

Received September 21, 2020, accepted October 13, 2020, date of publication October 19, 2020, date of current version November 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031925

Tree-Based Real-Time Advertisement Recommendation System in Online Broadcasting

SEONGJU KANG^{ID}, CHAEUN JEONG, AND KWANGSUE CHUNG^{ID}, (Senior Member, IEEE)

Department of Electronics and Communications Engineering, Kwangwoon University, Seoul 01897, South Korea

Corresponding author: Kwangsue Chung (kchung@kw.ac.kr)

This work was supported in part by the Institute for Information and Communications Technology Planning and Evaluation (IITP) funded by the Korea Government (MSIT) through the Media Interaction Technology Based on Human Reaction and Intention to Content in Ultra High Definition (UHD) Broadcasting Environment under Grant 2018-0-00735, and in part by the Research Grant of Kwangwoon University, in 2020.

ABSTRACT The viewing time of media content per week through TV is still dominant. Users are exposed to numerous advertisements, such as commercials, electronic home shopping, product placement (PPL), and T-Commerce while watching TV programs. Most of the advertisement systems provide a good overview of products. However, traditional advertising services do not consider user preferences, meaning it is difficult to expect anything more than mere exposure to them. We can adopt a recommendation system to predict the preference. However, existing recommendation systems find it difficult to satisfy the real-time requirements of online broadcasting because of the large overhead incurred in preference prediction processes. In this paper, we propose a real-time recommendation system to provide personalized advertisements. The proposed system generates tree models based on user historical data. To reduce the overhead of preference prediction, we introduce a sorted HashMap that enables fast tree searches. For sophisticated preference prediction, the proposed system normalizes the users' preferences by considering the characteristics of their tree model. Finally, we conduct experiments to evaluate the performance of the proposed tree-based recommendation system.

INDEX TERMS Preference, online broadcasting, recommendation system, personalized advertisement, tree model.

I. INTRODUCTION

Information overload seriously affects the utilization efficiency of data collected online [1]. In a world of information overload, advertisement services help users purchase products they like. Users are exposed to advertisements through all kinds of media, including the TV, newspapers, magazines, radio, tablet PCs, and smartphones. In online broadcasting systems with audio/video capabilities (such as TV, tablet PCs, and smartphones), advertisement services provide information about a product to a user in many ways.

Electronic home shopping was the initial way to conduct an online (or network-based) transaction [2]. Electronic home shopping allows users to shop for products and services from the comfort of their homes, in contrast to conventional shopping, which requires a visit to stores and shopping malls [3]. With the extensive use of smart TV and IPTV services,

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Luca Bernardi .

T-commerce (T-Commerce), which conducts inquiries or transactions in digital broadcasting, has emerged as a new e-commerce business model [4]. The product placement (PPL) method exposes sponsored products or presents them to users indirectly, in contrast with other conventional advertisements. Moreover, there are various types of advertisements offered, such as web banners, pop-up advertisements, and commercials.

Advertising services have become part of the media and are displayed everywhere. Today, media content has diversified, and media platforms have evolved into user-interactive services that take into account factors such as personal interests, preferences, profiles, and rating history. However, conventional advertising services are limited in their provision of user interactive services because they are presented without any consideration of user characteristics.

To achieve a user-interactive advertisement service, it is necessary to introduce technologies that are able to predict user preferences or characteristics. The recommendation

system is a high-quality technology that predicts a user's preference based on various factors, such as search history, evaluation score, and frequency of use [5]. Therefore, it can be implemented in the advertisement domain.

To improve the performance of the recommendation system, context information should be taken into account appropriately, according to target domains [6-11]. In the media content platform, a recommendation is mainly influenced by data generated when using the content, such as viewing history, viewing time, rating score, and feedback. In the case of an online shopping platform, recommended items are influenced by data such as historical data about purchases made by users, review data, and item characteristics.

For accurate recommendations, existing recommendation systems operate by predicting unrated scores and analyzing large-scale datasets considering various contextual information of each domain [12-14]. Moreover, recommendation systems that learn user characteristics using machine learning require large computing resources [15-17]. To recommend a user-interactive advertisement, a user's preference should be predicted in real-time. However, the existing recommendation mechanisms have large computation overhead and highly complex operations. For this reason, it is not appropriate to introduce them in online broadcasting.

In this paper, we propose a personalized advertisement recommendation system considering user preferences in online broadcasting. We introduce a tree model that predicts user preferences in real-time based on user-content interaction data. Furthermore, because it is inefficient to transmit all data to a central server for personalized advertisement services, we propose a preference prediction algorithm that enables sophisticated and quick recommendations at endpoints with constrained resources and small-scale datasets.

The rest of this paper is organized as follows. We describe the advertising system in online broadcasting in Section II. We also discuss previous research into the recommendation system and the issues encountered in introducing these systems into online broadcasts. In Section III, we present the proposed advertisement recommendation system. In Section IV, we evaluate the performance of our recommendation system. Finally, the conclusions of this study are discussed in Section V.

II. RELATED WORK

In this section, we describe existing recommendation systems and discuss their issues when used in online broadcasting. Furthermore, we present the technologies required to apply recommendation systems in online broadcasting.

A. ADVERTISEMENT IN ONLINE BROADCASTING

Despite the high demand for voluntary access to media content by mobile broadband users, media content viewing time per week over TV still dominates [18]. Therefore, in the online broadcasting environment, advertisements are aired mostly through TV. Advertisements on TV have been offered

to users by various methods, such as broadcast advertising, electronic home shopping, T-Commerce, and PPL.

Commercials are usually aired at the beginning or end of TV programs. The commercial rate is determined by viewer ratings, which in turn depend on the TV programs. In addition, commercials are not associated with direct purchases; hence, it is difficult to expect impact beyond the simple exposure of products. Electronic home shopping has emerged as a new trading technique for customer convenience in modern society.

