system computes the offer's score (the probability of buyin the offer by user) using the user's recommender model. The Ranker receives the corresponding preference model of the user. Then, we obtain, for each offer, the probability that it will be bought by the user and use that as the offer score. Finally, the Ranker sorts the travel offers according to their scores and presents them to the user.

C. Mobile user interface

After the ranked list of offers is shown to the user in the mobile user interface, the users will typically buy an offer from the list shown to them and ignore the rest. The user's historical data are updated with the offers in the list (where each offer is tagged as purchased/not purchased). This interface consists of collecting the feedback of the purchasing decision after the ranked recommendation. This interface supports interactions between the mobile user and the Context-Aware personalized system of recommendation.

D. Fog Data processing

In this module, raw datasets are filtered converted and stored into various databases. The aim of data filtering is to extract useful information to the recommender system. We categorize six databases that describe a way of identifying the dataset: Item Profile, User profile, User-Item Preference, Fog Server List, Contextual Information, Recommendation Output.

- Item Profile: this database contains item attributes, such as item ID, description and category, and virtual content size.
- User's profile: this database contains user's attributes, such as age and gender.
- User-Item Preference: this database contains user-item preference information that has been converted from raw data. A preference could either be explicit or implicit.
- Fog Server List: fog server ID, workload capability, processing capability, storage capability and power usage.
- Contextual Information: this database contains contextual information such as user's localization, time, and network bandwidth information.
- Recommendation Output: this database contains the final output. It stores the output generated during the process of creating a recommendation, such as similarity matrix, training data and testing data.

The basic idea is to maximize the use of a fog server closer to the user. We try to process and store data as close as possible to the user who connects to the target fog server. If the target fog server cannot provide such a service, it will send a request to neighboring fog servers. If neighboring fog servers cannot provide the requested service, then the request will be sent to the cloud.

IV. IMPLEMENTATION OF THE PROPOSED RECOMMENDER SYSTEM

In the following Section, we implement the Ubiquitous fog recommender system in smart shopping by deploying five fog servers. We perform a set of experiment based on a real-world dataset.

A. Integration of Ubiquitous Fog-Based RS in Smart Shopping

Ubiquitous fog recommender system is evaluated using a dataset containing ID address, destination IP address, connected fog server IP address, and the localization of the mobile user. We treat each ID address as a user, because it is a unique identifier of his Mobile in our fog environment. An item could refer to Idsensor connected to a product or a web site visited by a person. In our use case, we have used many types of nodes in the three floors. Static nodes such as beacons attached to different products and fog nodes in the different floors. Also, we use mobile nodes such as mobile phone. Each mobile user is identified by his mobile's API. Mobile user sensor is detected by mobile devices which contact the fog node with the proximity information. We have deployed also the fog-based hybrid recommender system on each fog server and an Alibaba cloud server. The mobile user can access the Internet and use smart devices by connecting to the corresponding fog server. The dataset is obtained from the deployed fog servers.

We have used different platforms to simulate fog servers. A typical platform is Windows10 OS with Intel Core i5 CPU@2.7 GHz and 16 GB memory. All algorithms were implemented using python with the following installed libraries: NumPy, pandas, matplotlib, scikit-learn, nltk, scipy. Anaconda has been used for python package management and deployment. We collect 200 records of different users' interaction with different items in the mall. All data are obtained from the deployed five simulated fog servers. Due to the small amount of data set, we use all the data and split data into 80% for training purpose, 20% as test data set.

B. Evaluation Data Utility vs RMSE and MAE

In the following experiments, each floor which includes fog server is the target level. Any user connected to fog servers deployed on these floors is considered in location. In the experiments, we vary the weight value w_j and attempt identifying the best value of each parameter to obtain the most accurate result. We use two popular evaluation methods for recommendation, mean absolute error (MAE) and root-mean-square error (RMSE), to justify our quality of the prediction. RMSE, as in (1), penalizes large errors by amplifying the differences between the predicted preferences items and the real ones:

$$\text{RMSE} = \sqrt{\textstyle\sum}(\text{test-rsl})^2/|\text{test}|. \qquad (1)$$

MAE, as in (2), is the average absolute deviation of the predicted ratings from the real ratings of items:

$$\text{MAE} = \textstyle\sum|\text{test-rsl}|/\text{test}. \qquad (2)$$

We set up a decay factor for the weight parameter wj to observe its impact on the prediction accuracy and data utility. Here, the decay factor has been set up as (w_j/n) for present location, the weight of the third level of location is $(w_j/n)^3$. wj is the weight at level j that controls prediction at each location level and affects the final prediction. The weights satisfy the following constraints: wj ε [0,1], w1+w2...+wj=1. In Figure 3, wj is fixed as 0.7, n varies from 1 to 4. This value has also been mapped to the amount of privacy levels, thus there are 4 privacy levels from ε [0, β] based on location.

