Adversarial Attacks on Content-Based Filtering Journal Recommender Systems

Zhaoquan Gu[1], Yinyin Cai[1], Sheng Wang[1], Mohan Li[1, *], Jing Qiu[1], Shen Su[1], Xiaojiang Du[2] and Zhihong Tian[1]

Abstract: Recommender systems are very useful for people to explore what they really need. Academic papers are important achievements for researchers and they often have a great deal of choice to submit their papers. In order to improve the efficiency of selecting the most suitable journals for publishing their works, journal recommender systems (JRS) can automatically provide a small number of candidate journals based on key information such as the title and the abstract. However, users or journal owners may attack the system for their own purposes. In this paper, we discuss about the adversarial attacks against content-based filtering JRS. We propose both targeted attack method that makes some target journals appear more often in the system and non-targeted attack method that makes the system provide incorrect recommendations. We also conduct extensive experiments to validate the proposed methods. We hope this paper could help improve JRS by realizing the existence of such adversarial attacks.

Keywords: Journal recommender system, adversarial attacks, Rocchio algorithm, k-nearest-neighbor algorithm.

1 Introduction

Due to the rapid development of information technologies, more and more data are produced every day. People would face the problem of selecting useful information from these enormous information streams. Recommender systems play an important role in solving the problem by filtering a lot of irrelevant information and providing the users with accurate contents and services. In this paper, we study journal recommender systems (JRS) that are designed for academic researchers.

Generally, the academic researchers might spend much time in selecting an appropriate journal for submitting an academic article. If the subject of the article does not match the selected journal, the researcher would only waste time and energy during the process. Hence, JRS could be developed to provide a small number of suitable journals and the research could save much time in the journal selection process. There are many methods in developing the JSR, including content-based filtering (CBF), user-based filtering

[1] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China.

[2] Department of Computer and Information Sciences, Temple University, Philadelphia, USA.

[*] Corresponding Author: Mohan Li. Email: limohan@gzhu.edu.cn.

Received: 24 March 2020; Accepted: 28 April 2020.

(UBF), collaborative filtering (CF), knowledge graph, matrix factorization, neural networks, etc., [Cao, Zhou and Gao (2019); Herlocker, Konstan, Terveen et al. (2004); Sha, Sun and Zhang (2019); Yin, Shi, Sun et al. (2019)]. Among these methods, the neural networks-based methods could achieve high recommendation accuracy, but they cost much time in training and they are inefficient in practice. JRS based on content filtering has the advantages of fast startup speed, ease of use, and high recommendation efficiency. This kind of recommendation systems are widely adopted. Therefore, we focus on the content-based filtering JRS (CBF-JRS) in this paper.

The recommendation algorithms could achieve high efficiency and the recommender systems have achieved great success in various applications, such as website research, item recommendation, medical advice, etc., [Bin, Sun, Cao et al. (2019)]. However, the recommendation algorithms face a lot of security risks. Some extant works show that these algorithms are fragile and vulnerable to various attacks, such as shilling attacks, adversarial example attacks, data poisoning attacks and model stealing [Christakopoulou and Banerjee (2018); Gu and Rigazio (2014); Lam and Riedl (2004); Massa and Avesani (2004); O'Mahony, Hurley and Silvestre (2005)]. However, few works investigate the security problems of the JRS for academic researchers. As more and more journal recommendation systems are adopted in practice, it is necessary to study the security threats of such systems. In this paper, we study the adversarial attacks against JRS, especially against the CBF-JRS.

We first developed a CBF-JRS on the basis of two traditional algorithms (the improved Rocchio algorithm [Miao and Kamel (2011)] and the k-nearest-neighbors algorithm (k-NN) [Jiang, Pang, Wu et al. (2012)]). The dataset for training the algorithms contains about 150,000 articles' information (title, keywords and abstract) from 583 journals. Assuming the system recommends about 5% of the journals (30 journals) when users input the article information, both algorithms can achieve high recommendation accuracy (more than 90%).

Then we constructed a term-weight table and proposed an adversarial attack method. The adversarial words are chosen from the table and the inserted words can cause the system to recommend incorrect results. Considering the inputted article information, we inserted a small number of adversarial words from the table for two goals: non-targeted attacks cause the systems to recommend the article to incorrect journals, while targeted attacks cause the systems to recommend the article to a specific selected journal. We conduct extensive experiments to evaluate two kinds of attacks respectively. We show that, by inserting only at most 3 words, the recommendation accuracy could be largely reduced by 40%-60%. We summarize the main contributions of this paper as follows:

- We develop a CBF-JRS with two recommendation algorithms: the improved Rocchio algorithm and the k-NN algorithm. The recommendation accuracy has been quantitatively evaluated;
- We present adversarial attacks against the CBF-JRS for both non-targeted and targeted attacks. Specially, we construct an adversarial term-weight table and insert some adversarial words to attack the systems;

- We conduct extensive experiments to evaluate the performance of the proposed attacks. The results show that the adversarial attacks could dramatically reduce the recommendation accuracy.

The rest of the paper is organized as follows. The second section introduces some existing works, including academic recommendation and attack methods against the recommender system. The third section introduces the preliminaries. Section 4 describes the details of developing the CBF-JRS, including the recommendation algorithms. Section 5 introduces the adversarial attacks against the CBF-JRS. The experimental results are provided in Section 6 and we conclude the paper in Section 7.