Electronic home shopping is a transaction method through interactive media with videos such as TV programs, and it provides information on products and online commerce [3]. Users conduct commercial transactions through the phone while watching home shopping programs. However, users do not know all the TV schedules; hence, they have to search for a channel to obtain online shopping services, and they cannot watch other TV programs while watching home shopping programs. Furthermore, since e-commerce is conducted for a specific product, a strategy is required to meet the needs of various users.

With the widespread use of smart TVs and IPTV platforms, e-commerce transactions using the telephone are replaced by those using a remote control. The T-commerce advertisement is an interactive service that can perform merchandise sales services with the use of a remote control while the viewer watches TV programs [4]. The user can purchase products via the T-commerce interface in the media content. T-commerce alleviates the problem of electronic home shopping by minimizing the section where advertisements are displayed. T-commerce can be inserted in any TV program, thus having the advantage of scalability.

PPL is a marketing technique in which references to brands or products are incorporated into other media content, such as a film or TV program, with specific promotional intent. PPL began to receive greater attention from advertisers and movie producers during the 1980s following product placement success in the movie *E.T. Hershey*, who advertised via PPL, claimed a 65% increase in sales of the product shown following the movie's release [19].

T-commerce and PPL offer user-interactive services by deploying existing advertising methods. However, they have limitations in providing intelligent and personalized services, as they do not take into account characteristics such as the age, region, gender, and preference of users. To address this issue, a recommendation system should be introduced to provide personalized advertisements in online broadcasting.

B. RECOMMENDATION SYSTEM

The tools for searching, visualizing, and exploring extensive collections make it easy to find the content that users want. However, using these tools is not enough if there is an enormous collection of data (such as online gathering) or if users do not know specific information about the content. To utilize online data collection effectively, recommendation technologies allow users to focus on the content they want.

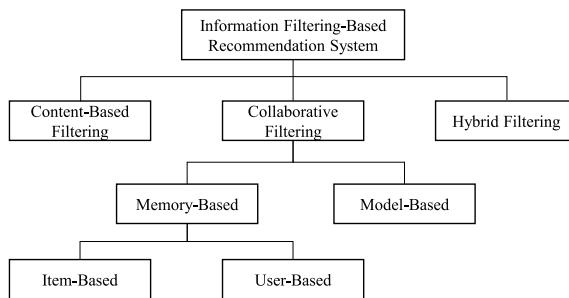


FIGURE 1. Classification of recommendation system based on information filtering.

The recommendation system helps users select the appropriate content or item in a wide variety of applications where a vast amount of data exists [20], [21]. Many recommendation algorithms have been studied based on information filtering techniques that remove duplicate or unwanted information. Fig. 1 illustrates the categorization of the recommendation system based on information filtering.

The content-based filtering (CBF) method has been studied to recommend items based on user information such as viewing time of contents, rating score, and review data [22]. The CBF predicts user preference for items by inferring the relationship among items or items and the user. To infer the relationship among items, we need to know various attributes of the products. All items are classified into keywords that mean the attributes of items. The term frequency inverse document frequency (TF-IDF) is a popular keyword extraction method that computes a weight based on the frequency of appearance of all item attributes [23]. The TF-IDF calculates the weight matrix for the attributes of items. The CBF method compares similarities among items based on the TF-IDF matrix and recommends items predicted with high rating scores.

Since the CBF method recommends based on a single user's data, the content-based approach has two main drawbacks: data sparsity and overspecialization. The data sparsity problem is a degradation of the recommendation accuracy when the user's historical data are insufficient. In contrast, when the user's historical data are adequate but are concentrated on specific categories, the system usually only recommends items very similar to what the users have already purchased or liked in the past [24]. This overspecialization of recommendations is often inconsistent with sales goals and the user's preferences.

To solve the problem of the CBF method, the collaborative filtering (CF) method considers not only the preferences of a single user but also those of other similar users. The main idea of the CF is to recommend items to a specific user based on the experiences of several users [25]. The CF methods are classified into the memory-based approach and the model-based approach. In the memory-based CF approach, many algorithms have been studied, such as user-based CF (UbCF) [26] and item-based CF (IbCF) [27].

Model-based CF is the best approach based on performance, and it uses machine learning or data mining techniques in the clustering, prediction, and classification phases.

Among the model-based CF approaches, matrix factorization (MF) based on low dimensional (low rank) factor models is one of the most popular methods since it is known to generate the most accurate model [28]. In addition, the coordinator-based approach predicts preferences by constructing networks and interacting with each group member. In [29], the coordinator integrates virtual users through various profile aggregation strategies to generate a trust-aware group. In [30], the coordinator assigns synthetic coordinates to accurately estimate the distance between the user and item. However, CF methods suffer from the sparsity problem and imbalance of rating data. For new users, there are not enough data to measure similarity. This lack of data leads to a cold start problem, which makes it impossible to predict preferences [31]. For new items, there is a first rater problem, where a recommendation cannot be made until someone uses the item [32].

Hybrid filtering methods combine the CBF and CF methods to resolve the weaknesses of each of them [11], [33]. CBF methods overcome the cold-start problem to predict a user's preference for new items, and CF methods ensure the accuracy of the recommendation system.

C. RECOMMENDATION IN ONLINE BROADCASTING

Modern video platforms such as YouTube and Netflix have already introduced a recommendation system. In video platforms, the recommendation system analyzes user characteristics and predicts the unknown score based on user historical data. The conventional recommendation systems represent relationships between the user and content using a multi-dimensional matrix. A matrix can easily represent various attribute relationships and simply implements comparison, search, and computation functions. In the CF recommendation system, similarities among items or users are calculated through matrix multiplication. When the number of users is n and the number of items is m , the iterative matrix multiplication is $O(n^2m)$. The time complexity increases tremendously as n and m increase. Many algorithms have been explored to reduce the complexity of matrix multiplication, but it is not possible to reduce the time complexity below $O(n^2)$ [34].

