Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.022827

check for updates

ARTICLE

Short Video Recommendation Algorithm Incorporating Temporal Contextual Information and User Context

Weihua Liu[1], **Haoyang Wan**[2,*] **and Boyuan Yan**[2]

[1]Network Security and Information Management Center, Jining University, Jining, 272000, China

[2]School of Computer Science, Qufu Normal University, Rizhao, 276800, China

*Corresponding Author: Haoyang Wan. Email: haoyangwan.w@gmail.com

Received: 28 March 2022 Accepted: 23 May 2022

ABSTRACT

With the popularity of 5G and the rapid development of mobile terminals, an endless stream of short video software exists. Browsing short-form mobile video in fragmented time has become the mainstream of user's life. Hence, designing an efficient short video recommendation method has become important for major network platforms to attract users and satisfy their requirements. Nevertheless, the explosive growth of data leads to the low efficiency of the algorithm, which fails to distill users' points of interest on one hand effectively. On the other hand, integrating user preferences and the content of items urgently intensify the requirements for platform recommendation. In this paper, we propose a collaborative filtering algorithm, integrating time context information and user context, which pours attention into expanding and discovering user interest. In the first place, we introduce the temporal context information into the typical collaborative filtering algorithm, and leverage the popularity penalty function to weight the similarity between recommended short videos and the historical short videos. There remains one more point. We also introduce the user situation into the traditional collaborative filtering recommendation algorithm, considering the context information of users in the generation recommendation stage, and weight the recommended short-form videos of candidates. At last, a diverse approach is used to generate a Top-K recommendation list for users. And through a case study, we illustrate the accuracy and diversity of the proposed method.

KEYWORDS

Recommendation algorithm; user contexts; short video; temporal contextual information

1 Introduction

With the popularity of smartphones and the advent of the era of unlimited traffic, short videos with fragmented content are more accessible to win the favor of the public than the traditional way of browsing pictures and texts. Consequently, short videos quickly occupy every aspect of people's life. Taking TikTok as an example, it has quickly taken over every aspect of people's lives because it integrated music, content, and ideas within a relatively short video length, which quickly meets users' needs. Based on the "2020 China Network Audiovisual Development Research Report", the scale of Chinese network audiovisual users has exceeded 900 million, short video users have reached 820

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

million and they spend an average of nearly 2 h a day watching short videos. The usage rate of short video software increases year by year, becoming the most widely used video media by Internet users. We can see the growth frequency in Fig. 1 [1].

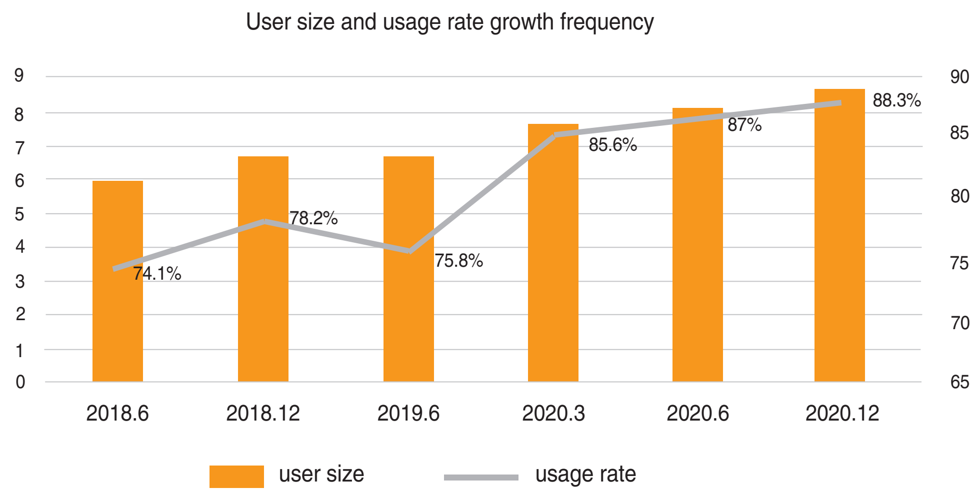

Figure 1: User size and usage rate growth frequency

In short video software, users who share and enjoy the moment of life customarily play the roles of viewer and producer simultaneously. However, the ensuing information overload makes it arduous to exert a tremendous fascination on users without an efficient recommendation algorithm. Hence, the recommendation system came into being. In contrast, much of the research in recommendation systems is based on collaborative filtering and content-based recommendation [2,3]. The idea of recommendation based on collaborative filtering is to find some similarity in the group's behavior. Consequently, the system makes decisions and recommendations for users in this way.

The above algorithms are divided into two categories: User-based collaborative filtering and item-based collaborative filtering. Unfortunately, the strategy also has some downsides: (1) The sparse interaction data and imprecise similarity method lead to limited recommendation effect. In addition, when faced with frequently updated data [4–7], the prediction efficiency of collaborative filtering is relatively low. (2) Content-based similarity measurement relies on a host of annotation data and does not necessarily reflect the similarity perceived by users, which greatly reduces the personalized effect of video recommendation.

This paper proposes a collaborative filtering algorithm which integrates time context information and user context. Similarly, it combines the specific context of user life and meets the requirements of personalized recommendation. Thus, the high-quality videos which receive little attention can enter the public view and achieve personalized and diversified recommendations.

Therefore, we propose an idea that is a collaborative filtering algorithm integrating time context information and user context. It has the characteristics of focusing on exploring and expanding of users' interests so that niche but high-quality videos can enter the public view and complete diversified personalized recommendations.

2 Related Work

2.1 Content-Based Recommendations

Content-based recommendations mainly use the meta information of items to make recommendations. After obtaining item tags through content analysis, we recommend items similar to those items that users have enjoyed in the past through their hobby records [8,9]. The implementation flow of the algorithm is shown in Fig. 2. The main operation process of this recommendation algorithm is divided into three parts: Firstly, all the items are extracted and recorded. Secondly, the interest model is constructed based on the analysis of user behavior data. Finally, the new items are sorted after the similarity calculation, and the items with higher similarity are recommended for the users. Whereas there remain critical limitations in content-based recommendation algorithm, and it mainly utilizes the information of items to make recommendations. Here is an example: In the video field, this algorithm only recommends videos with similar content for users (the content-based recommendation algorithm is invariably used in the "Guess Your Favorite" section on the home page of IQIYI.).