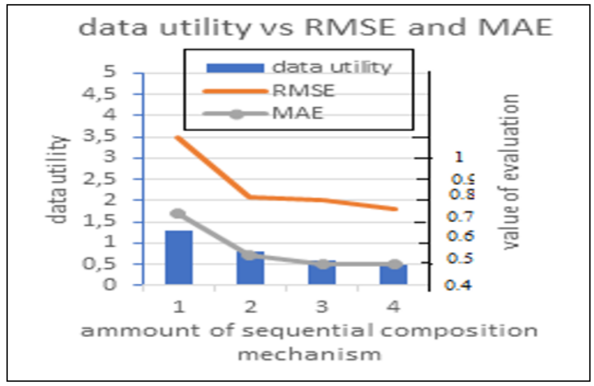

Figure 3. Data utility vs RMSE and MAE.

In Figure 3, we observe that all three measurements show a similar trend. The value of data utility decreases from 1.3 to 0. 5. The RMSE and MAE value both decreases. RMSE decreases from 3.5 to 0. 77. MAE decreases from 1.7 to 0. 5. If wj is higher, both MAE and RMSE results are better. If n is higher, both evaluation results are better as well. However, the trend becomes weaker**.**

C. Evaluation Values of Precision and Recall

Here, also each floor which includes fog server is the target level. w_k is the weight value of the target level. As a result, we need to be aware of the impact of recommended content size on the server. We use the recall and precision method to measure our result. Both methods (as defined in (3) and (4)) are broadly used in evaluating information retrieval and statistical classification. In general, precision represents the prediction accuracy, while recall represents the prediction scale. Ideally, both values would be high.

$$P = N\chi/N\rho. \qquad (3)$$

$$R = N\chi/N\Phi. \qquad (4)$$

$N\chi$ is the Number of correct items recommended and $N\rho$ is the Number of items recommended. $N\Phi$ is the Number of relevant and recommended items.

We vary the weight value w_k from 0 to 1 with the step size of 0,2 to observe the impact of result accuracy. We also modify the number of prediction items from 4 to 10 to observe the impact of result accuracy. In Figure 4, each band represents the change of the w_k from 0 to 1. The height of each point on a given band is the evaluation value of recall. The length between two points on a given band represents the difference between two results. Various bands represent different numbers of predictions, corresponding to items recommended to a fog server. A higher value of the prediction number means requiring more storage space on a fog server.

ISBN: 978-1-68558-077-3

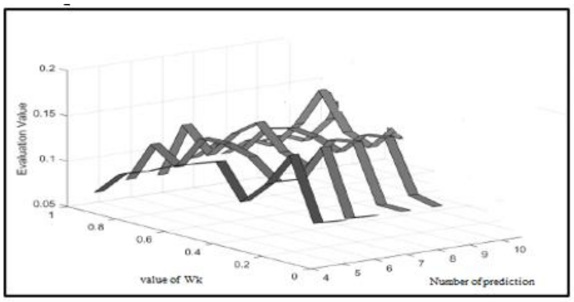

Figure 4. Evaluation value of recall.

Figure 4 shows that the more items are predicted, the higher the recall value is.

In Figure 5, the height of each point on a band represents the evaluation value of precision. We also observe that the more items are predicted, the lower the precision value. If w_k is 0.3, we obtain most accurate result on each band. So, if w_k is 0,3, we obtain the best evaluation results for both precision and recall. The prediction number does not impact the trend pattern of the evaluation value. However, the more items are predicted, the worse the predicted results are, and the higher the recall is.

