

# A hybrid big data movies recommendation model based k-nearest neighbors and matrix factorization

Abderrahmane Ez-Zahout<sup>1,2</sup>, Hicham Gueddah<sup>2</sup>, Abir Nasry<sup>1,2</sup>, Rabie Madani<sup>1,2</sup>, Fouzia Omary<sup>1,2</sup>

<sup>1</sup>Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco

<sup>2</sup>Department of Computer Sciences, Higher Normal School, Mohammed V University, Rabat, Morocco

## Article Info

### Article history:

Received Oct 25, 2021

Revised Jan 11, 2022

Accepted Jan 29, 2022

### Keywords:

Collaborative filtering

K-nearest neighbors algorithm

Matrix factorization

Recommender system

Singular value decomposition

## ABSTRACT

On the subject of broadcasting the information, finding someone's favorite book or movie in a sea of data containing books and movies has become a crucial issue. In an era when there are so many genres and types of movies and books, the customer may find it difficult to choose which to discover in the first place. Thus, personalized recommendation systems play an important role because of the value that is attributed to movies and books nowadays, and considering that there are so many to choose from that the user may not be able to have a specific target. In this context, our proposed work, design and implement a prototype of movie recommendation system while taking into consideration the real requirement for the search of movies and books. The research of movie recommendation system by using the k-nearest neighbors approach and collaborative filtering algorithm are adopted to extract the criteria for a good use case on recommender systems. At last, the results are as what was expected as they showed that the system has a good recommendation effect.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Abderrahmane Ez-Zahout

Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University

Street of Michlifén, Rabat, Morocco

Email: a.ezzahout@um5r.ac.ma

## 1. INTRODUCTION

In big data analytics, a recommender system provides the ability to extract relevant, new and desirable contents. Practically, in big data analytic engines such as in spark, SparkSQL core and SparkMILib [1] especially which gives us the ability to apply distributed collaborative filtering. As intelligent system, it provide the ability to resolve many problems related to our topic. Practically, in the stage of advanced big data analytic processing, we have to call and use some advanced and depth functions in machine learning and also related to graphs analysis in order process and extract relevant information for recommender systems. In the context of decisional system, those uses permit to speed up processing and take decisions with high score of precision. This famous cores are used to learn the probability distribution over a ranking matrix and by using layers in neural network architecture, this application can facilitate computing the similarity between different movies [2]. Recommender systems can be considered as a kind of tool, for which a user discloses his profile (represented in a general way by his preferences and tastes). In return, they project him a personalized and reduced image of a whole mass of information that exceeds our cognitive faculties [3].

At first sight, they resemble information search engines [4], moreover they can content to return a list of results ordered according to an element of the user's profile which in these systems is reduced quite simply to the keywords of his request. Practically, the returned results for the same list of keywords are identical as

long as the query is the same. In the absence of recommendation systems, a user in search of information on the web faced with this tide of information has recourse to these search engines among which are: Google, Yahoo! and Altavista. The nature of the information manipulated by these search engines is textual in multiple formats, for example the hypertext markup language (HTML) format of web pages. The field of application of recommendation systems can cover a very vast space of items such as: films, music, images commercial products. Moreover, they are more and more an integral part of e-commerce sites.

Recommendation systems have become so important that they can be found in every field. They have become crucial in finding information related to the user's preferences as their goal is to filter and adapt information based on each individual's liking. There are different methods used for having a good recommendation system: first one is to recommend basing on the content [5] and the second one is based on the similarity [6] of the user with other users which is called collaborative approaches [7].

## 2. RELATED WORK

Generally, the techniques for the recommender systems are divided into three categories: collaborative filtering, content-based filtering, and hybrid filtering. The content-based recommender system: this method is based on the data that is given by the user, either explicitly (for example: rating) or implicitly (for example: clicking on a link). After capturing this data, the system generates a profile for each user called "user's profile" that is used to generate the recommendation for that specific user. In this method, the system analyzes the user's past activity and attempts to recommend a similar item based on that history. It takes one user's profile at a time in order to be as accurate as possible. The collaborative filtering [8]-[10]: in the contrary of the past method, this filtering takes a bunch of user's profiles to analyze at the same time. It finds users whose taste are similar to those of a given user, then recommends items and products based on the assumption that they will fit to the given user's taste because of the similarities with the other users.

The last method, which is a hybrid method of the two cited above emerged in order to combine the strengths of them both as well as to overcome their weaknesses. It can be used in two different ways: the first one is that we first use the content-based filtering then pass the results to the collaborative recommender, and the other way is to start with the collaborative recommender then pass the results to the content-based filtering. It goes both ways. The last way is to use one single model, which is a combination of the two filters to generate the result. These modifications are also used to deal with other data problems such as data sparsity or scalability issues. Figure 1 describes the "taxonomy of the recommender system".

### 2.1. Organization

The recommender system faces many challenges. Some of these problems include cold start problems, data disparity and scalability [11], [12]. Let's review these problems in order to understand them and come up with a suitable solution for each of them:

#### – Cold start problem

Refers to the number of users needed in order to find a match. We need enough users in order to have reliable results. At the beginning, each user profile is empty since they haven't been able to review or evaluate anything yet, so the system does not have an idea about their taste. Therefore, recommendation on an item becomes difficult. Items, as well as users, go through the same problem at the beginning. The new item has not been reviewed by users since they are new to them. The solution to these two problems is to implement hybrid techniques. The Figure 1 gives a taxonomic exists classes in the literature of recommender systems.

#### – Data disparity

Finding same users who rate the same item can be a challenging task because most users do not rate items. When we lack information about a user, giving recommendation can be laborious.

#### – