

Methodology for Product Recommendation Based on User-System Interaction Data: A Case Study on Computer Systems E-Commerce Web Site

Tahir Enes Adak[1], Yunus Sahin[1], Mounes Zaval[2(✉)], and Mehmet S. Aktas[2]

[1] Research and Development Center, Casper Computer Systems, Istanbul, Turkey
{TahirEnes.Adak,Yunus.Sahin}@casper.com.tr
[2] Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey
mounes.zaval@std.yildiz.edu.tr, aktas@yildiz.edu.tr

Abstract. Within the scope of this study, we developed a product recommendation methodology for customers by analyzing shopping behaviors based on user-system interaction data collected on Casper Computer Systems' website. To achieve the "right product to the right customer" objective, we predict customer interests using a collaborative filtering algorithm on collected data from previous customer activities. In turn, this minimizes prediction errors and enables better-personalized suggestions of computer system configuration. We took advantage of the implicit feedback approach while modeling customer behaviors if they liked or disliked a particular product. After customer behavior data is collected, we form the customer-product matrix and generate personalized product suggestions for each customer with the help of user-item-based collaborating filtering and item-item-based collaborating filtering algorithms. Customer-website interaction is considered a key input variable in creating personalized recommendations. Customers are supposed to use the website and leave interaction data regarding product configurations they're interested in. To prove the efficiency of this methodology, we developed a prototype application. The product suggestion success rate of the application is tested on datasets generated from log data of the Casper website. Performance results prove that the developed methodology is successful.

Keywords: E-Commerce · Machine learning · Collaborative filtering · Recommendation systems

1 Introduction

E-commerce is one of the essential fields today, due to the noticeable increase in the emergence of E-commerce websites, especially during the Covid-19 pandemic, which prompted people to stay at home in order to preserve their lives from danger, so people were forced to buy their needs from the Internet [1,2]. Due

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13377, pp. 35–46, 2022.
https://doi.org/10.1007/978-3-031-10536-4_3

to this increase, recommendation systems are an important factor in attracting customers because they try to understand the needs and desires of the customer to suggest the most suitable products specifically for this customer based on the history of interaction between the system and the customer. In the field of the computer industry, there is a tremendous number of computer configurations that make unknowledgeable customers get lost among all these configurations, which leads to the purchase of a product that does not meet their needs, causing a bad experience for the customer. For this reason, companies have had to come up with a reasonable solution to overcome this problem. Casper company has the Easy builder tool _, which consists of simple, non-technical questions so the user can answer them. At the end of this process, this tool suggests the most suitable computer configuration for this user based on his answers. The resulted computer configuration is then to be considered the user's preferences.

The vast majority of technology customers are defined as everyday users who do not have expertise in features or do not want to take the time to research. In this case, there is a possibility that the purchased products are not suitable for their needs. Product vendors present products in different segments and cater to general customer profiles. These profiles group people with similar expectations but do not capture the accuracy of personalized product customization. There are risks such as product dissatisfaction and product return.

Motivation: The KUR/AL project (Easy Builder tool) brings together the products most suitable for the customers' needs by making personalized recommendations. Thus, users get products with the right features at a more affordable price, maximizing product and shopping satisfaction.

We introduce the needs that determine the goal of completing this project as follows:

- The need for decision support systems that will reduce Casper's return risk and increase customer satisfaction.
- The need for the customer to be able to reach the product they need much more easily and with a point shot.
- The need for software that will bring customers to Casper brand and that we can offer the right product to the customer.
- The need for decision support systems that will indirectly increase Casper brand perception and increase Casper's customer satisfaction.
- The need for software in which the customer can define the product he needs even if he does not know the detailed features of the computer.
- The need to provide the customer with the ability to easily access the most suitable computer for their needs with simple questions and answers.

Research Problem: This study proposes a novel methodology for collecting users' preferences from implicit feedback through their interaction with the E-commerce website and applies collaborative filtering algorithms to provide recommendations for active users. Therefore, we are investigating the answer to the following questions:

1. How can we model such interaction data to capture user preferences in computer configuration?
2. After modeling the data, how can we apply recommendation algorithms?
3. What would be the evaluation results of the applied methodology, and how useful is it?