2 Related work

Various researchers have been worked on recommender systems, especially in the academic field. Researchers have proposed various recommendation frameworks, most of them are based on collaborative filtering algorithms [Bin, Sun, Cao et al. (2019); Herlocker, Konstan, Terveen et al. (2004); Yin, Shi, Sun et al. (2019)], which creates a profile for each user or each recommended item to capture the features. Some algorithms construct the user-item rating matrix and apply matrix factorization methods to identify the latent semantic factors [Yi, Shen, Liu et al. (2019)]. In industrial areas, there are also some commercial academic recommender systems. For example, Google Scholar uses the Google Knowledge Graph to search for article information.

A collaborative-filtering recommendation algorithm is introduced in Wang et al. [Wang, Liu, Yang et al. (2015)], which trains the algorithm on the academic articles. Wang et al. [Wang, Li, Zhang et al. (2016)] have proposed several algorithms which take the text similarity, author similarity, intimacy and the influence among academic articles into consideration. An academic recommendation method is presented in Zhao et al. [Zhao, Wu and Liu (2016)], which works on the basis of the differences between the researchers' background knowledge and the researchers' goals. Furthermore, A recommendation framework for scientific articles is proposed in Chakraborty et al. [Chakraborty, Krishna, Singh et al. (2016)]. The framework is related to the queries of the users and it is evaluated artificially, which implies the framework is judged and improved by experts. A number of related works about recommender systems are investigated in Park et al. [Park, Choi, Kim et al. (2012)], which divides the recommender systems into eight categories to help those people who are interested in the recommendation systems.

Artificial intelligence has shown the potential in various applications including the recommendation system. However, these intelligent algorithms cannot be explained theoretically and they are vulnerable to adversarial attacks. Some of these attacks have been studied in Gu et al. [Gu, Hu, Zhang et al. (2020); Huang and Du (2014); Tian, Li, Qiu et al. (2019); Tian, Luo, Qiu et al. (2019); Xiao, Li, Huang et al. (2017)]. There are also some works on the attacks and the defenses of recommender systems. Lam et al. [Lam and Riedl (2004)] have found that users could shill recommender systems by lying to the systems in order to have their products recommended more often than those of their competitors. O'Mahony et al. [O'Mahony, Hurley and Silvestre (2005)] reviews a lot of related works on malicious attacks against recommender systems. These works imply that it is possible to mount successful attacks with little domain knowledge. A framework is also proposed in Christakopoulou et al. [Christakopoulou and Banerjee (2018)] for generating fake user profiles. The fake profiles can easily fool machine learning models for the recommendation. Among these works, few of them study the attacks against journal recommendation systems.

3 Preliminaries

In this section, we first introduce the process of constructing a content-based filtering journal recommendation system (CBF-JRS). Then, we formulate the problem of attacking the JRS for non-targeted goal and targeted goal.

3.1 System model of the CBF-JRS

We first introduce the model of a CBF-JRS. Suppose the users of the system input the article information, including the title, keywords, and abstract of an article, the system would output a list of recommended journals that best suit the article information. As depicted in Fig. 1, the CBF-JRS is constructed with different recommendation algorithms and it is trained on a journal dataset.

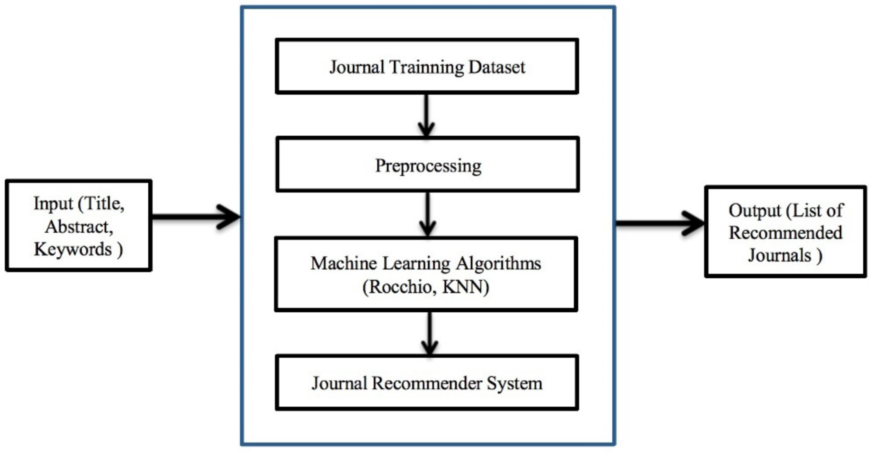

Journal Trainning Dataset
Preprocessing
Machine Learning Algorithms (Rocchio, KNN)
Journal Recommender System
Input (Title, Abstract, Keywords)
Output (List of Recommended Journals)

Figure 1: The model of the CBF-JRS

The journal dataset contains the published papers from some well-known publishers such as IEEE, Springer, Elsevier, ACM, etc. The journal information includes the full name of the journal and the ISSN. The article information includes the title, the keywords, and the abstract. These data are used to train the recommendation algorithms.

We design the recommendation algorithms on the basis of two machine learning algorithms: the improved Rocchio algorithm [Miao and Kamel (2011)] and the k-nearest-neighbors algorithm (k-NN) [Jiang, Pang, Wu et al. (2012)]. The details are provided in Section 4.