Several studies have been conducted to reduce the overhead of the matrix-based recommendation system. In [16], the authors proposed the discrete factorization machine (DFM)-based model to achieve efficient and accurate performance of the recommendation system by minimizing quantization loss for a large number of feature dimensions. In [35], the authors proposed an effective and efficient recommendation framework with the all-weighted scheme and fast optimization scheme. They designed a learning algorithm based on the elementwise alternating least squares (eALS) technique to efficiently optimize a matrix factorization (MF) model with variably weighted missing data.

To overcome the computational challenge of the non-deterministic polynomial time-hard determinantal point processes (NP-hard DPP), the authors of [36] proposed an algorithm to accelerate the greedy maximum a posteriori

(MAP) inference. There have been many efforts to reduce computational complexity. However, no solution has been proposed for the native cause of using a matrix. The matrix dimensions are increased depending on the input data size. In addition, the matrix factorization method requires a large amount of training data for accurate recommendations, and it cannot immediately adapt to changes in user preferences in real-time because a model is trained in offline mode. Therefore, high computational power and resources are required to process user-content interaction data generated in real-time online broadcasting.

A novel data model is required to predict a user's preferences in constrained resource environments. In [37], the authors proposed a recommendation system based on the tree-based deep model (TDM). The TDM-based system predicts the user's preference using the max-heap like tree probability formulation. Since the TDM-based system defines a tree model with a fixed size, it has a limitation in classifying the attributes of items in detail. In addition, the TDM system requires iterative learning to predict preference probability. The authors of [38] proposed a recommendation system based on a decision tree. Since the system proposed in [38] makes a recommendation decision based on predefined rules, it has scalability issues that cannot consider the characteristics of various users.

In this paper, we introduce a tree model to solve the problem of limited scalability and the high computational complexity of the matrix. We also propose a new preference prediction algorithm that can meet the real-time requirements of online broadcasting.

III. PROPOSED SYSTEM

Conventional advertisement systems attempt to instill a desire for and good impression of products on the user. In online broadcasting, these systems make the user interested in products by providing advertisement services without considering user preferences. Advertising efficiencies are affected by factors such as viewer ratings and type of TV program. To obtain high-efficiency advertising regardless of external factors, a service should be provided for products that have a high probability of being of interest to the user. We propose an advertisement recommendation system considering the preferences of users to provide personalized advertising services in online broadcasting.

In this section, we describe the proposed tree model for sophisticated and reliable preference prediction. We discuss how the proposed tree model alleviates the problems of existing recommendation systems, such as over-specialization, cold start, and real-time requirements. Furthermore, we demonstrate how to generate a tree model using the user's historical data and how to predict user preferences on the basis of a tree model.

A. TREE MODEL

To recommend personalized advertising services, user preferences should be predicted in real-time. However,

existing recommendation systems based on matrices have difficulty satisfying real-time requirements. In this paper, we propose a tree model to predict user preferences in real-time. The tree is a directed acyclic graph (DAG) without a loop and circuit. It is suitable for representing hierarchical data and can perform data searches with logarithmic time complexity. In addition, the tree is able to manage data dynamically. To predict similarity and preference for an item, matrix-based recommendation systems need all the attributes of items. Therefore, matrix structure has a limitation of scalability and cannot compress redundant data. Contrary to the matrix, the tree can reduce unnecessary computations by merging duplicated data and eliminating unused data. Therefore, the recommendation system with a tree model is preferable for predicting a user's preferences in real-time.

Fig. 2 shows the structure of the proposed tree model. The root node of the tree is described by user information to identify the user. All nodes except the root node contain information of all items that are recorded to user historical data. Items are classified into categories according to their attributes, characteristics, and properties. Furthermore, each category can be divided into subcategories. As a result, all items have a hierarchical category structure such as '*Main-category / sub-category1 / sub-category2*'.

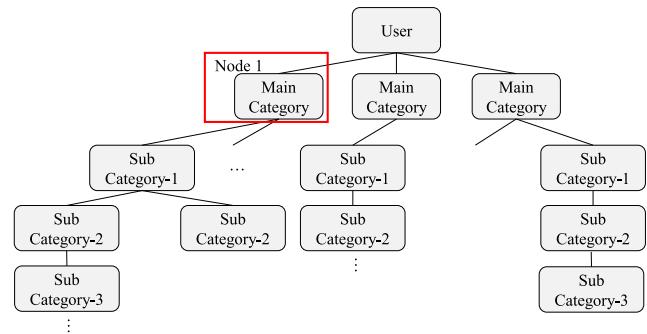


FIGURE 2. Structure of the proposed tree model.

To compare the similarity of two items, we define the longest common category (LCC) as the category node to which the two items mutually belong. Since categories are defined according to characteristics of the item, the higher the similarity between items, the longer LCC length they have. For example, there are three items '*electronic / mobile / phone*', '*electronic / mobile / tablet*', and '*electronic / office / desktop*' in historical data. The LCC_{tablet} node is '*electronic / mobile*' between phone and tablet, and the $LCC_{desktop}$ node is '*electronic*' between phone and desktop. Since the length of LCC_{tablet} is longer than $LCC_{desktop}$, it can be inferred that the phone is more similar to a tablet than a desktop.

To predict user preferences for new items, traditional recommendation systems require additional processes such as similarity comparison among items, user-content model training, and feature-map analysis. However, the proposed

system predicts the similarities of items and user preferences based on searching the LCC node in the tree model. Thus, our system does not require any extra process. We introduce a sorted HashMap data structure to reduce the time for searching LCC nodes in the tree model. The sorted HashMap stores pairs of $\langle key, value \rangle$, and the stored data are sorted by $\langle key \rangle$. If data are managed based on sorted HashMap, the binary search algorithm enables us to search $\langle value \rangle$ in logarithmic time complexity. For example, Node 1 is stored as a pair of $\langle Main\ Category, Node\ 1 \rangle$ in the HashMap of the root node.