Contribution: Within the scope of this study, we introduce a novel software architecture that produces recommendations by a method based on the customer's tastes, identifying customers with similar tastes and suggesting different product tastes of the identified customers. The developed software offers the optimum product in different configurations according to customers' price and usage preferences. A prototype of this software is implemented and evaluated so we can measure how successful such software could be. The results of the evaluation were promising.

Organization of the Paper: After we have introduced our motivations and the research problem that circulates this research, we move on to present the fundamental concepts and similar works in the literature. Then we explain the methodology that guided the process of implementing this project. Furthermore, we report the used technologies and the evaluation results for the recommendation system software architecture.

2 Fundamental Concepts and Literature Review

2.1 Fundamental Concepts

Collaborative Filtering adopts a methodology that aims to consider the point of view of other users of similar behavior when trying to guess whether an item will appeal to the desired user [3]. This way, CF draws attention to other users' opinions and brings new factors to the recommendation process other than depending solely on the user profile. Let us assume a website with two frequent users that contains three items: item_X, item_Y, and item_Z; since we only have two users, then they are the most similar users of the website; If the first user likes all three items, and the second user only liked item_X and item_Y, then we can predict item_Z being desired by the second user as well based on the first user opinion [4]. Designing a successful collaborative filtering system needs an efficient algorithm that can overcome environmental challenges, including:

- **Shilling attacks.** The vulnerability that appears through creating an enormous number of fake profiles in the system anonymously with the purpose of targetting a particular item to increase its popularity [5–7].
- **Synonymy.** Usually, the recommendation system cannot determine the exact similarity of two similar items but with different names in the system. This condition imposes challenges and difficulties in generating item recommendations for the users [8].

- **Cold Start.** When new items or users are added to the system with no prior engagements or ratings, the recommendation system usually cannot predict ratings for such new products. Consequently, it cannot generate recommendations for such new users [8–10].
- **Others.** like Data Sparity, Scalability, etc. [8,11–15].

Choosing the most suitable similarity metric plays a crucial role for the system to succeed; each metric follows a unique formula that deals with the rating matrix differently, so the similarity results will be different. There are many similarity metrics in the literature, such as Tanimoto similarity metric, Pearson correlation similarity metric, and many others [12,16].

The rating matrix represents the user-system interaction information extracted from analyzing e-commerce websites' log files. It could be constructed in two different ways; either the rows represent users and columns represent products/items or the other way around. Hence, two branches of collaborative filtering algorithms emerge based on how we allocate users and items. Both **User-Based CF** and **Item-Based CF** follow the same approach for calculations. Still, UBCF performs the calculation on a user-item rating matrix, and IBCF performs it on an item-user rating matrix which is the transpose matrix of the user-item rating matrix. Therefore UBCF is concerned with calculating similarities between users in contrast to IBCF, which focuses on the similarities between items [16]. Tables 1 and 2 demonstrate the difference between the explained two formations of the rating matrix. On the left-hand side, we have our item-user matrix _computer configurations being the item in our case_, and on the right-hand side, we have the user-item matrix.

Where: CC → Computer Configuration and $x_{i,j}$ → Rating of $item_j$ by $user_i$. Figure 1 concludes the collaborative algorithm approach procedure. So, for a random user U_a, we assign a predicted rating score for each item in the item vector that this specific user did not rate or engage with directly, then we rank this

Table 1. Item-user rating matrix

CCs\Users	user_1	user_2	user_3
CC_1	$x_{1,1}$	$x_{2,1}$	
CC_2		$x_{2,2}$	$x_{3,2}$
CC_3	$x_{1,3}$	$x_{2,3}$	

Table 2. User-item rating matrix

Users\CCs	CC_1	CC_2	CC_3
user_1	$x_{1,1}$		$x_{1,3}$
user_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$
user_3		$x_{3,2}$	

vector's elements based on the predicted score. Therefore, we obtain a finalized list of top-N recommendations.