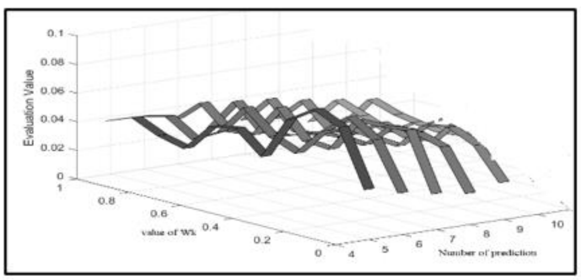

Figure 5. Evaluation value of precision.

The obtained results prove that the efficiency of Ubiquitous fog-based RS and its sample algorithms are feasible and can run independently from the cloud server. The system helps fog servers choose the most frequently requested content to purchase in order to save bandwidth, storage resources and used network resources. It also provides much more accurate recommendation results for certain items based on fog server location.

V. Conclusion and Future Work

In this research, we presented a literature review of the current recommender systems and then we discussed the different challenges in order to present our new conceptual framework of recommendation. The innovative aspect of designed model is how to address the problem of information overload ubiquitous environment, and become a tool of fog computing optimization. Further, the proposed system of recommendation improved the user's experience in the smart shopping center where we used IoT devices.

Experimental results demonstrated that Ubiquitous fog-based RS provided highly accurate and personalized recommendations to mobile users. It considered the fog server as well as contextual data of mobile user. Furthermore, it incorporated feedbacks collected from mobile users. Adding to that, it improved customers' experiences stored in the server and anticipated new users' needs. In future research, we intend to extend our proposal to areas with deep learning algorithms and reinforcement learning which can be used to improve the current research and overcome limitations.

References

[1] M . Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," IEEE Access, vol. 5, pp. 19293–19304, 2017.

[2] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," IEEE Internet of Things Journal, Vol. 3, No. 6, pp. 854–864, 2016.

[3] Christos Mettouris and George A. Papadopoulos, "Ubiquitous recommender systems," Computing, vol. 96, no. 3, pp. 223-257, 2014.

[4] G. Adomavicius, B. Mobasher, R. Francesco, and A. Tuzhilin, "Context-aware recommender systems," AI Magazine, vol. 32, no. 3, pp. 67-80, 2011.

[5] Elena Hernández-Nieves , Guillermo Hernández , Ana-Belén Gil-González , Sara Rodríguez-González , and Juan M. Corchado, "Fog computing architecture for personalized recommendation of banking products," Expert Systems With Applications, 2020.

[6] J. H. Hong, J. Ramos, and AK. Dey, "Toward personalized activity recognition systems with a semipopulation approach," IEEE Trans Human-Mach Syst, vol. 46, no. 1, pp. 101–112, 2016.

[7] Prateek Parhi, Ashish Pal, and Manuj Aggarwal, "A survey of methods of collaborative filtering techniques," 2017 International Conference on Inventive Systems and Control (ICISC), 2017.

[8] H. Yoo and K. Chung, "Mining-based lifecare recommendation using peer-to-peer dataset and adaptive decision feedback," Peer-to-Peer Networking and Applications, vol. 11, pp. 1309–1320, 2018.

[9] Hua Chen, "Personalized recommendation system of e-commerce based on big data analysis," Journal of Interdisciplinary Mathematics vol. 21, no. 5, pp. 1243-1247, 2018.

[10] X. Su and T. M. Khoshgoftaar, " A survey of collaborative filtering techniques," Advances in artificial intelligence, 2009.

[11] L. Yao, Q. Z. Sheng, A. H. Ngu, H. Ashman, and X. Li, "Exploring recommendations in internet of things," 37th international ACM SIGIR conference on Research & development in information retrieval, pp. 855–858, 2014.

[12] S. Asiri and A. Miri, "An iot trust and reputation model based on recommender systems," 14th Annual Conference on Privacy, Security and Trust (PST), pp. 561–568, 2016.

[13] S. Chakraverty and A. Mithal, "Iot based weather and location aware recommender system," 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 636–643, 2018.

[14] Sukanya Thiprak and Werasak Kurutach, "Ubiquitous computing technologies and Context Aware Recommender Systems for Ubiquitous Learning," 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE. 2015

 ISBN: 978-1-68558-077-3

[15] T. H. Chung, and J. W. Burdick, "Analysis of Search Decision Making Using Probabilistic Search Strategies," *IEEE Trans. Robot*, pp. 132–144, 2012.