Fig. 3 shows the time complexity of the data search according to the number of items k in the user historical data. The matrix-based recommendation systems compute similarities in $O(k)$ operations, whereas similarities of items can be computed in $O(\log_2 k \cdot \log_2 k)$ operations by using the sorted HashMap in the tree model.

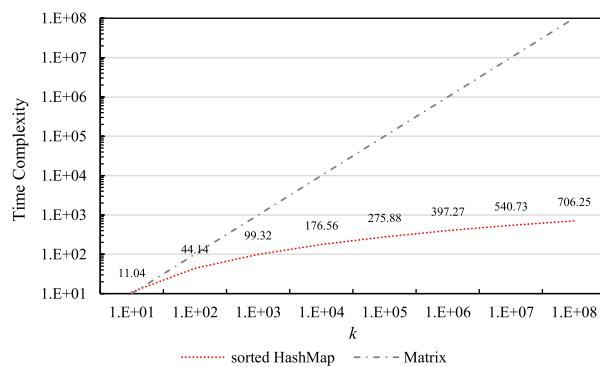


FIGURE 3. The time complexities of the prediction with matrix and sorted HashMap.

The matrix-based recommendation system has a time complexity of $O(k)$ because it calculates similarities by comparing the target item with all items. On the other hand, the proposed tree-based recommendation system performs a binary search to find subcategories with $O(\log_2 k)$ time complexity. Items are classified into one hierarchical category, and all subcategories have one parent category. In addition, the proposed tree model does not constitute any cycle, and the path from the root node to all categories of nodes is unique. In the worst case, the proposed tree model becomes a binary tree with the depth of the tree equal to $\log_2 k$. In the process of finding LCC nodes for similarity comparison, a binary search is performed at every depth. Therefore, the time complexity of the proposed tree model-based recommendation system is $O(\log_2 k * \log_2 k)$. In cases where k is greater than 1000, there is a significant difference in the performance of matrix-based and tree-based recommendation systems. Therefore, the proposed recommendation system using the tree model can significantly reduce the preference prediction time.

B. TREE MODEL FORMULATION

The proposed system creates a tree model based on the user's historical data to recommend personalized advertisements.

TABLE 1. Tree node components.

Component	Description
c	The category code of the node
n_c	The number of category c in user's historical data
N_{child}	Child nodes of the category c node
d	The depth of the category c
n_d	Total number of categories at depth d
d_{max}	The maximum depth of the descendant node
P_c	The user's preference for the category c
w_i	The weight of historical data in the i^{th} section

The tree consists of nodes that represent various attributes of hierarchical categories of items. Table 1 describes all the components for the tree node. Term c is a category that is mapped into a short string code for a simplified comparison process. We denote n_c as the number of c in the user historical data. The N_{child} stores child nodes that are subcategories of c . We implement N_{child} using a sorted HashMap data structure.

The proposed system performs the binary search based on the key value of n_{child} , iteratively until the LCC node is determined. We denote d as the depth of the current node from the root node. The value of d starts at 0 from the root node and is incremented by 1 as it moves down to the child node. Therefore, d means the granularity about how detailed c represents the characteristics of the item. The d_{max} is the largest d value of descendant nodes and indicates the maximum granularity of categories to which the current node belongs. For example, when the 'electronic / mobile / phone' data are inserted into the tree model, then d is 1 and d_{max} is 3 at the 'electronic' category node.

Term P_c denotes the user's preference for category c and is computed as given in (1). In Equation (1), d/d_{max} and n_c/n_d are normalized terms for depth and breadth, respectively. These two expressions are adopted to improve the accuracy of the recommendation. In addition, to consider the user's preference trend, we divide the historical data into s sections and assign different weight values. We denote w_i as the weight of the i^{th} section. As the latest data represent the user preference characteristics well, we define W as a monotonically increasing vector, and the sum of all elements in W is 1.

$$P_c = \sum_{i=1}^N \frac{d}{d_{max}} \cdot \frac{n_c}{n_d} \cdot w_i \quad (1)$$

For the reliability of the recommendation, we define the maturity level. The maturity level is a criterion that the tree model can sufficiently represent the characteristics of the user. We denote l_{mat} as the maturity level of the tree model. The proposed system triggers recommendations based on l_{mat} . If the size of the tree model is less than l_{mat} , the proposed system recommends the advertisement to the user without considering the preference for the item. In this case, the proposed system focuses on increasing historical data by providing all advertisements. When the model size is greater than l_{mat} , i.e., the user's data are sufficiently collected, the proposed system predicts P_{lcc} that is, P_c of the LCC node. If the tree model contains item 'electronics / mobile / tablet',

then the preference for the item ‘electronics/mobile / phone’ is predicted as P_{mobile} . The existing recommendation systems operate based on the Top-N method to determine the items to be recommended when the user is in offline mode. However, the Top-N method is difficult to satisfy the real-time requirements. To predict the user’s preference for the items in real-time, we need to design a new recommendation algorithm.

The proposed system triggers the recommendation based on the preference threshold as given in (2). We denote P_{th} as the preference threshold, b_{max} as the maximum breadth of the tree model, P_{leaf} as P_c of the leaf node, and n_{leaf} as n_c of the leaf node. The proposed system computes P_{th} adaptively based on b_{max} . If the user prefers various categories, the user tree has a large b_{max} . Therefore, P_{th} decreases. In this case, the user is more likely to receive advertisements for various categories. In contrast, if the user prefers specific categories, b_{max} decreases. When P_{th} is too large, the recommendation system is faced with an overspecialization problem. To avoid this problem, the proposed system adopts the maximum values of b_{max} and l_{mat} .

$$P_{th} = \frac{\sum P_{leaf}}{\max(l_{mat}, b_{max}) \cdot \sum n_{leaf}} \quad (2)$$

Algorithm 1 Algorithm for Recommendation

Input: User’s tree model $Tree$, item i
Output: Boolean value

```

1:  if size of  $Tree < l_{mat}$  then
2:    return true
3:  end if
4:  else
5:    Compute  $P_{lcc}$  and  $P_{th}$  by searching  $Tree$ 
6:    if  $P_{lcc} = 0$  then
7:      return true
8:    end if
9:    else if  $P_{lcc} < P_{th}$  then
10:      return false
11:    end else if
12:    else
13:      return true
14:    end else
15:  end else

```

If P_{lcc} is greater than P_{th} , the recommendation is triggered. If the LCC node does not exist in the tree model, an advertisement is recommended without considering user preferences to collect the information related to a new category. Finally, the proposed system updates the user’s tree model based on the user’s feedback data in the recommended advertising service. The details of our recommendation algorithm are described in Algorithm 1. The inputs given are the user tree model and an item, and the output is a Boolean value that determines whether the advertisement is recommended or not. If the output is true, the proposed system recommends an advertisement for the item.