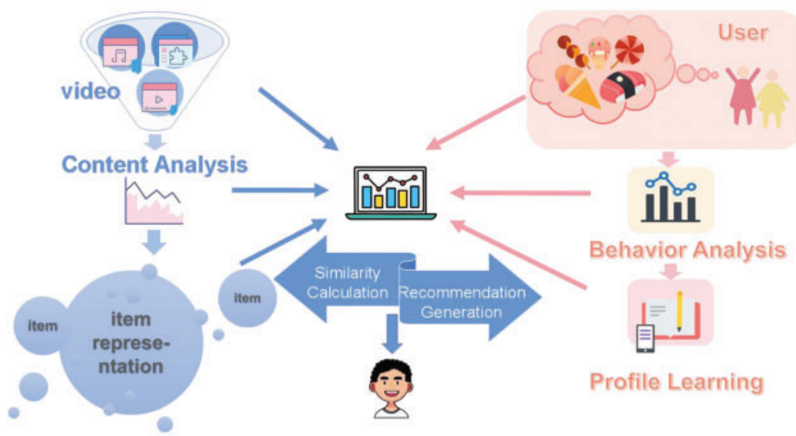

Figure 2: The main process of content-based recommendation

The idea of content-based recommendation is shown in Fig. 3 below. However, for short video recommendation, it is formidable to extract interest tags from content-based recommendations. Consequently, the emergence of collaborative filtering can not only meet the "point" recommendation to users, but also meet the situation that the range of user interests is still unchanged.

2.2 Collaborative Filtering Recommendation

The interactive information of users and the items for recommendation are usually used by the collaborative filtering algorithms [10–13]. It seems to be a recommendation algorithm under collective wisdom, which is widely used in the field of recommendation systems. Generally, collaborative filtering-based recommendations can be mainly divided into three categories: User-based collaborative filtering, item-based collaborative filtering, and model-based collaborative filter [14]. User-based collaborative filtering algorithm is mainly based on the idea that "the target user will like the items that similar users like." Accordingly, three steps are given in algorithm: (1) We leverage the similarity between users to find out the items similar to the target user's preference items. (2) Target users' score is predicated on the corresponding items. (3) Items with the highest prediction score are recommended to target users. So, the principle is shown in Fig. 4.

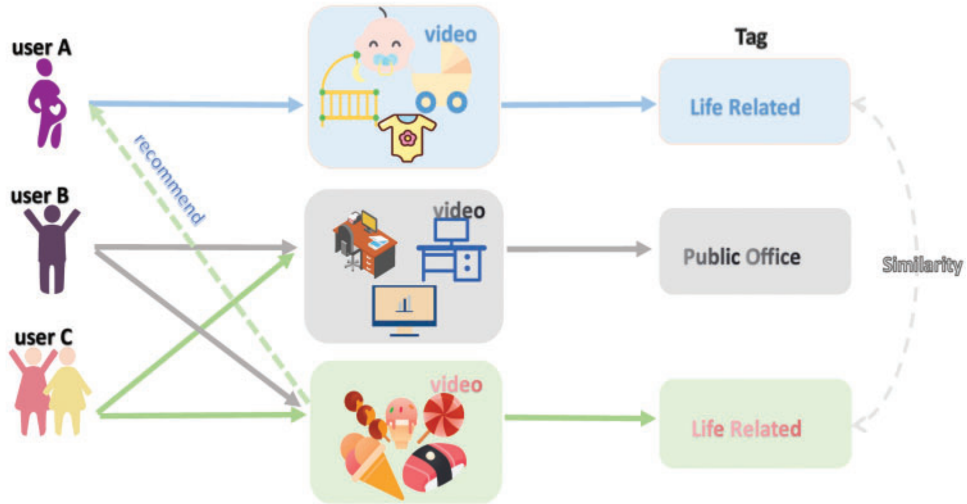

Figure 3: Content-based recommendation

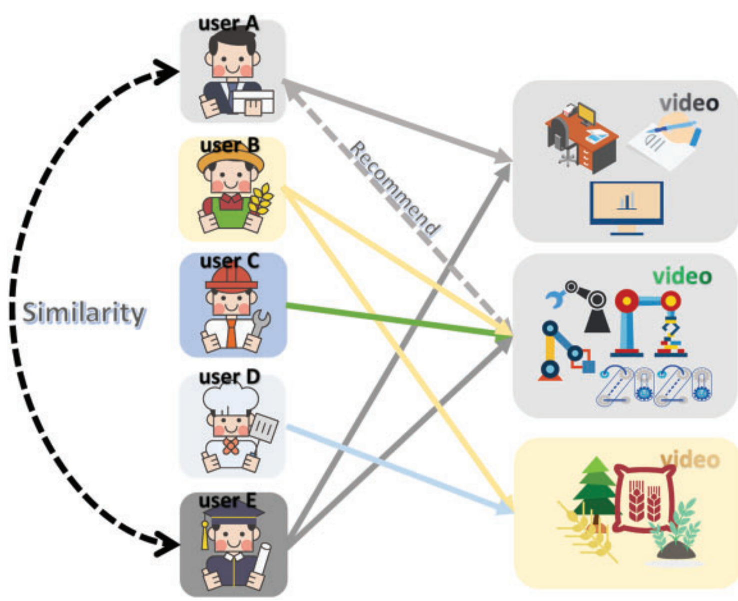

Figure 4: User-based collaborative filtering

Item-based collaborative filtering algorithm needs to calculate the similarity between similar items, and then recommend Top-N items with high similarity ranking to users. The implementation principle is shown in Fig. 5. In order to avoid the influence of sparse data on the accuracy of recommendation algorithm, a model-based collaborative filtering recommendation algorithm is proposed in [15]. This algorithm mainly distills neural network, cluster analysis, hidden semantic model, and matrix decomposition [16] to predict the score and recommendation of blank data.

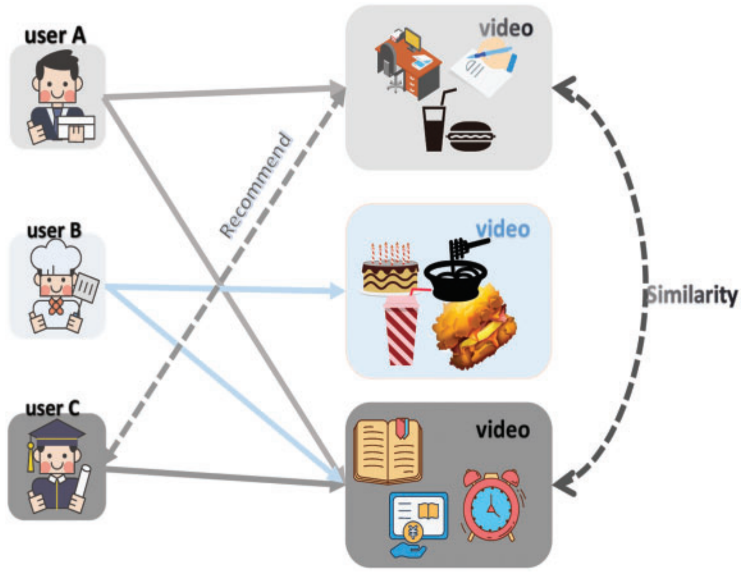

Figure 5: Item-based collaborative filtering

Whereas, dynamic changes in user preferences and practical problems of cold start [17–23] are arduous to figure out, resulting in certain deficiencies in the existing methods. Therefore, the recommendation system can not provide users with accurate personalized services (Cold start condition is a new form of problem [24–26], which is reflected between users and projects.). To update the recommended videos in real-time and solve the cold start problem, we adopt a collaborative filtering recommendation algorithm that incorporates temporal contextual information and user contexts.