3.2 Problem definition

In this paper, we focus on the adversarial attacks against the CBF-JRS. Denote the system as S and a list of journals as $J = \{J_1, J_2, \dots, J_n\}$; considering an input article (denoted as $Article$), the system could recommend k suitable journals and we denote the recommendation results as $R(Article) = \{J_1(A), J_2(A), \dots, J_k(A)\} \subseteq J$. Suppose the article belongs to a correct journal J_i (we say the ground-truth label of the article is J_i), the system S would make correct recommendations such that $J_i \in R(Article)$. In our developed system, the article information includes $\{title,\ keywords,\ abstract\}$.

The adversarial attacks against the CBF-JRS would recommend incorrect results. We formulate the two different problems as follows:

Problem 1: Non-targeted Adversarial Attack (NTAA) makes the system S recommend incorrect journals by modifying the article information slightly.

Formally, the problem is to generate a similar article (denoted as $Article_{adv}$) such that $J_i \notin R(Article_{adv})$ and $Article_{adv}$ differs slightly from $Article$.

Problem 2: Targeted Adversarial Attack (TAA) makes the system S recommend a specific selected journal by modifying the article information slightly.

Formally, the problem is to generate a similar article (denoted as $Article_{adv}$) such that a specific journal J_{adv} could be recommended by the system ($J_{adv} \in R(Article_{adv})$) and $Article_{adv}$ differs slightly from $Article$.

The difference between the two problems lie in that the NTAA problem only confuses the recommendation result and reduces the recommendation accuracy, while the TAA problem would make the system recommend a targeted journal.

4 Design and implementation of CBF-JRS

We introduce the design and the implementation of the CBF-JRS. We first describe the preprocessing step of Fig. 1, then we present the recommendation algorithms that are adopted in the CBF-JRS.

4.1 Preprocessing

We introduce the preprocessing steps to establish a connection between the words and the articles such that an article can be efficiently represented.

Term Frequency (TF) is commonly adopted to evaluate the connection between the words and the articles. If a word appears more frequently in an article, the word can be considered as more relevant to the article. However, using the TP method is not always a good idea since words like *the, is, and* are always appeared with high frequency. These words are usually not related to the main theme of the paper. Inverse Document Frequency (IDF) is first proposed in Zhao et al. [Zhao, Wu and Liu (2016)], which assumes the more times a word appears in a corpus, the lower contribution it makes to the characteristics of a paper. IDF assigns higher weights to low frequency words in the corpus.

Term Frequency-Inverse Document Frequency (TF-IDF) combines the IDF and the TF ideas. The weight of a term (word) in a document (article) by TF-IDF can be formulated mathematically as Eq. (1),

$$W(d,t) = TF(d,t) * \log(\frac{N}{df(t)}) \quad (1)$$

where W represents the weight and N represents the number of classes. These separate words can be processed with the above theoretical basis. We can calculate the IDF value of each word in the corpus. Then, the words with very low IDF values are treated as stop words, and they are deleted from the list of words because such words are not representative. In the CBF-JRS, the words with very low frequency are usually the personal idioms of the authors, and they are not related to the overall characteristics of the journals. Hence, these words are also deleted from the list.

We regard each word as a vector and count the term frequency of those words in each article. The words appearing in the title, the keywords, or the abstract may have different weights and we assign their term frequencies as Eq. (2):

$$W(title) : W(keyword) : W(abstract) = 5:3:1 \quad (2)$$

4.2 Recommendation

In order to recommend suitable journals to the users, we need to find the similarity between the input article information and the journal information (the published articles) in the dataset. In this paper, we use the cosine similarity between two vectors (A and B) to calculate the similarity of two articles as Eq. (3)

$$\text{similarity} = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}. \quad (3)$$

When the user inputs the article information, the TF-IDF value of each word can be calculated by Eq. (1). The article can be converted to an n-dimensional vector where the non-zero dimension indicates the corresponding word has appeared at least once in the article information. Then the similarity between the article and the articles in the dataset can be calculated. In our system, we select two commonly adopted algorithms for recommendation; they are the Rocchio algorithm and the k-nearest-neighbors algorithm. We introduce how these algorithms can be applied.

Rocchio algorithm: The Rocchio algorithm [Miao and Kamel (2011)] uses a set of documents to construct a centroid vector for each class. This centroid vector can be computed as an average vector of a class of the document vectors as Eq. (4).

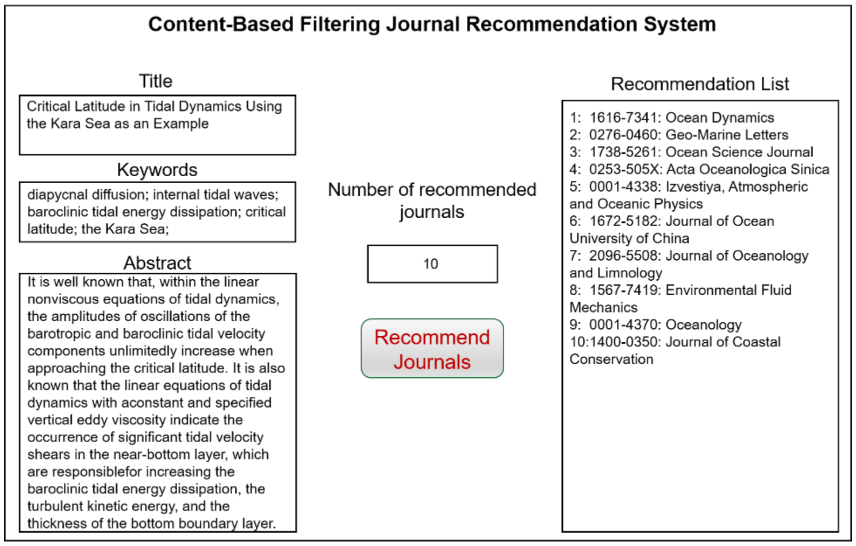
$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}_d \quad (4)$$

In our work, each journal represents a class of the algorithm's output and the centroid vector can be regarded as the average value of all articles' vectors in the class. The inputted article could be calculated on the basis of these centroid vectors to compute the cosine similarity. Then the k journals with highest similarity scores are selected as the recommendation results, where k can be determined by the user.