[16] M. Sewak and S. Singh, "Iot and distributed machine learning powered optimal state recommender solution," Internet of Things and Applications (IOTA), International Conference on, pp. 101–106. IEEE, 2016.

[17] Weiwei Yuan , Donghai Guan, Lei Shu and Jianwei Niu, "Recommender Searching Mechanism for Trust Aware Recommender Systems in Internet of Things," Automatika, vol. 54, no. 4, pp. 427–437, 2013.

[18] S. Asthana, A. Megahed, and R. Strong, " A recommendation system for proactive health monitoring using iot and wearable technologies," AI & Mobile Services (AIMS), IEEE International Conference on, pp. 14–21. IEEE, 2017.

[19] K. Rasch, "An unsupervised recommender system for smart homes," Journal of Ambient Intelligence and Smart Environments, vol. 6, pp. 21–37, 2014.

[20] Zanda, Andrea, Santiago Eibe, and Ernestina Menasalvas, "SOMAR: A social mobile activity recommender," Expert Systems with Applications vol. 39, no. 9, pp. 8423-8429, 2017.

[21] S. P. Erdeniz, I. Maglogiannis, A. Menychtas, T. N. Trang Tran, and A. Felfernig "Recommender systems for iot enabled m-health applications," IFIP International conference on artificial intelligence applications and innovations, pp. 227–237, 2018.

[22] N. Koubai and F. Bouyakoub, "My restaurant: A smart restaurant with a recommendation system," International Journal of Computing and Digital Systems, vol. 8, pp. 143–156, 2019.

[23] Umair Javed, K. Shaukat, I. Hameed, Farhat Iqbal, Talha Mahboob Alam, and S. Luo, "A Review of Content-Based and Context-Based Recommendation Systems," Int. J. Emerg. Technol, 2021.

[24] Eva Zangerle and Christine Bauer, "Evaluating Recommender Systems: Survey and Framework," ACM Computing Surveys (CSUR), 2022.

[25] I. Hwang, M. Kim, and H. J. Ahn, "Data pipeline for generation and recommendation of the iot rules based on open text data," In Advanced Information Networking and Applications Workshops (WAINA), 30th International Conference on, pp. 238–242. IEEE, 2016.

[26] F. Ali, S. R. Islam, D. Kwak, P. Khan, N. Ullah, Sang-jo Yoo, and K.S. Kwak, "Type-2 fuzzy ontology–aided recommendation systems for iot–based healthcare," Computer Communications, vol. 119, pp. 138–155, 2018.

[27] Erion Çanon and Maurizio Morisio, "Hybrid recommender systems: A systematic literature review," Intelligent Data Analysis, vol. 21, no. 6, pp. 1487–1524, 2018.

[28] Saumil Dharia, Magdalini Eirinaki, Vijesh Jain, Jvalant Patel, Iraklis Varlamis, Jainikkumar Vora, and Rizen Yamauchi, "Social recommendations for personalized fitness assistance," Personal & Ubiquitous Computing, vol. 22, no. 2, pp. 245-257, 2018.

[29] V. Subramaniyaswamy, Gunasekarn Manogaran, R. Logesh, V. Vijayakumar, Naveen Chilamkurti, D. Malathi, and N. Senthilselvan, "An ontology-driven personalized food recommendation in IoT-based healthcare system," The Journal of Supercomputing, vol. 75, pp. 3184–3216, 2018.

[30] J. Shokeen and C. Rana, "A study on features of social recommender systems," Artificial Intelligence Review, vol. 53, no. 2, pp. 965–988, 2020.

[31] J. D. West, I. Wesley-Smith, and C. T. Bergstrom, "A recommendation system based on hierarchical clustering of an article-level citation network," IEEE Transactions on Big Data, vol. 2, no. 2, pp. 113–123, 2016.

[32] X. He and X. Ke, "Research summary of recommendation system based on knowledge graph," 3rd International Conference on Big Data Engineering, pp. 104–109, 2021.

[33] S. Lu, H. Yu, X. Wang, Q. Zhang, F. Li, Z. Liu, and F. Ning, "Clustering method of raw meal composition based on pca and kmeans," 37th Chinese Control Conference (CCC), pp. 9007–9010, 2018.

[34] A. Smiti and Z. Elouedi, "Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques," IEEE 16th International Conference on Intelligent Engineering Systems (INES)," pp. 573–578, 2012.