2.3 User Situation

The concept of temporal contextual information has been incorporated into collaborative filtering recommendation algorithms, which made it capture more individuals' attention [27–30]. Moreover, users may expect various recommendation results in divergent contexts [31–34]. In recommendation, take the user's living habits as an example. Before going to bed, most users may prefer to watch a short video as sleep assistance. Similarly, we infer that users prefer to browse videos reflecting national pride during the national day. Therefore, incorporating the conception of user context into algorithm makes it reap huge fruits, so as to realize the diversity of recommendations.

2.4 Hybrid Recommendation

Various conditions often constrain users' decision-making process [35–40]. Similarly, in the field of recommendation systems, none of the recommendation algorithms can be perfectly compatible with all scenarios. Hence, according to the report of studying, targeted recommendations are made in diverse scenarios. For example, item-based collaborative filtering recommendations are more suitable for situations where the number of items is extensive, and users' preferences are similar and stable over time. At the same time, user-based collaborative filtering recommendations are more socially interactive and explanatory, which are suitable for tracking hot spots and over-filtering trends. The conventional method is organically combined with different recommendation algorithms to improve the accuracy of recommendations and users' satisfaction, so as to overcome the shortcomings of algorithms.

Much of the research shows that the hybrid recommendation algorithm mainly includes a single-chip hybrid recommendation algorithm, parallel hybrid recommendation algorithm, and pipeline hybrid recommendation algorithm.

The monolithic hybrid paradigm is an end-to-end solution. That is, multiple recommendation algorithms are integrated into the same algorithm system, meanwhile, the integrated recommendation algorithm provides recommendation services uniformly. And the specific implementation logic is shown in Fig. 6.

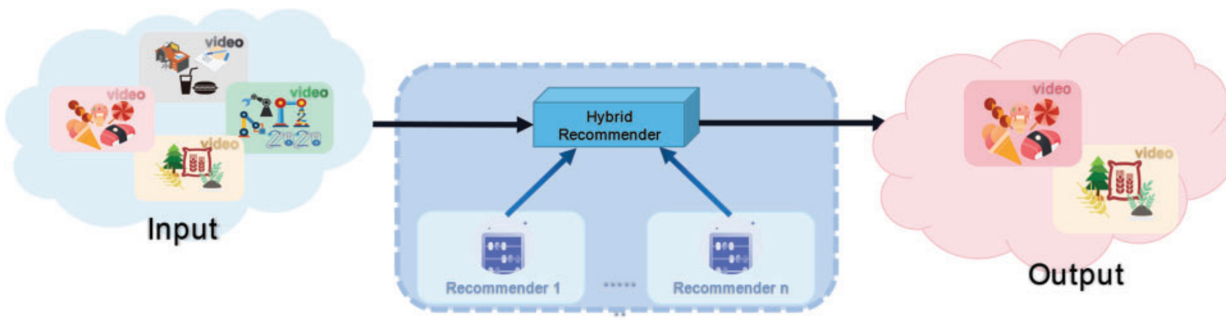

Figure 6: Monolithic hybrid paradigm implementation logic

The parallel hybrid paradigm uses multiple recommendation systems for all project data. Moreover, results generated by each recommendation system are aggregated by weighted average and other methods to generate the user's recommendation list. The specific implementation logic is shown in following Fig. 7.

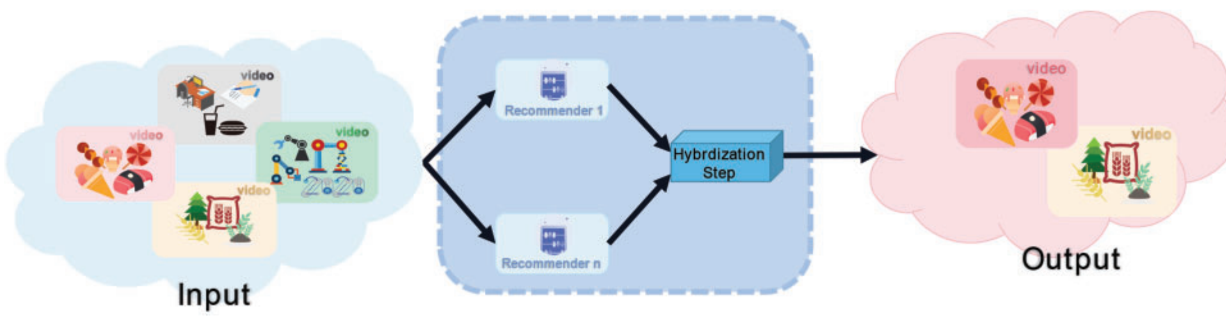

Figure 7: Parallel hybrid paradigm implementation logic

The pipeline hybrid paradigm is essentially shared in large e-commerce and social media. The YouTube platform invariably uses this hybrid scheme as an example. After promptly generating user preferences and item scores, the algorithm sorts the Top-K videos according to a more accurate recommendation method to generate a user recommendation list. The specific implementation logic is shown in Fig. 8. In the short video recommendation software, we can get better recommendation results in each context and improve user perception by mixing different recommendation algorithms.

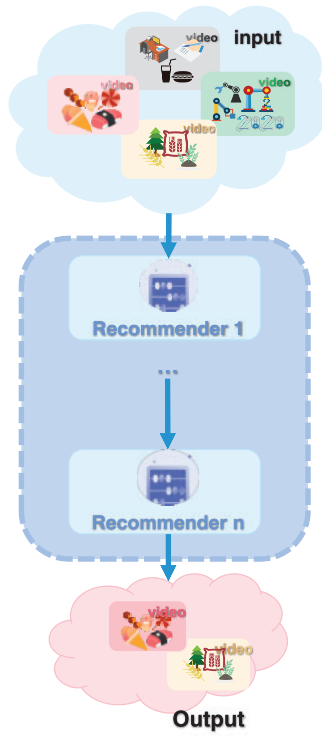

Figure 8: Pipeline hybrid paradigm implementation logic

3 Collaborative Filtering Contextual Information and User Context

3.1 Recommended Algorithm for Integrating Temporal Context Information (TI-CF Algorithm)

Collaborative filtering algorithm needs to calculate the similarity of items based on user's behavior offline. Then, according to the user's historical behavior, the calculated similarity is leveraged to recommend items for users. In TikTok, as an example, this algorithm leads to the recommendation of short video topics reflecting on head bloggers. It makes users easily "held hostage" by the trend. Such self-satisfied "happiness" that users are indulged in is quintessentially acknowledged as information cocoons. Recommendation algorithms, incorporated with temporal contextual information, are reflected in two separate temporal effects:

Variability: The recent video topics users like are more efficient than the topics with distant historical records. Consequently, different weights can describe different behaviors in recommendation.