K-nearest-neighbor: The k-NN algorithm [Jiang, Pang, Wu et al. (2012)] is often adopted for text categorization, which is similar to the Rocchio algorithm. They both use the TF-IDF as the value of each dimension in the vector. The difference between the two algorithms is that the k-NN algorithm finds the d nearest neighbors (sorted by the similarity score from high to low) in all documents of the training set. The candidate class that a specific document belongs to would be determined by the class that its d nearest neighbors belong to. Assuming the document vector is $\boldsymbol{A}$, the algorithm works as the following equation:

$$f(A) = \operatorname{argmax}_{j} S(A, C_j) = \sum_{d_i \in K-NN} sim(A, d_i) y(d_i, C_j) \tag{5}$$

where sim is a function of similarity calculation, d_i represents a neighbor of A, and C_j represents a class. In our system, the inputted article can be regarded as the document and each journal represents each class C_j in Eq. (5).


Content-Based Filtering Journal Recommendation System

Title

Critical Latitude in Tidal Dynamics Using the Kara Sea as an Example

Keywords

diapycnal diffusion; internal tidal waves; baroclinic tidal energy dissipation; critical latitude; the Kara Sea;

Abstract

It is well known that, within the linear nonviscous equations of tidal dynamics, the amplitudes of oscillations of the barotropic and baroclinic tidal velocity components unlimitedly increase when approaching the critical latitude. It is also known that the linear equations of tidal dynamics with aconstant and specified vertical eddy viscosity indicate the occurrence of significant tidal velocity shears in the near-bottom layer, which are responsiblefor increasing the baroclinic tidal energy dissipation, the turbulent kinetic energy, and the thickness of the bottom boundary layer.

Number of recommended journals

10

Recommend Journals

Recommendation List

1: 1616-7341: Ocean Dynamics
2: 0276-0460: Geo-Marine Letters
3: 1738-5261: Ocean Science Journal
4: 0253-505X: Acta Oceanologica Sinica
5: 0001-4338: Izvestiya, Atmospheric and Oceanic Physics
6: 1672-5182: Journal of Ocean University of China
7: 2096-5508: Journal of Oceanology and Limnology
8: 1567-7419: Environmental Fluid Mechanics
9: 0001-4370: Oceanology
10:1400-0350: Journal of Coastal Conservation

Figure 2: An example of the journal recommendation system

We implemented the CBF-JRS and Fig. 2 shows an example. Suppose an oceanology researcher finishes an article and the user inputs the article information into the system. For example, the title is "Critical Latitude in Tidal Dynamics Using the Kara Sea as an Example" and the other information is shown in the figure. Actually, this article is published in the journal "*Izvestiya, Atmospheric and Oceanic Physics*" *(ISSN: 0001-4338)*, and this article is not trained in constructing the recommendation system. The figure shows that the system recommends 10 journals and the correct label of the article is in the recommendation list.

5 Adversarial attacks on JRS

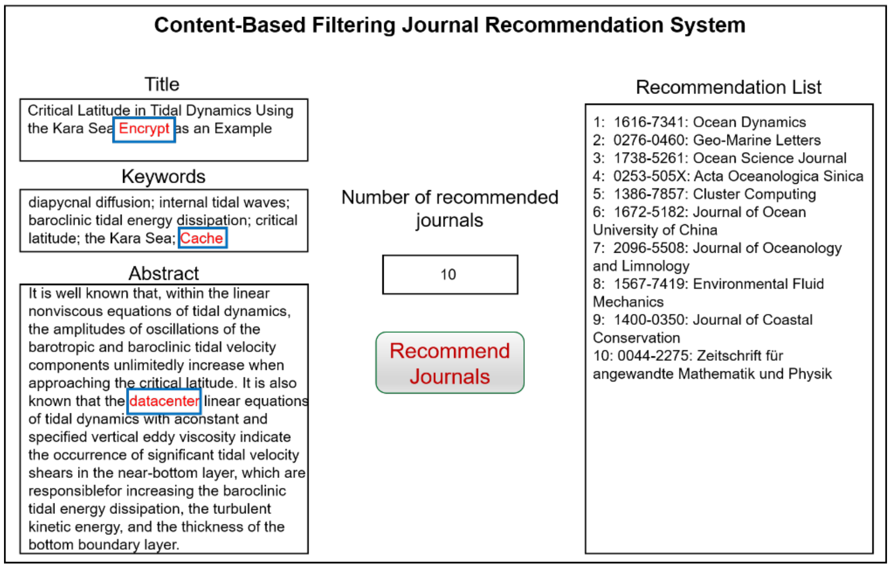
In this section, we introduce two kinds of adversarial attacks on JRS. The recommendation algorithms are developed on the basis of the Rocchio algorithm and the k-NN algorithm. We propose the adversarial attacks against the recommendation system.