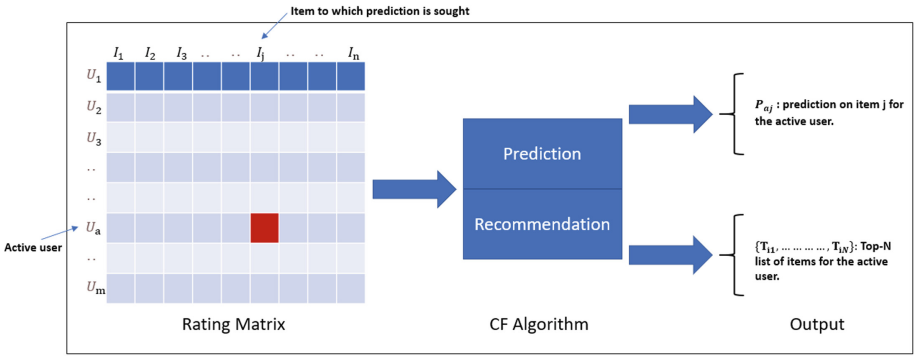

Fig. 1. CF process [17]

2.2 Literature Review

Within the scope of the research, the transactions made by the customers using the Easy Builder application on the Casper site will be monitored, and product suggestions that can be made to the customers will be determined. In the suitable computer configuration mapping to the right customer, the collaborative filtering method will predict what the customers might be interested in based on the customer activity data in the past. This way, it will be possible to offer customers a computer configuration that will match their needs and expectations. Here, the uses of the collaborative filtering method in the literature are examined, and the most successful use of this method in the research is studied [18–24]. This study assumes that the customers have previously used the Easy Builder application at a sufficient level and left information about the configurations they are interested in to make configuration suggestions to the customers. There exist studies that focus on analyzing user's browsing graph data to extract knowledge [25–28]. However, in this study, we focus on user's purchase behaviour on the computer systems e-commerce website.

3 Methodology

Figure 2 illustrates the pipeline of the applied methodology for building a recommendation system for Casper e-commerce website. This system provides recommendations for those users who completed the questions series of the easy builder tool sufficiently; then, we capture their preferences based on the easy builder tool's results. We then construct a preferences vector that concludes user's interactions in the system, and from that, we construct the rating matrix

that would be provided to the collaborative filtering algorithms. At the end of this process, we obtain a recommendation engine that captures the users' interests, predicts what might appeal to them, and then recommends the most suitable product to deliver the right product to the right customer.

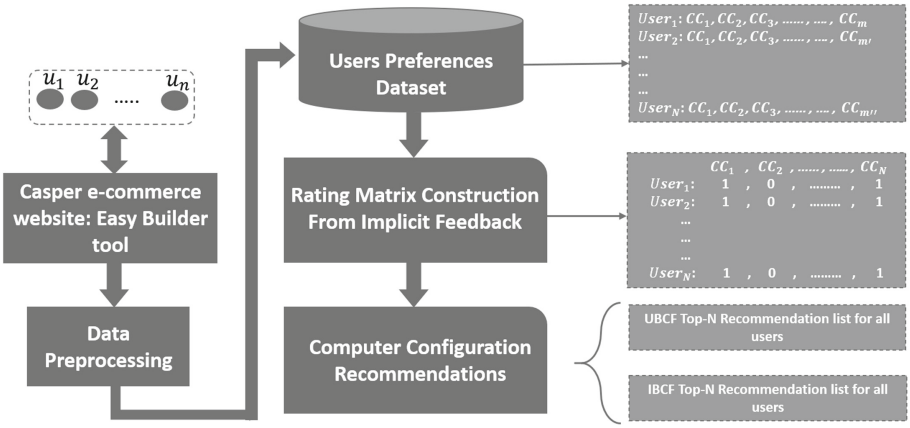

Fig. 2. The proposed software architecture for computer configurations recommendation system