Time interval: Video topics with a short time interval preferred by users are more important than the ones with a high time interval.

When calculating the similarity using the contextual information, the value of cosine similarity is calculated in Eq. (1):

$$sim\,(i,j) = \frac{|N\,(i)| \cap |N\,(j)|}{\sqrt{|N\,(i)| \cup |N\,(j)|}} \tag{1}$$

Because the long-tail effect affects the recommended results, to prevent user from following the heat blindly, we give weight to the punishing parameters to reduce the influence of popular head products in Eq. (2):

$$sim(i,j) = \frac{\sum_{u\subset N(i)\cap N(j)} \frac{1}{log\left(1+N(i)\right)}}{\sqrt{|N(i)|\cdot|N(j)|}} \tag{2}$$

Considering the variability, users' preferences in the near period should have higher weight than their preferences in the distant period [17]. Consequently, we introduce a time decay function in Eq. (3):

$$sim(i,j) = \frac{\sum_{u\subseteq N(i)\cap N(j)} \frac{1}{log\left(1+N(i)\right)}}{\sqrt{|N(i)|\cdot|N(j)|}} \tag{3}$$

$T\left(\left|t_{ui}-t_{uj}\right|\right)$ is a time decay function, and the calculation formula is shown in Eq. (4):

$$T\left(\left|t_{ui}-t_{uj}\right|\right) = \frac{1}{1+\varepsilon\cdot\left|t_{ui}-t_{uj}\right|} \tag{4}$$

$T\left(\left|t_{ui}-t_{uj}\right|\right)$ is the time decay parameter which uses different values to indicate the adjustment inspired by user preferences. Its value is directly proportional to the change speed of user preferences and has a more robust real-time performance.

3.2 Recommended Algorithm for Integrating the User Context (UF-CF Algorithm)

Incorporating temporal contextual information in recommendation algorithm improves diversity and accuracy. To achieve effects of personalized and diversified further, we expect to exactly predict user preferences, fit user satisfaction, and use optimization algorithms to expand recommendation styles. So users can expand their horizons of the world and jump out of the virtual environment provided via short videos. We propose a recommendation algorithm that incorporates the user context. The item score matrix is established through historical data statistics, and each item is given different scores under different situations. This matrix is shown in Table 1.

Table 1: User context matrix

Project	V_1				...	V_m			
	v_{11}	v_{12}	v_{13}	v_{1p}		v_{m1}	v_{m2}	v_{m3}	v_{mp}
i_1									
i_2									
i_4									
..									
i_n									

Here, $i_1, i_2, \ldots, i_m$ represent the short videos in software, $V_1, V_2, \ldots, V_m$ represent the considered user context, such as time, place, etc. $V_{11}, V_{12}, \ldots, V_{1p}$ represent the classification of different user contexts, such as V represents a holiday, then $v_{11}, v_{12}, \ldots v_{1m}$ can represent Dragon Boat Festival, Tanabata, national.

We get the recommendation candidate set generated for each user. And then, we use item-context matrix to calculate the rating of each item with the user's current context in the candidate set. It can be described in Eq. (5).

$$r_{i,V} = \frac{\sum_{v\in V} r_{i,v}}{|V|} \tag{5}$$

$r_{i,V}$ is the score of the item i in context V, V is the set of external contexts which the target user is currently in. $r_{i,v}$ is the score of item i in each specific context in V. The candidate set is rearranged based on the value of $r_{i,v}$, and the top N items with the highest ratings are selected to form the final recommendation list.

3.3 Generates a Recommendation List

The final recommendation list is generated in three steps: First, the recommendation list uses the recommendation algorithm incorporating temporal context information (TI-CF) to predict the scores. Second, the recommendation algorithm combined with user Context (UF-CF) is used to predict the score and generate the recommendation list K_2. At last, the predicted scores of K_1 and K_2 are given different weights to generate a new Top-N recommendation list for the user. To prevent the situation of information cocooning, randomness is added to the algorithm by using the two-eight principle, i.e., furthermore, the algorithm recommends 80% of the videos, and the system randomly assigns the remaining 20%. We introduced the specific flow of this recommendation algorithm in Fig. 9.

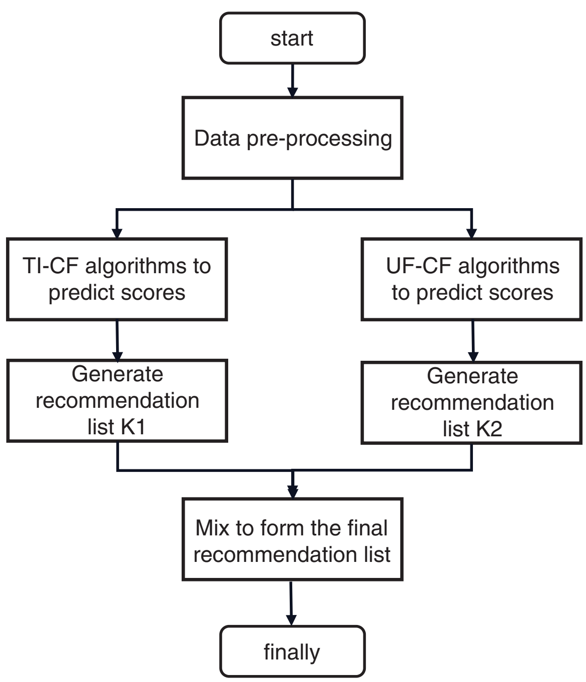

Figure 9: Specific algorithm flow

To describe this algorithm exhaustively, the symbols mentioned in this paper and their meanings are shown in Table 2.

Table 2: Symbols and their meanings

Symbol	Meaning
M	The number of users
N	The number of concatenated sets of all users preference videos
V	A set of videos that users like
C_i	The co-occurrence matrix for user i
C	The global co-occurrence matrix
V_j	The video j that the user has interacted with
S	The similarity of videos
ω	The weight of user's interest
k	The number of the recommendation

The pseudo-code of this algorithm can be described by Algorithm 1.