IV. EXPERIMENT AND EVALUATION

In this section, to evaluate the performance of the proposed algorithm, we conduct comparative experiments with existing algorithms.

A. DATASETS

The experiments are conducted using large-scale real-world datasets: Amazon Product Data.¹ The Amazon dataset consists of 24 main categories, such as grocery, books, and music. Each category of data contains item metadata, historical data, and user review data [39], [40]. In this paper, we randomly sample 1557 users, 79344 reviews, and 55101 items from the electronic category. Sampled items are classified into 1541 categories and are assigned rating scores between 1 and 5 points. Since the item is scored by the user who purchased that item, we can define the scored items as items that the user is interested in. In addition, the higher the user’s preference for the item, the higher the rating score of the item. Therefore, items that have a rating score between 1 and 2 points are defined as cases where the user has a low preference, and items that have a rating score between 3 and 5 points are defined as items of high user preference. To evaluate the performance of the recommendation system, approximately 60000 data were used to create the preference tree model, and the remaining 19000 data were used as test data for the experiment.

B. EVALUATION MEASUREMENT

We evaluate the four metrics: precision, recall, f-measure, and accuracy, as given in (3-6) [41]. True positive (TP) means a case where a high preference item is recommended, and true negative (TN) means a case where a low preference item is not recommended. False positive (FP) means when an item with low preference is recommended, and false negative (FN) means when an item with high preference is not recommended. Precision denotes the proportion of predicted positive cases that are correctly real positives. Recall is the proportion of real positive cases that are correctly predicted as positive, meaning a hit ratio of recommendation results. The f-measure is an arithmetic mean of the metrics which is performance evaluated by considering the trade-off of precision and recall.

$$\text{Precision} = \frac{TP}{TP + TN} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

We perform comparative experiments with conventional recommendation algorithms to evaluate the performance of the proposed tree model. We compare the proposed tree

¹ <https://jmcauley.ucsd.edu/data/amazon/>

model with the maxheap tree model designed in [37] to evaluate the performance. For a fair experiment, the maxheap algorithms compute the preference of the leaf node, as given in (1). Thereafter, the preference of the non-leaf node is calculated based on the maxheap-like tree probability formulation proposed in [37]. We compare tree-based algorithms with CF-based algorithms to evaluate the robustness of the proposed tree model. The proposed-tree and maxheap-tree algorithms search the LCC node using only the user tree model to predict the user's preference. The proposed-CF and maxheap-CF algorithms predict the preference by considering both the tree model of a single user and the tree model of similar users.

To evaluate recommendation accuracy and computational efficiency, we compare tree-based algorithms with the matrix-based UbCF (matrix-UbCF) algorithm. The matrix-UbCF makes similar groups by comparing among the user feature matrices that are calculated based on the cosine similarity formula [42]. The user's preference for the target item is predicted by considering all feature matrices of the similar group. In addition, we compare the proposed-tree algorithm with the knowledge-based CF (knowledge-CF) algorithm. The knowledge-CF algorithm filters similar items related to user historical data using semantic queries and predefined rules based on user ontology. We implement the knowledge-CF algorithm based on the decision tree-based recommendation system proposed in [38]. Unlike other recommendation algorithms, the knowledge-CF assigns different weights to each rating score. That is, the knowledge-CF considers both negative feedback (1-2 points) and positive feedback (3-5 points).

We conduct the experiment according to the similar group size k of CF from 5 to 35. Both CF algorithms compute P_{lcc} as the average of the preference of the LCC nodes measured in the tree model of similar k users. Succinctly, we conduct comparative experiments with six algorithms: proposed-tree, proposed-CF, maxheap-tree, maxheap-CF, matrix-UbCF, and knowledge-CF. We set the l_{mat} value to 10 and divide historical data into 4 sections and configure the W vector to [0.1, 0.2, 0.3, 0.4] for each section.

C. EXPERIMENTAL ENVIRONMENT

Fig. 4 shows the system configuration for performance evaluations. For the experiments, we use 19000 data from Amazon as the real-time interaction data stream. The STB classifies the data stream received in real-time into user-specific data and predicts the preference based on the corresponding algorithms. The STB retrieves advertisement data from an external database server and forwards these data to the user device. We use the Nvidia Shield TV Set-Top Box. The Nvidia Shield is operated on the Android platform and allows us to build and execute user applications. For this reason, we implement the proposed advertisement system as an Android application.

The STB, advertisement database server, and devices have message queuing telemetry transport (MQTT) communication module for recommendation service. The MQTT broker

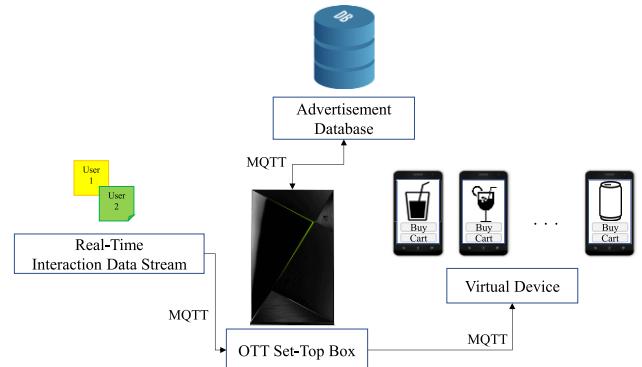


FIGURE 4. System configuration for performance evaluation.

of the STB uses Mosquitto Software [43], and the MQTT clients of the user devices and database server use the Eclipse Paho Library [44]. To conduct comparative experiments with knowledge-CF, we design the ontology on the basis of a decision tree using the protégé [45]. In addition, we utilize the Jena framework to develop semantic functions [46].