Capturing Users Preferences: Casper's e-commerce website has an overwhelming amount of computer configurations, Which makes eliciting users' preferences difficult. Additionally, regular customers lack the adequate education to build or select their computer, so when they try to do so, there is a high possibility that the purchased product is not what they wanted at first. Casper company introduces the easy builder tool (KUR/AL) as a way to overcome this challenging problem preventing users' dissatisfaction; besides, it captures users' interests. The easy builder tool is a series of multi-choice, easy-to-answer questions related to the user's purpose of using the computer (education, work, daily usage, gaming), desired platform (laptop, desktop), and budget. The system suggests the most suited computer configurations based on the user's answers by completing these questions. Since this is the case, we consider that this user is interested in the final product, which satisfies the user's needs. The more the user interacts with this tool, the more we can understand their needs and interests through analyzing these user-system interaction data later.

Data Pre-processing and Preparing: The outputs and results of the easy builder tool are collected and stored in the websites' log files, so we need to capture this information and convert it into a format that satisfies our goal. Besides, websites' log files have noisy and dirty raw data that needs to be cleaned in order to be useful for data analysis tasks. Subsequently, for each active user

of the websites that used the easy builder tool, we make an array that includes all the easy builder tool's computer configuration results and consider that these configurations represent this user's preferences.

Rating Matrix Construction from Implicit Feedback: Based on the easy builder tool results, we constructed a vector array that captures users' preferences for each user. However, we should convert this into numbers to have the rating matrix that can be fed to UBCF and IBCF algorithms. Since the user is not providing any explicit rating for the computer configuration, we follow the implicit feedback methodology by calculating the frequency of a particular product presented to the users at the end of the Q&As series and the user has clicked through the resulted product; then for each user, we normalize these frequencies bringing back the values into the closed interval [1,5]. Hence, We define our rating matrix $Y \in R^{U \times I}$ _Where: U, I $\rightarrow$ the size of the users, items sets, respectively_ as:

$$y_{ui} = \begin{cases} [1,5] \text{ Normalized CC's View number in the easy builder results;} \\ 0, \qquad\qquad \text{Otherwise.} \end{cases} \tag{1}$$

Table 3 reveals for us an example of the rating matrix, which will be given to the user-based and item-based collaborative filtering algorithms. The output of this stage is a very similar matrix, with the columns representing the set of all computer configurations and the rows representing the set or the opposite.

Table 3. Rating matrix

Users\CCs	CC_1	CC_2	...	CC_I
$user_1$	1	0	...	3
$user_2$	4	2	...	5
...	...	...	...	...
...	...	...	...	...
$user_U$	3	2	...	0

Applying User-Based and Item-Based Collaborative Filtering Algorithms: After processing and converting the data into the proper format, we can apply both user-based and item-based collaborative filtering algorithms based on the rating matrix that we obtained in the previous section. Pursuing the success of our recommendation system, we followed a simple pipeline to get the best results:

– **Calculating Similarity:** the Log Likelihood similarity metric is applied to calculate both users' and items' similarities. **Log Likelihood similarity** is a

statistical test that is independent of the explicit preferences values. Its value expresses how unlikely it is for two customers to have the same interests [12,29].
- **Neighborhood Selection:** we take into account only the most 15 similar users/items that contribute to the process of generating recommendations for computer configurations.
- **UB/IB Recommender:** we provide the calculated similarity and neighborhood information to the generic user/item-based recommender.

4 Prototype and Evaluation

4.1 Prototype and Evaluation Strategy

The prototype implementation of this research was carried out using C# and .Net framework to provide the web API services of this project. For integrating user-based and item-based collaborative filtering algorithms into this project, the NReco library has been utilized. NReco Recommender [30] is .Net implementation of the non-hadoop version of Apache Mahout collaborative filtering engine. Naming the classes and public methods is the same as in Apache Mahout but aligned with C# naming conventions. And with the help of user-based and item-based recommender classes, we implemented our approach on top of the dataset explained in the next section. We chose the root mean squares error as our metric for evaluating the system, and we report the results in the experimental results section. The goal of the developed prototype is to bring the mentioned software architecture in the methodology section to real life so we can observe and test its usefulness and efficiency.