Algorithm 1: Short Video Recommendation Algorithm Incorporating Temporal Contextual Information and User Context

Input: User-item table
Item-context rating matrix
K: the number of recommended items
Output: a final recommendation list

for each $i \in [1, M]$ **do**
  for each $j \in [1, N]$ **do**
    for each $k \in [j, N]$ **do**
      if $j \mathrel{!}= k$
        Then $C[j][k] = C[k][j] = 1$
        else $C[j][k] = 0$
      end if
    end for
  end for
  update the co-occurrence matrix C_i for user i
  $C = C + C_i$
end for
for V_j in $\left(V - V_j\right)$
  calculate S according to Eq. (2)
  update S according to Eqs. (3),(4)
end for
  Generate TOP-K video recommendation list K_i in ascending order of S
for V_j in $\left(V - V_j\right)$
  update ω according to Eq. (5)
  calculate $S * \omega$
end for

(Continued)

Algorithm 1: (Continued)

22: Generate TOP-K video recommendation list L_2 in ascending order of $S * \omega$
23: update $L = \text{intersect}\,(K_1,\ K_2)$
24: $K * 0.2$ videos were randomly selected from the video library to generate the final recommendation list L

4 Case Study

To further illustrate the proposed algorithm in this paper, the operational steps of the algorithm are explained in this section with a case study.

Step 1: Construct the co-occurrence matrix

As shown in Table 3, the User-item table records the information of each user's favorite videos. The essential thing is to construct the short video similarity matrix W when using the recommendation algorithm. Therefore, it integrates temporal context information to itemize video recommendations for different users. In this paper, we use cosine similarity to solve the similarity degree between two videos. And the co-occurrence matrix of all users is constructed before constructing the short video similarity matrix W.

Table 3: User-item table

User	*Video*
U_1	$\{a, b, d\}$
U_2	$\{b, c, e\}$
U_3	$\{c, d\}$
U_4	$\{b, c, d\}$
U_5	$\{a, b\}$

The co-occurrence matrix can be constructed for each user based on the above data. In practice, the number of users' preferences will vary from person to person. In this case, we extend the co-occurrence matrix of each user to $N*N$, where N is the number of the connection sets of all users' preferred videos. And $N = 5$ holds in this example. Taking user U_1 as an example, Table 4 shows the co-occurrence matrix of user U_1. We construct the co-occurrence matrix of each user and then sum it to obtain the co-occurrence matrix of all items. The co-occurrence matrix of all videos is shown in Table 5.

Table 4: The co-occurrence matrix of user U_1

	Video a	*Video b*	*Video c*	*Video d*	*Video e*
Video a	0	1	0	1	0
Video b	1	0	0	1	0
Video c	0	0	0	0	0
Video d	1	1	0	0	0
Video e	0	0	0	0	0

Table 5: Co-occurrence matrix

	Video a	*Video b*	*Video c*	*Video d*	*Video e*
Video a	0	1	0	2	0
Video b	1	0	2	2	1
Video c	0	2	0	2	1
Video d	2	2	2	0	0
Video e	0	1	1	0	0

Step 2: Similarity calculation

Once obtain co-occurrence matrix, the video similarity matrix can be calculated. Hence, we take the calculated similarity of video a and video b as an example. It is known through the co-occurrence matrix N (a) = 2, that means two users like item a and item b at the same time. According to the User-Item table, we can count N (a) = 2 and N (b) = 3. And according to the improved Eq. (2), the similarity of item a and item b can be calculated in Eq. (6).

$$sim\,(a,b) = \frac{\frac{1}{log\,(1+2)}}{\sqrt{|2\,|\cdot|\,3|}} \approx 0.26 \tag{6}$$

Similarly, the video similarity matrix W can be obtained after finding the similarity of any two videos. The specific values of the video similarity matrix are shown in Table 6. To ensure the accuracy of the recommendation, we normalize the video similarity matrix. The final video similarity matrix W is shown in Table 7.

Table 6: Video similarity matrix

	Video a	*Video b*	*Video c*	*Video d*	*Video e*
Video a		0.26		0.45	
Video b	0.26		0.33	0.29	0.29
Video c		0.33		0.29	0.29
Video d	0.45	0.29	0.29		
Video e		0.29	0.29		

Table 7: Video similarity matrix

	Video a	*Video b*	*Video c*	*Video d*	*Video e*
Video a		0.58		1	
Video b	0.58		0.73	0.64	0.64
Video c		0.73		0.64	0.64
Video d	1	0.64	0.64		
Video e		0.64	0.64		

Step 3: Generate recommended list K_1

After finding the video similarity matrix W, we can make video recommendations for users accordingly. Here, we take the recommendation for U_3 as an example: User U_3 preference in the list of videos is $\{c, d\}$, and the video similarity matrix shows that the videos similar to video d are $\{a, b, c\}$. The videos similar to video c are $\{b, d, e\}$. Hence, the upcoming list of recommended videos for user U_3 is $\{a, b, e\}$. Considering the variability of user preferences, the time decay function is introduced in this step according to Eqs. (3) and (4). Then we assume that the interval time (unit/day) for user U_3 to watch different videos in Table 8.

Table 8: Interval time

	Day
$\lvert t_d - t_a \rvert$	1
$\lvert t_c - t_b \rvert$	2
$\lvert t_a - t_b \rvert$	2
$\lvert t_c - t_e \rvert$	5

The result of solving using the improved similarity Eq. (3) is:

$$sim'(d, a) = 1 \times \frac{1}{1 + 0.9 \times 1} \approx 0.53$$

$$sim'(c, d) = 0.73 \times \frac{1}{1 + 0.9 \times 2} \approx 0.26$$

$$sim'(a, b) = 0.58 \times \frac{1}{1 + 0.9 \times 2} \approx 0.21$$

$$sim'(c, e) = 0.64 \times \frac{1}{1 + 0.9 \times 5} \approx 0.12$$

We calculate the similarity between all the videos in the recommended sequence and the videos liked by U_3. Then the recommended list can be sorted according to the value of similarity $P(U_3,\ i)$. Therefore, the recommended list for U_3 is $\{a, b, e\}$ and the results are shown in Table 9.

Table 9: Video similarity matrix

	a	*b*	*e*
$P(U_3,\ i)$	0.53	0.47	0.12

Step 4: Constructing an item-context scoring matrix

In order to consider the timeliness and diversity of the recommendation, we adopt the recommendation algorithm which incorporates the user content. Making the recommendations for U_3: Through historical data statistics for U_3, we assign different ratings to each item in different contexts. We use the number of times that U_3 watches different videos in different contexts as the basis for rating. Then we get the item-context rating matrix as shown in Table 10.