5.1 Non-targeted adversarial attack

Non-targeted adversarial attack (NTAA) against the CBF-JRS assumes an adversary can modify the article information slightly such that the recommended results are quite different. For example, a competitor company of a journal recommendation system might want to make the system unreliable, or the JRS producer tries to verify the robustness of the system. This kind of attack might be adopted in the above scenarios.

Considering the NTAA problem, the intuitive idea is to insert several unrelated words in the article such that the recommendation lists do not contain the correct journal. For an inputted article: $Article = \{title, keywords, abstract\}$ and its corresponding journal label J_i. The system S would make correct recommendations satisfying $J_i \in R(Article)$. In order to construct an adversarial input $Article_{adv}$, we compute the term-weight table [Lan, Tan, Su et al. (2009)] of the TF-IDF values for the journal J_i and these words are sorted by the decreasing order of the weight in the journal. We denote these words as $\{word_1, word_2, \ldots, word_N\}$ and insert several words with low weight into the article information.

For example, we pick several words and insert them as the blue rectangles in Fig. 3. The correct journal does not appear in the recommended journals and more unrelated journals are recommended. This implies the inserted words could cause disruption to the recommendation system and lead to incorrect recommendations under this kind of adversarial attack.


Content-Based Filtering Journal Recommendation System

Title

Critical Latitude in Tidal Dynamics Using the Kara Sea Encrypt as an Example

Keywords

diapycnal diffusion; internal tidal waves; baroclinic tidal energy dissipation; critical latitude; the Kara Sea; Cache

Abstract

It is well known that, within the linear nonviscous equations of tidal dynamics, the amplitudes of oscillations of the barotropic and baroclinic tidal velocity components unlimitedly increase when approaching the critical latitude. It is also known that the datacenter linear equations of tidal dynamics with aconstant and specified vertical eddy viscosity indicate the occurrence of significant tidal velocity shears in the near-bottom layer, which are responsiblefor increasing the baroclinic tidal energy dissipation, the turbulent kinetic energy, and the thickness of the bottom boundary layer.

Number of recommended journals

10

Recommend Journals

Recommendation List

1: 1616-7341: Ocean Dynamics
2: 0276-0460: Geo-Marine Letters
3: 1738-5261: Ocean Science Journal
4: 0253-505X: Acta Oceanologica Sinica
5: 1386-7857: Cluster Computing
6: 1672-5182: Journal of Ocean University of China
7: 2096-5508: Journal of Oceanology and Limnology
8: 1567-7419: Environmental Fluid Mechanics
9: 1400-0350: Journal of Coastal Conservation
10: 0044-2275: Zeitschrift für angewandte Mathematik und Physik

Figure 3: An example of NTAA

Since the weight of each word reflect the connection between the word and the journal. There are two methods to conduct such adversarial attacks. One method is to remove the words with large weight from the article information, which could reduce the probability to classify the article to the journal. However, this might reduce the article readability. Hence, we adopt the other method by inserting several words with low weight, which also reduces the recommendation accuracy.

5.2 Targeted adversarial attack

Different from NTAA, the targeted adversarial attack (TAA) aims to cause the system to recommend some specific target journal, which is more complicated. This kind of attack may exist in the following scenario: a new journal J_{adv} wants to receive more submissions and be recommended more often by the journal recommendation system.

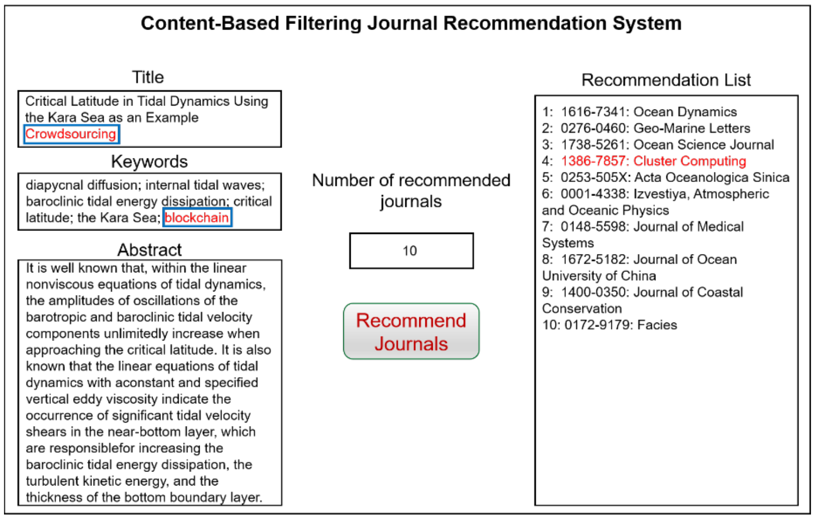
Considering the above scenario, the following steps might be conducted for the purpose. To begin with, the dataset adds the new journal information and several related articles. The system would train the recommendation algorithms and the new journal could be recommended correctly. Then, we construct a term-weight table in which the terms with high weights are more likely to be recommended. Finally, we propose the adversarial attack against the system by inserting the words with high weights.

For the more general TAA problem against the recommendation system, we compute the term-weight table of the specific journal J_{adv}. The table contains the following information: each term (word) that appears in the dataset, the IDF value, the value in a specific journal, and the weighted value of each term. The adversarial attack selects several words with the highest weighted value and inserts them into the input information. This attack method could completely change the recommendation results and the specific journal J_{adv} could be recommended more often by the system. This is because the system would recommend the journals according to the weighted value of the words. A word is considered as more related to the journal if the value is higher. Hence, an article with more related words could be recommended to the journal with higher probability by the system. The adversarial attack method selects such words to attack the system for the target goal.