4.2 Dataset

This study's dataset is created by obtaining the user's unique identity, product's unique identity, and viewing rate over the data of the products visited by the users on www.casper.com.tr. On this website, Casper provides corporate services and makes sales. It is stored in CSV and delivered to the collaborative filtering algorithms. The dataset comprises three fields (UserId, AssetId, and Viewing Rate normalized between 1 and 5) with approximately 100K records, as shown below. The dataset was split into two parts for the sake of testing by 80% for training and 20% for testing, which corresponds to around 80,000 and 20,000 records, respectively (Fig. 3).

```
514,163,3
514,172,2
514,175,5
514,176,5
514,180,3
514,190,3
514,194,4
```

Fig. 3. Sample dataset

4.3 Test Design

For the purpose of understanding how successful the developed system is based on the applied methodology explained in this paper, we utilized the root mean squares error (RMSE); RMSE is a statistical metric that is used to evaluate machine learning models by calculating the difference between the model's predictions values and the actual observations values, using the formula:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}\left(o_i - \hat{e_i}\right)^2} \tag{2}$$

where:

- N → Number of instances.
- o_i → Observation preference value.
- $\hat{e_i}$ → Estimated preference value.

RMSE value is interpreted in terms of accuracy so that the smaller its value is, the better the model is performing. Therefore, we seek to accomplish this work by obtaining as low RMSE value as possible [31,32].

4.4 Experimental Results

Table 4 reports the evaluation results of our prototype implementation. These results are positive and promising, prompting us to investigate the designed methodology further and combine more than two collaborative filtering algorithms to increase Casper company's customers' satisfaction.

Table 4. RMSE evaluation results

Algorithm	Item-based CF	User-based CF
RMSE	0.98	1.03

We again observe that the item-based collaborative filtering algorithm performed better than the user-based collaborative filtering algorithm in real-world cases. Even though these two algorithms are theoretically the same, the item-based CF algorithm proves its superiority over the user-based CF algorithm.

5 Conclusion and Future Work

This research introduces a methodology for product recommendation based on user-system interaction data on computer systems E-Commerce Web Site accompanied by an appropriate software architecture that illustrates the necessary stages of implementing such methodology in real life. We have also implemented and tested a prototype with the help of NReco framework to ensure the positive outcomes of this project. The results are promising, with 0.98 and 1.03 RMSE scores for item-based CF recommendation engine and user-based CF recommendation engine, respectively. In future work, we plan on integrating multiple recommendation algorithms into this system, including deep learning algorithms. Also, we will try running these algorithms on more extensive and extended datasets.

Acknowledgement. Thanks to Casper Computer Systems company for supporting this study by providing all the necessary requirements and datasets.

References

1. Bhatti, A., et al.: E-commerce trends during covid-19 pandemic. Int. J. Future Gener. Commun. Netw. **13**(2), 1449–1452 (2020)
2. Hanson, W.A., Kalyanam, K.: Internet Marketing and E-commerce. (Student ed.). Thomson/South-Western (2020)
3. Goldberg, K., et al.: Eigentaste: a constant time collaborative filtering algorithm. Inf. Retrieval **4**, 133–151 (2001)
4. Miller, B., Konstan, J., Riedl, J.: Pocketlens: toward a personal recommender system. ACM Trans. Off. Inf. Syst. **22**(3), 437–476 (2004). Copyright: Copyright 2012 Elsevier B.V., All rights reserved
5. Gunes, I., Kaleli, C., Bilge, A., Polat, H.: Shilling attacks against recommender systems: a comprehensive survey. Artif. Intell. Rev. **42**(4), 767–799 (2012). https://doi.org/10.1007/s10462-012-9364-9
6. Kaur, P., Goel, S.: Shilling attack models in recommender system. In: 2016 International Conference on Inventive Computation Technologies (ICICT), vol. 2, pp. 1–5. IEEE (2016)
7. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: Proceedings of the 13th International Conference on World Wide Web, pp. 393–402 (2004)
8. Pandey, A.K., Rajpoot, D.S.: Resolving cold start problem in recommendation system using demographic approach. In: 2016 International Conference on Signal Processing and Communication (ICSC), pp. 213–218. IEEE (2016)
9. Li, X.: Collaborative filtering recommendation algorithm based on cluster. In: Proceedings of 2011 International Conference on Computer Science and Network Technology, vol. 4, pp. 2682–2685. IEEE (2011)
10. Jindal, H., Singh, S.K.: A hybrid recommendation system for coldstart problem using online commercial dataset. Int. J. Comp. Eng. Appl. **7**(1), 100 (2014)
11. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**, 76–80 (2003)