Table 10: Item-context rating matrix

	Locations			*Time*		
	Dormitory	School	Home	Morning	Noon	Evening
Video a	1	0	2	1	3	3
Video b	0	3	0	1	0	4
Video c	5	2	0	1	1	3
Video d	1	0	3	0	0	2
Video e	0	0	5	2	4	5

Step 5: Generate recommendation list K_2

Adding the scoring attributes for items in different contexts, we calculate the weights for users' historical data and candidate lists dynamically. In this way, we can recommend different videos for users in different contexts and improve the temporal diversity in recommendations. This section only considers the influence of time and location factors on the recommendation results. Taking advantage of the item-context scoring matrix, the scoring and normalized result for different videos in the list can be obtained according to Eq. (6). And the rating list is shown in Table 11. We can select the first two videos with the highest score in the list, and recommend them to user. The recommendation list for user U_3 is $\{e,\ c\}$.

Table 11: Rating list of short videos

	$\mathrm{r}_{i\cdot}C$
Video a	0.625
Video b	0.5
Video c	0.75
Video d	0.375
Video e	1

Step 6: Generate final recommended list

By mixing the recommendations of the two algorithms, we can see that the final recommendation result is $\{a, b, e, c\}$. In order to increase the randomness of the algorithm and expand the overall diversity of the recommended videos, we randomly introduce videos that are not in the candidate recommendation list. Therefore, assuming a situation where we need to recommend five videos for user U_3. Then we can randomly recommend 20% videos for the user, i.e., The final recommendation result maybe $\{a, b, e, c, a'\}$.

Step 7: Comparison experiment

We add comparison into the case study, for example, by modifying temporal context information or user context information to reflect the role of the algorithm in this respect.

Modifying the interval time of user U_3 can reflect the different recommendation results of the same user. The interval time (unit/day) of user U_3 watching different videos is shown in Table 12. Moreover, the ranking results of the value of the value of similarity $P\left(U_3,\ \mathrm{i}\right)$ are shown in Table 13.

$$sim'\left(d,a\right) = 1 \times \frac{1}{1 + 0.9 \times 1} \approx 0.53$$

$$sim'\left(c,b\right) = 0.73 \times \frac{1}{1 + 0.9 \times 4} \approx 0.16$$

$$sim'\left(a,b\right) = 0.58 \times \frac{1}{1 + 0.9 \times 3} \approx 0.16$$

$$sim'\left(c,e\right) = 0.64 \times \frac{1}{1 + 0.9 \times 1} \approx 0.34$$

Table 12: Interval time

	Day
$\lvert t_d - t_a \rvert$	1
$\lvert t_c - t_b \rvert$	4
$\lvert t_a - t_b \rvert$	3
$\lvert t_c - t_e \rvert$	1

Table 13: Video similarity matrix

	a	*b*	*e*
$P\left(U_3,\ i\right)$	0.53	0.32	0.34

Judging from the above operation, the recommended result for user U_3 after modifying the time context information is $\{a, e, b\}$.

Similarly, by modifying item context rating matrix of user U_3, the influence of different context information on recommendation results can be illustrated. The modified item-context rating matrix isshown in Table 14. Rating list of short videos for user U_3 is shown in Table 15.

Table 14: Item-context rating matrix

	Locations			*Time*		
	Dormitory	School	Home	Morning	Noon	Evening
Video a	1	2	3	0	0	1
Video b	0	3	0	1	1	4
Video c	1	0	3	0	0	2
Video d	5	0	2	3	1	1
Video e	0	0	1	0	0	1

Table 15: Rating list of short videos

	$r_i.C$
Video a	0.58
Video b	0.75
Video c	0.5
Video d	1
Video e	0.17

Judging from the above operation, the recommended result for user U_3 after modifying item context information is $\{d, b\}$.

Consequently, we can get the final recommendation result for U_3 is $\{a, e, b, d, b'\}$.

Through a comparison experiment, we comprehend that diverse recommendation results can be obtained for the same user according to the distinct temporal context information and user context information. At last, it embodies the personalized diversity of the proposed algorithm from the above mentioned.

5 Conclusion and Further Discussions

To address the current situation of information cocoons in short video recommendations, we propose a collaborative filtering algorithm that integrates temporal context information and user context. Initially, by an algorithm that integrates temporal context information, we optimize the similarity of items and recommend varied items which are similar to users' preference. Additionally, the user context (such as time, location, mood, etc.) is used to obtain the recommendation list. In the end, the results of the two recommendation lists are mixed in different proportions to generate the final recommendation list. We validate the feasibility of our work through a case study from the real world. Nevertheless, the effectiveness of this method has not been specifically measured. Therefore, in the future, we will verify the accuracy and diversity of our proposed method through more comparative experiments.