For example, we select *Cluster Computing (ISSN: 1386-7857)* as the target journal. We first compute the term-weight table for the journal and some words are shown in Tab. 1. Considering the proposed example in Figs. 1 and 2, we insert two words ("blockchain" and "crowdsourcing") into the article information and the recommended journals in Fig. 4 contain the target journal *Cluster Computing* (labeled in red). This example shows that the system could be attacked by recommending a specific journal.

Table 1: An example of term-weight table for *Cluster Computing*

Term	IDF	Value in Journal	Weighted Value
blockchain	1.77	49.44	87.5088
datacenter	1.27	49.7	63.119
crowdsourcing	1.33	46.7	62.111
privacy	0.96	55.65	53.424
cache	0.94	54.87	51.5778
encrypt	1.11	44.5	49.395
encryption	1.23	33.3	40.959



Content-Based Filtering Journal Recommendation System

Title

Critical Latitude in Tidal Dynamics Using the Kara Sea as an Example Crowdsourcing

Keywords

diapycnal diffusion; internal tidal waves; baroclinic tidal energy dissipation; critical latitude; the Kara Sea; blockchain

Abstract

It is well known that, within the linear nonviscous equations of tidal dynamics, the amplitudes of oscillations of the barotropic and baroclinic tidal velocity components unlimitedly increase when approaching the critical latitude. It is also known that the linear equations of tidal dynamics with aconstant and specified vertical eddy viscosity indicate the occurrence of significant tidal velocity shears in the near-bottom layer, which are responsiblefor increasing the baroclinic tidal energy dissipation, the turbulent kinetic energy, and the thickness of the bottom boundary layer.

Number of recommended journals

10

Recommend Journals

Recommendation List

1: 1616-7341: Ocean Dynamics
2: 0276-0460: Geo-Marine Letters
3: 1738-5261: Ocean Science Journal
4: 1386-7857: Cluster Computing
5: 0253-505X: Acta Oceanologica Sinica
6: 0001-4338: Izvestiya, Atmospheric and Oceanic Physics
7: 0148-5598: Journal of Medical Systems
8: 1672-5182: Journal of Ocean University of China
9: 1400-0350: Journal of Coastal Conservation
10: 0172-9179: Facies

Figure 4: An example of TAA

6 Experimental results

In this section, we conduct extensive experiments to evaluate the performance of the proposed attacks and we show the experimental results.

6.1 Experiment setup

The journal dataset contains 583 journals from two famous publishers: Springer and Elsevier. There are about 150,000 articles that have already been published in these journals. The article information includes the title, the keywords, and the abstract of each article.

As described in Section 4.1, each article is represented by an n-dimensional vector and each dimension indicates the TF-IDF value of the corresponding word. We first remove the punctuation marks except for the short bar, then we remove the tense, singular and plural repeated words; we generated about 300,000 words from the 150,000 articles. We construct the stop-word library as follows. We assume a word is a stop-word if it appears in more than 570 journals of all 583 journals. In addition, we remove the words with low frequency. Specifically, words with term frequency below 3 are considered as unfamiliar words and they are removed from the list of words. Combining the preprocessing steps, we constructed 80,000 words from the dataset and each article can be represented by an 80,000-dimensional vector. We show some examples of the ID-Term-IDF dictionary in Tab. 2.

Table 2: An example of the ID-Term-IDF dictionary, decreasing by IDF

Id	Term	IDF
143	erythemal	2.0667
189	thundercloud	2.0667
284	electrets	2.0667
483	yamato	2.0667
486	irminger	2.0667
696	surfaceozone	2.0667
841	freons	2.0667
899	explosivity	2.0667
922	kunashir	2.0667

In order to analyze the accuracy of JRS, we divide the dataset into a training set and a test data with the ratio 9:1. Specifically, the articles are sorted according to the publication data; we use the 10% articles that are published recently as the test data and the other articles compose the training dataset.

We also evaluate the recommendation performance that the article is divided into a suitable category. The 583 journals in the dataset are divided into 10 different categories: Geography, Astronomy, Engineering Technology, Management Science, Chemistry, Environmental Science and Ecology, Biology, Mathematics, Physics, and Medicine. In order to improve the recommendation accuracy, we first recommend *m* categories for an article and then recommend k journals. In our experiments, we increase *m* from 1 to 5 and increase the number of recommended journals k from 10 to 30.

6.2 Recommendation accuracy

We evaluate the recommendation accuracy of the system. We first evaluate the performance when we only recommend the article to some categories. As shown in 5 where the x-axis represents the number of recommended categories *m* and the y-axis indicates the recommendation accuracy, the recommendation accuracy of both algorithms increase when *m* increases from 1 to 5. Since many articles are interdisciplinary, the accuracy is not very high when only one category is recommended. When *m=2*, the accuracy gets close to 90%.