12. Bagchi, S.: Performance and quality assessment of similarity measures in collaborative filtering using mahout. Procedia Comp. Sci. **50**, 229–234 (2015). Big Data, Cloud and Computing Challenges
13. Claypool, M., et al.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR 1999 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, California. ACM (1999)
14. Su, X., Khoshgoftaar, T.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 2–7 (2009)
15. Sarwar, B.M., et al.: Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In: Proceedings of the Fifth International Conference on Computer and Information Technology, vol. 1. Citeseer, pp. 291–324 (2002)
16. Papagelis, M., Plexousakis, D.: Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. Eng. Appl. Artif. Intell. **18**(7), 781–789 (2005)
17. Bokde, D.K., et al.: Role of matrix factorization model in collaborative filtering algorithm: a survey. ArXiv abs/1503.07475 (2015)
18. Duzen, Z., Aktas, M.S.: An approach to hybrid personalized recommender systems. In: 2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–8. IEEE (2016)
19. Arpacı, A., Aktaş, M.: Investigation of different approaches for recommendation system. In: ELECO 2018 (2018)
20. Tas, K., et al.: On the implicit feedback based data modeling approaches for recommendation systems. In: 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp. 1–6. IEEE (2021)
21. Uzun-Per, M., et al.: An approach to recommendation systems using scalable association mining algorithms on big data processing platforms: a case study in airline industry. In: 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–6. IEEE (2021)
22. Uzun-Per, M., et al.: Scalable recommendation systems based on finding similar items and sequences. Concurrency Comput. Pract. Exp. e6841 (2022). https://doi.org/10.1002/cpe.6841
23. Uzun-Per, M., et al.: Big data testing framework for recommendation systems in e-science and e-commerce domains. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 2353–2361. IEEE (2021)
24. Aktas, M.S., et al.: A web based conversational case-based recommender system for ontology aided metadata discovery. In: Fifth IEEE/ACM International Workshop on Grid Computing, pp. 69–75. IEEE (2004)
25. Olmezogullari, E., Aktas, M.: Representation of click-stream datasequences for learning user navigational behavior by using embeddings. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 3173–3179. IEEE (2020)
26. Uygun, Y., et al.: On the large-scale graph data processing for user interface testing in big data science projects. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 3173–3179. IEEE (2020)
27. Oz, M., et al.: On the use of generative deep learning approaches for generating hidden test scripts. Int. J. Softw. Eng. Knowl. Eng. **31**(10), 1447–1468 (2021)
28. Olmezogullari, E., Aktas, M.: Pattern2Vec: representation of clickstream data sequences for learning user navigational behavior. Concurrency Comput. Pract. Exp. **34**(9), e6546 (2022)

29. Vivek, M.B., Manju, N., Vijay, M.B.: Machine learning based food recipe recommendation system. In: Guru, D.S., Vasudev, T., Chethan, H.K., Sharath Kumar, Y.H. (eds.) Proceedings of International Conference on Cognition and Recognition. LNNS, vol. 14, pp. 11–19. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5146-3_2
30. NReco Recommender: Official library website. https://www.nrecosite.com/recommender_net.aspx
31. Hananto, A.L., Sulaiman, S., Widiyanto, S., Rahman, A.Y.: Evaluation comparison of wave amount measurement results in brass-plated tire steel cord using RMSE and cosine similarity. Indones. J. Electr. Eng. Comput. Sci **22**(1), 207 (2021)
32. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)?-Arguments against avoiding RMSE in the literature. Geoscientific Model Develop. **7**(3), 1247–1250 (2014)