There still exist several limitations in our research work. First, the paramount problem that restricts the development of short video platforms is storage, in the case of drastic growth for users and videos. Therefore, we will introduce cost-efficient edge computing technologies with energy-saving [41–44] into video recommendations to overcome existing difficulties. Second, exploiting the rich story expressed in a short video will maximize the understanding of user's preference to improve the accuracy of recommendations. So more useful content understanding technologies are necessary to be introduced into our recommendation scenario to pursue higher recommendation performances. Third, privacy concern often exists in user-related decision systems [45–49]. And how to secure sensitive user information during recommendations is another research direction in our future study. At last, network structure [50–53] and routine selection [54–57] are two main components that form a typical social network and influence the performance of short video recommendation network. Therefore, we will continue to refine our work by introducing more social network elements in the future; this way, cold start issues in short video recommendations would be alleviated.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Zhang, J., Yang, J., Wang, L., Jiang, Y., Qian, P. et al. (2021). A novel collaborative filtering algorithm and its application for recommendations in e-commerce. *Computer Modeling in Engineering & Sciences, 126(3),* 1275–1291. DOI 10.32604/cmes.2021.012112.
2. Zou, H., He, Y., Zheng, S., Yu, H., Hu, C. (2018). Online group recommendation with local optimization. *Computer Modeling in Engineering & Sciences, 115,* 217–231. DOI 10.3970/cmes.2018.00194.
3. Zhou, K., Zhao, L., Lin, T. (2018). A flexible and uniform string matching technique for general screen content coding. *Multimedia Tools and Applications, 77(4),* 1–25. DOI 10.1007/s11042-018-5624-2.
4. Sun, L., Ping, G., Ye, X. (2022). Privbv: Distance-aware encoding for distributed data with local differential privacy. *Tsinghua Science and Technology, 27(2),* 412–421. DOI 10.26599/TST.2021.9010027.
5. Qi, L., Hu, C., Zhang, X., Khosravi, M. R., Wang, T. et al. (2021). Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. *IEEE Transactions on Industrial Informatics, 17(6),* 4159–4167. DOI 10.1109/TII.9424.
6. Pang, J., Huang, Y., Xie, Z., Li, J., Cai, Z. (2021). Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution. *Tsinghua Science and Technology, 26(5),* 759–771. DOI 10.26599/TST.2021.9010026.
7. Tian, Y., Zheng, R., Liang, Z., Li, S., Wu, F. X. et al. (2021). A data-driven clustering recommendation method for single-cell RNA-sequencing data. *Tsinghua Science and Technology, 26(5),* 772–789. DOI 10.26599/TST.2020.9010028.
8. Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P. et al. (2016). Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics, 5(2),* 99–113. DOI 10.1007/s13740-016-0060-9.
9. Zhang, S., Liu, H., He, J., Han, S., Du, X. (2021). Deep sequential model for anchor recommendation on live streaming platforms. *Big Data Mining and Analytics, 4(3),* 173–182. DOI 10.26599/BDMA.2021.9020002.
10. Wang, F., Zhu, H., Srivastava, G., Li, S., Qi, L. et al. (2021). Robust collaborative filtering recommendation with user-item-trust records. *IEEE Transactions on Computational Social Systems, 9(4),* 986–996. DOI 10.1109/TCSS.2021.3064213.
11. Nitu, P., Coelho, J., Madiraju, P. (2021). Improvising personalized travel recommendation system with recency effects. *Big Data Mining and Analytics, 4(3),* 139–154. DOI 10.26599/BDMA.2020.9020026.
12. Wang, Z., Liu, J., Shen, S., Li, M. (2021). Restaurant recommendation in vehicle context based on prediction of traffic conditions. *International Journal of Pattern Recognition and Artificial Intelligence, 35(10),* 2159044. DOI 10.1142/S0218001421590448.
13. Li, Q., Cao, Z., Tanveer, M., Pandey, H. M., Wang, C. (2020). A semantic collaboration method based on uniform knowledge graph. *IEEE Internet of Things Journal, 7(5),* 4473–4484. DOI 10.1109/JIoT.6488907.
14. Su, X., Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, 2009(1),* 421425. DOI 10.1155/2009/421425.
15. Vekariya, V., Kulkarni, G. R. (2012). Hybrid recommender systems: Content-boosted collaborative filtering for improved recommendations. *2012 International Conference on Communication Systems and Network Technologies,* pp. 649–653.
16. Huang, J., Tong, Z. Feng, Z. (2022). Geographical POI recommendation for Internet of Things: A federated learning approach using matrix factorization. *International Journal of Communication Systems,* e5161.
17. Xue, Z., Wang, H. (2021). Effective density-based clustering algorithms for incomplete data. *Big Data Mining and Analytics, 4(3),* 183–194. DOI 10.26599/BDMA.2021.9020001.

18. Liu, Y., Song, Z., Xu, X., Rafique, W., Zhang, X. et al. (2021). Bidirectional GRU networks-based next POI category prediction for healthcare. *International Journal of Intelligent Systems, 37(7),* 4020–4040. DOI 10.1002/int.22710.
19. Wang, F., Li, G., Wang, Y., Rafique, W., Khosravi, M. R. et al. (2022). Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city. *ACM Transactions on Internet Technology*.
20. Peng, C., Zhang, C., Xue, X., Gao, J., Liang, H. et al. (2022). Cross-modal complementary network with hierarchical fusion for multimodal sentiment classification. *Tsinghua Science and Technology, 27(4),* 664–679. DOI 10.26599/TST.2021.9010055.
21. Sandhu, A. K. (2022). Big data with cloud computing: Discussions and challenges. *Big Data Mining and Analytics, 5(1),* 32–40. DOI 10.26599/BDMA.2021.9020016.
22. Li, T., Wang, H., He, D., Yu, J. (2022). Blockchain-based privacy-preserving and rewarding private data sharing for IoT. *IEEE Internet of Things Journal, 9(16),* 15138–15149. DOI 10.1109/JIOT.2022.3147925.
23. Liu, J., Wang, X., Yue, G., Shen, S. (2018). Data sharing in vanets based on evolutionary fuzzy game. *Future Generation Computer Systems, 81,* 141–155. DOI 10.1016/j.future.2017.10.037.
24. Li, F., Yu, X., Ge, R., Wang, Y., Cui, Y. et al. (2022). BCSE: Blockchain-based trusted service evaluation model over big data. *Big Data Mining and Analytics, 5(1),* 1–14. DOI 10.26599/BDMA.2020.9020028.
25. Zhang, W., Li, Z., Chen, X. (2021). Quality-aware user recruitment based on federated learning in mobile crowd sensing. *Tsinghua Science and Technology, 26(6),* 869–877. DOI 10.26599/TST.2020.9010046.
26. Liu, Y., Wang, F., Yang, Y., Zhang, X., Wang, H. et al. (2021). An attention-based category-aware GRU model for next POI recommendation. *International Journal of Intelligent Systems, 36(7),* 3174–3189. DOI 10.1002/int.22412.
27. Wang, S., Cong, Y., Zhu, H., Chen, X., Qu, L. et al. (2020). Multi-scale context-guided deep network for automated lesion segmentation with endoscopy images of gastrointestinal tract. *IEEE Journal of Biomedical and Health Informatics, 25(2),* 514–525. DOI 10.1109/JBHI.2020.2997760.
28. Wu, Z., Li, G., Shen, S., Lian, E., Xu, G. (2020). Constructing dummy query sequences to protect location privacy and query privacy in location-based services. *World Wide Web Journal, 24(1),* 25–49. DOI 10.1007/s11280-020-00830-x.
29. Lu, D., Tong, D., Chen, Q., Zhou, W., Zhou, J. et al. (2021). Exponential synchronization of stochastic neural networks with time-varying delays and lévy noises via event-triggered control. *Neural Processing Letters, 53(3),* 2175–2196. DOI 10.1007/s11063-021-10509-7.
30. Xu, X., Fang, Z., Zhang, J., He, Q., Qi, L. et al. (2021). Edge content caching with deep spatiotemporal residual network for iov in smart city. *ACM Transactions on Sensor Networks, 17(3),* 1–33. DOI 10.1145/3447032.
31. Zhou, F., Ma, C. (2017). Mittag-leffler stability and global asymptotically ω-periodicity of fractional-order bam neural networks with time-varying delays. *Neural Processing Letters, 47(1),* 71–98.
32. Pi, J., Hu, K., Gu, Y., Qu, L., Li, F. et al. (2016). Robust scale adaptive and real-time visual tracking with correlation filters. *Ieice Transactions on Information & Systems, 99(7),* 1895–1902.
33. Li, H., Zhu, Y., Wang, J., Liu, J., Shen, S. et al. (2017). Consensus of nonlinear second-order multi-agent systems with mixed time-delays and intermittent communications. *Neurocomputing, 251,* 115–126.
34. Xu, X., Tian, H., Zhang, X., Qi, L., He, Q. et al. (2022). DisCOV: Distributed COVID-19 detection on X-ray images with edge-cloud collaboration. *IEEE Transactions on Services Computing, 15(3),* 1206–1219. DOI 10.1109/TSC.2022.3142265.
35. Li, Z., Zhang, Y. (2021). The boundedness and the global mittag-leffler synchronization of fractional-order inertial cohen-grossberg neural networks with time delays. *Neural Processing Letters, 54(1),* 597–611.
36. Yan, W., Li, G., Wu, Z., Wang, S., Yu, P. S. (2020). Extracting diverse-shapelets for early classification on time series. *World Wide Web Journal, 23(6),* 3055–3081.