FIGURE 5. The average measured precision, recall, f-measure and accuracy. (a) proposed-CF, (b) proposed-tree, (c) maxheap-CF, (d) maxheap-tree, (e) matrix-UbCF, and (f) knowledge-CF.

D. RESULT

In the first experiment, we measure the average of four metrics for all k . Fig. 5 shows the results of the performance for four metrics of each algorithm. We evaluate how normalization terms contribute to the preference prediction by comparing the proposed tree-based algorithms with

the maxheap tree-based algorithms. The proposed-tree and proposed-CF algorithms demonstrate an accuracy improvement of between 3.7% and 6.3% in the experiment compared to the maxheap-tree and maxheap-CF algorithms, respectively. Since the recommendation algorithms based on the proposed tree model predict a user's preference for each category of items, it is possible to elaborately predict the preference when the item does not exist in historical data. Since the maxheap model does not consider the granularity level of the user's preferred categories, the shallower the depth of the LCC node is, the less accurate the preference prediction will be. In conclusion, the normalization terms allow predicting the preferences of categories not in history more accurately.

We observe that the knowledge-CF algorithm has higher performance than the matrix-UbCF and maxheap tree-based algorithms. Since the matrix-UbCF algorithm recommends items based on the feature matrices of a similar group, it demonstrates the higher performance of precision compared with other algorithms. This result shows that the cosine similarity formula is suitable for predicting the overall preference of the group. However, the matrix-UbCF algorithm has the lowest performances for the other three metrics because it does not adequately take into account the characteristics of the individual. The maxheap tree-based algorithms recommend items based on the maximum value of the preference, so historical data about a low preference for items are almost ignored. In contrast, the knowledge-CF algorithm updates the user ontology considering both negative feedback and positive feedback. Therefore, the knowledge-CF algorithm predicts preferences more accurately than the matrix-UbCF, maxheap-tree, and maxheap-CF algorithms by filtering target items and related items.

Since the proposed-tree algorithm predicts the preference based on a single user tree model, data sparsity or overspecialization problems may occur. In the second experiment, we evaluate whether the proposed tree model is reliable and sophisticated enough to express user history data by conducting comparative experiments between the proposed-tree and proposed-CF algorithm. We measure the mean absolute error (MAE) for preference prediction according to k of CF from 5 to 35. MAE is defined as (7), where N_{test} denotes the number of test data, $| \bullet_{abs} |$ means the absolute value, and s is the type of metric: precision, recall, f-measure, or accuracy.

$$MAE = \frac{1}{N_{test}} \sum_{i \in s} |R_{tree,i} - R_{CF,i}|_{abs} \quad (7)$$

Fig. 6 shows the MAE for the measured performance of the proposed-tree and proposed-CF algorithms. We can observe that precision and recall have a trade-off relationship and that the precision of proposed-CF decreases as the k value increases. As the k increases, the characteristics of a similar group are reflected more than the user characteristics in preference prediction processes, which causes a problem that the user's preference is overestimated. When k is greater than 30, the result shows that the precision of the proposed-CF

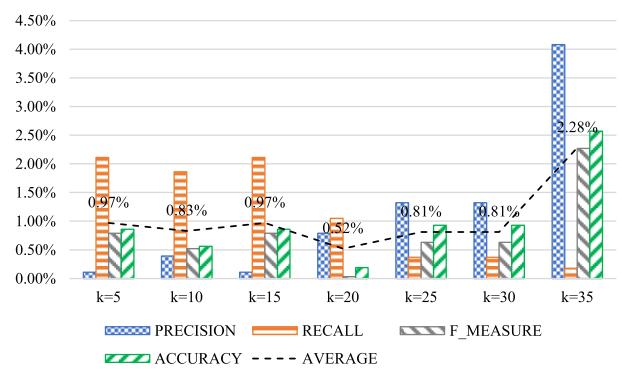


FIGURE 6. Mean absolute error of preference between proposed-tree and proposed-CF algorithms according to k .

decreases because the user's characteristics are not properly considered. In addition, from Fig. 5, we confirm that the proposed-tree algorithm has achieved better performance compared to the proposed-CF algorithm. The proposed tree model records user preferences for each category to which the item belongs, from the main category to the sub-category. The proposed system predicts user preference for an item by searching for an LCC node in the tree model. Therefore, it is possible to predict the preferences of new items more accurately and more reliably than the existing recommendation systems. This observation shows that the proposed system predicts preferences for items without data sparsity and overspecialization problems.

TABLE 2. Execution time for recommendation trigger.

Algorithm	Preference prediction time (ms)
Proposed-CF	16755.7
Proposed-tree	12.45
Matrix-UbCF	53261.3
Knowledge-CF	34414.8

In the final experiment, we measure the execution time for the recommendation trigger. Table 2 presents the results of measuring the latency of the proposed-CF, proposed-tree, Matrix-UbCF, and Knowledge-CF algorithms. Since the maxheap tree-based algorithms operate the same mechanism as the proposed tree-based algorithms, we do not present the experimental results in Table 2. The proposed-CF algorithm takes approximately 16755 ms on average to compare all tree models to search the group similar to the user. On the other hand, the proposed-tree algorithm takes approximately 12 ms because it predicts the preferences for item categories in the user tree model without searching for a similar group.