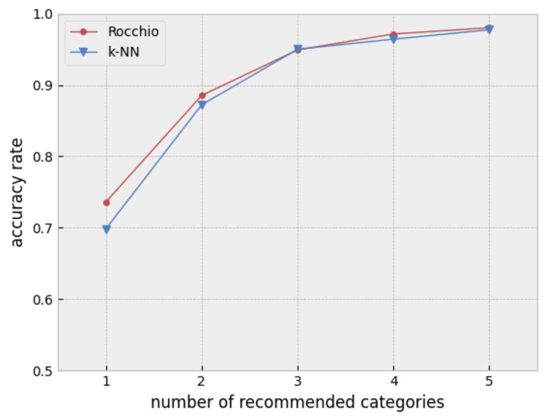


Figure 5: Categories recommendation accuracy

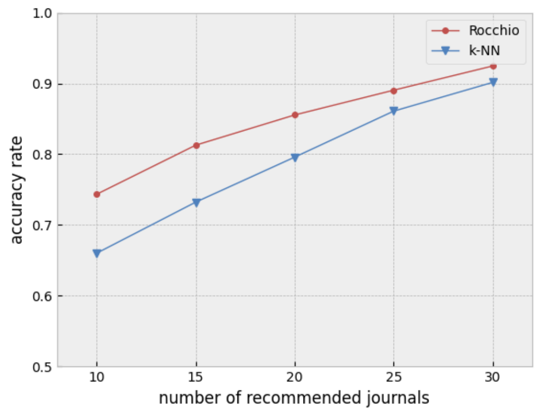


Figure 6: Journals recommendation accuracy

We evaluate the recommendation accuracy in Fig. 6 when k journals are recommended. When k increases from 10 to 30, the accuracy of both algorithms increase. When we recommend 30 journals (only 5.1% of the total number of all journals), the recommendation accuracy could be about 90%, which is acceptable for a practical journal recommendation system. Through our experiments, the other recommended candidate journals are quite similar to the correct journal. The experimental results show that both algorithms can achieve good performance. In the following experiments, we only show the attacking performance against the Rocchio algorithm since it works better than the k-NN algorithm.

6.3 Performance of non-targeted adversarial attacks

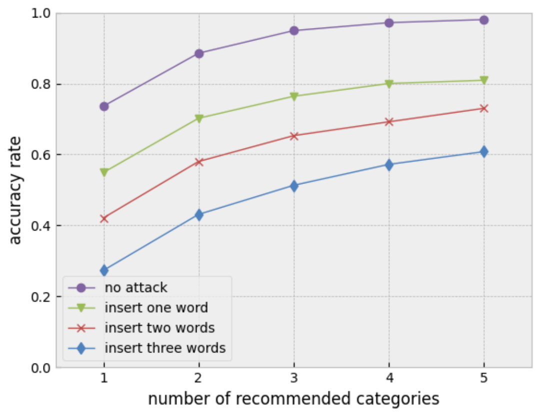


Figure 7: Categories recommendation accuracy under NTAA

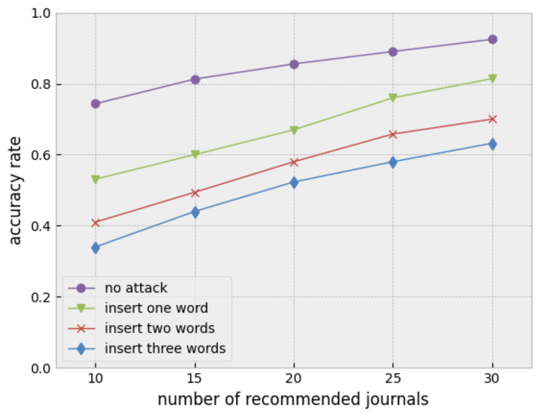


Figure 8: Journals recommendation accuracy under NTAA

We show the attack performance against the CBF-JRS when we insert different number of words into the article information. We show the attack performance for both category recommendation and journal recommendation. As shown in Fig. 7, the x-axis represents the number of recommended categories and the y-axis represents the accuracy. When we insert no, one, two and three items (words), the four curves depict the recommendation accuracy respectively. When more words are inserted, the recommendation accuracy could be reduced more largely. Specifically, the recommendation accuracy can be reduced by about 30%-50%. In addition, we evaluate the attack performance when *k* (increasing from 10 to 30) journals are recommended in Fig. 8. When more items (words) are inserted for the adversarial attack, the recommendation accuracy is largely reduced. For example, when 10 journals are recommended, the accuracy decreases from 74% to 33% when three words are inserted. These experimental results show that the proposed adversarial attack is effective against the CBF-JRS.

6.4 Performance of targeted adversarial attacks

We evaluate the TAA performance. When we assign a specific journal $\boldsymbol{J_{adv}}$ to each article, if journal $\boldsymbol{J_{adv}}$ appears in the recommendation list, we call it a successful target recommendation. We evaluate the successful ratio of the targeted recommendation. As shown in Fig. 9, the x-axis shows the number of recommended journals and the y-axis shows the targeted recommendation accuracy, inserting more words could increase the target recommendation accuracy. Specifically, by inserting three words, the target journal could be recommended with more than 80% accuracy when $k = 30$. The experimental results show that the TAA method could achieve significant attack performance against the CBF-JRS.

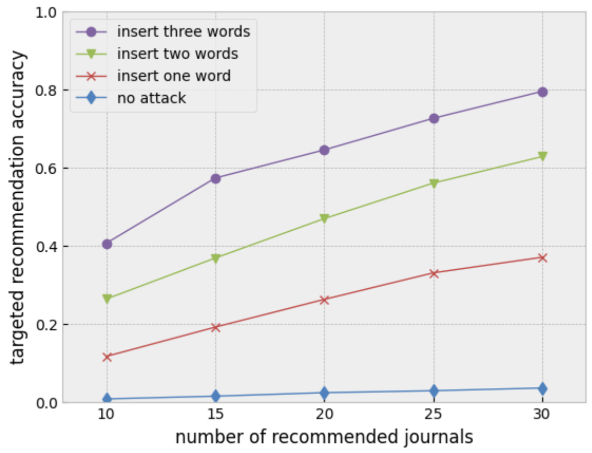


Figure 9: The journal recommend probability under TAA

Combining these results, the proposed CBF-JRS could achieve good recommendation performance. However, such systems are vulnerable to the adversarial attacks that may dramatically reduce the recommendation accuracy.