37. Yuan, L., He, Q., Chen, F., Zhang, J., Qi, L. et al. (2021). CSEdge: Enabling collaborative edge storage for multi-access edge computing based on blockchain. *IEEE Transactions on Parallel and Distributed Systems, 33(8),* 1873–1887.
38. Kim, D., Park, C., Oh, J., Yu, H. (2017). Deep hybrid recommender systems via exploiting document context and statistics of items. *Information Sciences, 417(6),* 72–87.
39. Qi, L., Song, H., Zhang, X., Srivastava, G., Xu, X. et al. (2021). Compatibility-aware web API recommendation for mashup creation via textual description mining. *ACM Transactions on Multimidia Computing. Communications and Applications, 17(1s),* 1–19.
40. Qi, L., He, Q., Chen, F., Zhang, X., Dou, W. et al. (2020). Data-driven web APIs recommendation for building web applications. *IEEE Transactions on Big Data, 8(3),* 685–698. DOI 10.1109/TBDATA.2020.2975587.
41. Liu, J., Wang, X., Shen, S., Yue, G., Li, M. et al. (2020). A Bayesian Q-learning game for dependable task offloading against DDoS attacks in sensor edge cloud. *IEEE Internet of Things Journal, 8(9),* 7546–7561. DOI 10.1109/JIOT.2020.3038554.
42. Liu, J., Wang, X., Shen, S., Fang, Z., Li, M. et al. (2021). Intelligent jamming defense using DNN stackelberg game in sensor edge cloud. *IEEE Internet of Things Journal, 9(6),* 4356–4370. DOI 10.1109/JIOT.2021.3103196.
43. Wang, T., Bhuiyan, M. Z. A., Wang, G., Qi, L., Wu, J. et al. (2019). Preserving balance between privacy and data integrity in edge-assisted Internet of Things. *IEEE Internet of Things Journal, 7(4),* 2679–2689. DOI 10.1109/JIoT.6488907.
44. Shen, Y., Shen, S., Wu, Z., Zhou, H., Yu, S. (2022). Signaling game-based availability assessment for edge computing-assisted IoT systems with calware dissemination. *Journal of Information Security and Applications, 66,* 103140. DOI 10.1016/j.jisa.2022.103140.
45. Wu, Z., Shen, S., Zhou, H., Li, H., Lu, C. et al. (2021). An effective approach for the protection of user commodity viewing privacy in e-commerce website. *Knowledge-Based Systems, 220,* 106952. DOI 10.1016/j.knosys.2021.106952.
46. Wu, B., Chen, X., Wu, Z., Zhao, Z., Mei, Z. et al. (2021). Privacy-guarding optimal route finding with support for semantic search on encrypted graph in cloud computing scenario. *Wireless Communications and Mobile Computing, 2021(17),* 1–12. DOI 10.1155/2021/6617959.
47. Wu, Z., Shen, S., Lian, X., Su, X., Chen, E. (2020). A Dummy-based user privacy protection approach for text information retrieval-sciencedirect. *Knowledge-Based Systems, 195,* 105679. DOI 10.1016/j.knosys.2020.105679.
48. Nosouhi, M. R., Yu, S., Sood, K., Grobler, M., Jurdak, R. et al. (2021). UCoin: An efficient privacy preserving scheme for cryptocurrencies. *IEEE Transactions on Dependable and Secure Computing*. DOI 10.1109/TDSC.2021.3130952.
49. Li, T., Wang, H., He, D., Yu, J. (2021). Synchronized provable data possession based on blockchain for digital twin. *IEEE Transactions on Information Forensics and Security, 17,* 472–485. DOI 10.1109/TIFS.2022.3144869.
50. Xiong, H., Fan, Y. (2021). How to better identify venture capital network communities: Exploration of a semi-supervised community detection method. *Journal of Social Computing, 2(1),* 27–42. DOI 10.23919/JSCTUP.8964404.
51. Nosouhi, M. R., Yu, S., Sood, K., Grobler, M., Jurdak, R. et al. (2022). Reconfigurable intelligent surfaces for wireless communications: Overview of hardware designs, channel models, and estimation techniques. *Intelligent and Converged Networks, 3(1),* 1–32. DOI 10.23919/ICN.2022.0005.
52. Gu, W., Gao, F., Li, R., Zhang, J. (2021). Learning universal network representation via link prediction by graph convolutional neural network. *Journal of Social Computing, 2(1),* 43–51. DOI 10.23919/JSCTUP.8964404.

53. Zhi, P., Zhao, R., Zhou, H., Zhou, Y., Ling, N. et al. (2021). Analysis on the development status of intelligent and connected vehicle test site. *Intelligent and Converged Networks, 2(4),* 320–333. DOI 10.23919/ICN.2021.0023.
54. Waggoner, P. D., Shapiro, R. Y., Frederick, S., Gong, M. (2021). Uncovering the online social structure surrounding COVID-19. *Journal of Social Computing, 2(2),* 157–165. DOI 10.23919/JSC.2021.0008.
55. Zhou, G., Pan, C., Ren, H., Wang, K., Chai, K. K. et al. (2022). User cooperation for IRS-aided secure MIMO systems. *Intelligent and Converged Networks, 3(1),* 86–102. DOI 10.23919/ICN.2022.0001.
56. Kohler, T. A., Bird, D., Wolpert, D. H. (2022). Social scale and collective computation: Does information processing limit rate of growth in scale. *Journal of Social Computing, 3(1),* 1–17. DOI 10.23919/JSC.2021.0020.
57. Liu, G., Wu, J., Wang, T. (2021). Blockchain-enabled fog resource access and granting. *Intelligent and Converged Networks, 2(2),* 108–114. DOI 10.23919/ICN.2021.0009.