We expect that employing the caching mechanism for similarity groups in the proposed-CF algorithm will cause a further reduction in prediction time. However, the caching mechanism requires additional memory and introduces the overhead of retrieving cached data. The matrix-UbCF predicts preferences by performing matrix multiplication,

which has a high time complexity, so it takes longer to perform recommendations than tree-based algorithms. In particular, it takes approximately 3 times more execution time than the proposed-CF algorithm that compares all user trees. The knowledge-CF algorithm, which has similar performance to the proposed-CF algorithm, takes approximately twice as much time due to the overhead in reconfiguring the ontology and filtering the related items with semantic queries. Consequently, the proposed-tree algorithm satisfies real-time requirements and provides better prediction performance than that of the proposed CF algorithm without additional computations or requirements.

V. CONCLUSION

In this paper, we presented various techniques that enable the realization of personalized advertising services in online broadcasting. To provide personalized advertising services, the advertisement system needs to know the user preferences of items. The recommendation algorithm allows the advertisement system to predict user preferences. However, existing recommendation systems suffer from the large overhead incurred in learning the model for preference prediction or in selecting groups of similar users. In content-based filtering methods, overspecialization and data sparsity problems may occur as preference is predicted based on the historical data of a single user.

To resolve these issues, we proposed tree-based real-time advertisement recommendation system. We adopted a tree model to predict a user's preferences in real-time. In addition, we employed a sorted HashMap data structure for fast tree searching. To alleviate the cold start and overspecialization problem, the proposed system normalizes the preferences of users considering the granularity level of the user's preferred categories and characteristics of the user. In addition, to solve the data sparsity problems, we define the maturity level, which is a criterion that the tree model can sufficiently represent the characteristics of the user. Finally, we conducted comparative experiments that validate the effectiveness of the proposed recommendation system, both in accuracy and the robustness of recommendation. We implemented our system based on the Nvidia Shield TV Set-Top Box. In addition, we verified that the proposed system satisfies real-time requirements in online broadcasting and alleviates problems such as cold start, data sparsity, and overspecialization.

REFERENCES

- [1] C. Yin, L. Shi, and J. Wang, "Improved collaborative filtering recommendation algorithm based on differential privacy protection," in *Advanced Multimedia and Ubiquitous*, vol. 518. Singapore: Springer, 2019, pp. 253–258.
- [2] A. E. Grant, K. K. Guthrie, and S. J. BALL-ROKEACH, "Television shopping: A media system dependency perspective," *Commun. Res.*, vol. 18, no. 6, pp. 773–798, Dec. 1991, doi: [10.1177/009365091018006004](https://doi.org/10.1177/009365091018006004).
- [3] J. Gould and T. F. Golob, "Shopping without travel or travel without shopping? An investigation of electronic home shopping," *Transp. Rev.*, vol. 17, no. 4, pp. 355–376, Oct. 1997, doi: [10.1080/01441649708716991](https://doi.org/10.1080/01441649708716991).
- [4] T. Kang, "A new business model for E-commerce: T-commerce and its implications on financial institutions," *J. Global Bus. Trade*, vol. 12, no. 1, pp. 37–42, 2017. [Online]. Available: <https://ssrn.com/abstract=2998559>
- [5] P. Kumar and R. S. Thakur, "Recommendation system techniques and related issues: A survey," *Int. J. Inf. Technol.*, vol. 10, no. 4, pp. 495–501, Dec. 2018, doi: [10.1007/s41870-018-0138-8](https://doi.org/10.1007/s41870-018-0138-8).
- [6] D. Ayata, Y. Yaslan, and M. E. Kamasak, "Emotion based music recommendation system using wearable physiological sensors," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 196–203, May 2018.
- [7] C. Iwendi, S. Khan, J. H. Anajemba, A. K. Bashir, and F. Noor, "Realizing an efficient IoT-assisted patient diet recommendation system through machine learning model," *IEEE Access*, vol. 8, pp. 28462–28474, 2020.
- [8] Z. Zhang, Y. Xiao, W. Zhu, X. Jiao, K. Zhu, H. Deng, and Y. Shen, "A context-aware recommendation system based on latent factor model," in *Proc. 13th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Jul. 2017, pp. 1–6.
- [9] P. Stakhievich and Z. Huang, "An experimental study of building user profiles for movie recommender system," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun.; IEEE 17th Int. Conf. Smart City; IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Zhangjiajie, China, Aug. 2019, pp. 2559–2565.
- [10] C.-D. Wang, Z.-H. Deng, J.-H. Lai, and P. S. Yu, "Serendipitous recommendation in E-Commerce using innovator-based collaborative filtering," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2678–2692, Jul. 2019.
- [11] Y. Park, J. Oh, and H. Yu, "RecTime: real-time recommender system for online broadcasting," *Inf. Sci.*, vols. 409–410, pp. 1–16, Oct. 2017, doi: [10.1016/j.ins.2017.04.038](https://doi.org/10.1016/j.ins.2017.04.038).
- [12] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. Lun Lee, "Billion-scale commodity embedding for E-commerce recommendation in alibaba," 2018, *arXiv:1803.02349*. [Online]. Available: <http://arxiv.org/abs/1803.02349>
- [13] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based Big-Data-Driven routing recommendation approach for heterogeneous networks," *IEEE Netw.*, vol. 33, no. 1, pp. 64–69, Jan. 2019.
- [14] N. Z. Nizam and J. A. Jaafar, "Interactive online advertising: The effectiveness of marketing strategy towards customers purchase decision," *Int. J. Hum. Technol. Interact.*, vol. 2, no. 2, pp. 9–16, 2018.
- [15] W.-C. Kang and J. McAuley, "Candidate generation with binary codes for large-scale Top-N recommendation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1523–1532.
- [16] Y. Zhang, H. Yin, Z. Huang, X. Du, G. Yang, and D. Lian, "Discrete deep learning for fast content-aware recommendation," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 717–726, doi: [10.1145/3159652.3159688](https://doi.org/10.1145/3159652.3159688).
- [17] S. Chou, J. R. Jang, and Y. H. Yang, "Fast tensor factorization for large-scale context-aware recommendation from implicit feedback," *IEEE Trans. Big Data*, vol. 6, no. 1, pp. 201–208, Mar. 2020.
- [18] F. Boronat, M. Montagud, D. Marfil, and C. Luzon, "Hybrid broadcast/broadband TV services and media synchronization: Demands, preferences and expectations of Spanish consumers," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 52–69, Mar. 2018.
- [19] J. A. Karrh, K. B. McKee, and C. J. McKee, "Practitioners' evolving views on product placement effectiveness," *J. Advertising Res.*, vol. 43, no. 2, pp. 138–149, Jun. 2003.
- [20] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "Scientific paper recommendation: A survey," *IEEE Access*, vol. 7, pp. 9324–9339, 2019.
- [21] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Syst. Appl.*, vol. 89, pp. 404–412, Dec. 2017, doi: [10.1016/j.eswa.2017.08.008](https://doi.org/10.1016/j.eswa.2017.08.008).
- [22] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," in *Smart Intelligent Computing and Applications*, vol. 105. Singapore: Springer, 2018, pp. 391–397.
- [23] Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, "Hot topic detection based on a refined TF-IDF algorithm," *IEEE Access*, vol. 7, pp. 26996–27007, 2019.
- [24] O. B. Porat and M. Tennenholz, "A game-theoretic approach to recommendation systems with strategic content providers," in *Proc. 32nd Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1118–1128.
- [25] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018, doi: [10.1016/j.eswa.2017.09.058](https://doi.org/10.1016/j.eswa.2017.09.058).
- [26] H. Koohi and K. Kiani, "A new method to find neighbor users that improves the performance of collaborative filtering," *Expert Syst. Appl.*, vol. 83, pp. 30–39, Oct. 2017, doi: [10.1016/j.eswa.2017.04.027](https://doi.org/10.1016/j.eswa.2017.04.027).