7 Conclusions

In this paper, we implement the CBF-JRS on the basis of two algorithms. The recommendation accuracy of the system could exceed 90%. However, such content-based filtering systems are vulnerable to the adversarial attacks. In this paper, we present two kinds of such attacks. The non-targeted adversarial attack can reduce the recommendation accuracy of the system, while the targeted adversarial attack can increase the probability of recommending some specific target journal. The experimental results also show the performance of the recommender system and the proposed attacks. We hope this work could draw the attention of the recommender systems and it is necessary to design robust recommendation algorithms against adversarial attacks.

Acknowledgement: We would like to thank anonymous reviewers and the associate editor for the constructive comments in improving this work.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grant Nos. U1636215, 61902082, the Guangdong Key R & D Program of China 2019B010136003, and Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019).

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

References

Bin, S.; Sun, G.; Cao, N.; Qiu, J.; Zheng, Z. et al. (2019): Collaborative filtering recommendation algorithm based on multi-relationship social network. *Computers, Materials & Continua*, vol. 60, no. 2, pp. 659-674.

Cao, M.; Zhou, S.; Gao, H. (2019): A recommendation approach based on product attribute reviews: improved collaborative filtering considering the sentiment polarity. *Intelligent Automation and Soft Computing*, vol. 25, no. 3, pp. 595-604.

Chakraborty, T.; Krishna, A.; Singh, M.; Ganguly, N.; Goyal, P. et al. (2016): FeRoSA: a faceted recommendation system for scientific articles. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, vol. 9652, pp. 528-541.

Christakopoulou, K.; Banerjee, A. (2018): Adversarial recommendation: attack of the learned fake users. https://arxiv.org/abs/1809.08336.

Gu, S.; Rigazio, L. (2014): Towards deep neural network architectures robust to adversarial examples. https://arxiv.org/abs/1412.5068.

Gu, Z.; Hu, W.; Zhang, C.; Wang, L.; Zhu, C. et al. (2020): Restricted region based iterative gradient method for non-targeted attack. *IEEE Access*, vol. 8, pp. 25262-25271.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. (2004): Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53.

Huang, X.; Du, X. (2014): Achieving big data privacy via hybrid cloud. *IEEE Conference on Computer Communications Workshops*.

Jiang, S.; Pang, G.; Wu, M.; Kuang, L. (2012): An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, vol. 39, no. 1, pp. 1503-1509.

Lam, S. K.; Riedl, J. (2004): Shilling recommender systems for fun and profit. *Proceedings of the 13th International Conference on World Wide Web*, pp. 393-402.

Lan, M.; Tan, C. L.; Su, J.; Lu, Y. (2009): Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721-735.

Massa, P.; Avesani, P. (2004): Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems*, vol. 3290, pp. 492-508.

Miao, Y. Q.; Kamel, M. (2011): Pairwise optimized Rocchio algorithm for text categorization. *Pattern Recognition Letters,* vol. 32, pp. 375-382.

O'Mahony, M. P.; Hurley, N. J.; Silvestre, G. C. M. (2005): Recommender systems: attack types and strategies. *Twentieth National Conference on Artificial Intelligence & the Seventeenth Innovative Applications of Artificial Intelligence Conference*, pp. 334-339.

Park, D. H.; Kim, H. K.; Choi, I. Y.; Kim, J. K. (2012): A review and classification of recommender systems research. *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059-10072.

Sha, X.; Sun, Z.; Zhang, J. (2019): Attentive knowledge graph embedding for personalized recommendation. https://arxiv.org/abs/1910.08288v1.

Tian, Z.; Li, M.; Qiu, M.; Sun, Y.; Su, S. (2019): Block-DEF: a secure digital evidence framework using blockchain. *Information Sciences*, vol. 491, pp. 151-165.

Tian, Z.; Luo, C.; Qiu, J.; Tian, Z.; Guizani, M. (2019): A distributed deep learning system for web attack detection on edge devices. *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963-1971.

Wang, Q.; Li, W.; Zhang, X.; Lu, S. (2016): Academic paper recommendation based on community detection in citation-collaboration networks. *Web Technologies and Applications*, vol. 9932, pp. 124-136.

Wang, Z.; Liu, Y.; Yang, J.; Zheng, Z.; Wu, K. (2015): A personalization-oriented academic literature recommendation method. *Data Science Journal*, vol. 14, no. 17, pp. 1-9.

Xiao, L.; Li, Y.; Huang, X.; Du, X. (2017): Cloud-based malware detection game for mobile devices with offloading. *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2742-2750.

Yi, B.; Shen, X.; Liu, H.; Zhang, Z.; Zhang, W. et al. (2019): Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4591-4601.

Yin, C. Y.; Shi, L. F.; Sun, R. X; Wang, J. (2019): Improved collaborative filtering recommendation algorithm based on differential privacy protection. *Journal of Supercomputing*. https://doi.org/10.1007/s11227-019-02751-7.

Zhao, W.; Wu, R.; Liu, H. (2016): Paper recommendation based on the knowledge gap between a researcher's background knowledge and research target. *Information Processing & Management*, vol. 52, no. 5, pp. 976-988.