- [27] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for Top-N recommendation," *ACM Trans. Inf. Syst.*, vol. 37, no. 3, p. 33, Jul. 2019.
- [28] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.
- [29] X. Wang, Y. Liu, J. Lu, F. Xiong, and G. Zhang, "TruGRC: trust-aware group recommendation with virtual coordinators," *Future Gener. Comput. Syst.*, vol. 94, pp. 224–236, May 2019.
- [30] H. Papadakis, C. Panagiotakis, and P. Fragopoulou, "SCoR: A synthetic coordinate based recommender system," *Expert Syst. Appl.*, vol. 79, pp. 8–19, Aug. 2017.
- [31] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai, "Addressing the item cold-start problem by attribute-driven active learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 631–644, Apr. 2020.
- [32] M. N. Moreno, "Web mining based framework for solving usual problems in recommender systems. A case study for movies' recommendation," *Neurocomputing*, vol. 176, pp. 72–80, Feb. 2016, doi: 10.1016/j.neucom.2014.10.097.
- [33] M. Xu and S. Liu, "Semantic-enhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks," *IEEE Access*, vol. 7, pp. 17493–17502, 2019.
- [34] V. V. Williams, *Multiplying Matrices in O(n^{2.373}) Time*. Stanford, CA, USA: Stanford Univ. Press, 2014.
- [35] H. Li, X. Diao, J. Cao, and Q. Zheng, "Collaborative filtering recommendation based on all-weighted matrix factorization and fast optimization," *IEEE Access*, vol. 6, pp. 25248–25260, 2018.
- [36] L. Chen, G. Zhang, and E. Zhou, "Fast greedy map inference for determinantal point process to improve recommendation diversity," in *Proc. NIPS*, 2018, pp. 5622–5633.
- [37] H. Zhu, "Learning tree-based deep model for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2018.
- [38] S. S. Rathore and S. Kumar, "A decision tree logic based recommendation system to select software fault prediction techniques," *Computing*, vol. 99, no. 3, pp. 255–285, Mar. 2017.
- [39] R. He and J. McAuley, "Ups and Downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. 25th Int. Conf. World Wide Web*, Montreal, QC, Canada, 2016, pp. 507–517.
- [40] J. McAuley, C. Targett, Q. Shi, and A. V. D. Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Santiago, Chile, 2015, pp. 43–52.
- [41] M. W. David, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Tech.*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: <http://www.bioinfo.in/contents.php?id=51>
- [42] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 709–720.
- [43] Mosquitto. Accessed: Feb. 10, 2020. [Online]. Available: <http://mosquitto.org>
- [44] Eclipse Paho. Accessed: Feb. 10, 2020. [Online]. Available: <https://www.eclipse.org/paho/>
- [45] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen, "Creating semantic Web contents with protege-2000," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 60–71, Mar. 2001.
- [46] B. McBride, "Jena: A semantic Web toolkit," *IEEE Internet Comput.*, vol. 6, no. 6, pp. 55–59, Nov. 2002.



SEONGJU KANG received the B.S. degree from the Electronics and Communications Engineering Department, Kwangwoon University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include intelligent recommendation systems, semantic web, and the IoT.



CHAEUN JEONG received the B.S. degree from the Electronics and Communications Engineering Department, Kwangwoon University, Seoul, South Korea, in 2019, where she is currently pursuing the Ph.D. degree. Her research interests include recommendation systems, semantic web, and deep learning algorithm.



KWANGSUE CHUNG (Senior Member, IEEE) received the B.S. degree from the Electrical Engineering Department, Hanyang University, Seoul, South Korea, the M.S. degree from the Electrical Engineering Department, Korea Advanced Institute of Science and Technology (KAIST), Seoul, and the Ph.D. degree from the Electrical Engineering Department, University of Florida, Gainesville, FL, USA. He spent a period of ten years with the Electronics and Telecommunications Research Institute (ETRI) as a Research Staff. He was an Adjunct Professor with KAIST, from 1991 to 1992. He was also a Visiting Scholar with the University of California, Irvine, from 2003 to 2004. His research interests include communication protocols and networks, QoS mechanisms, and the intelligent IoT systems.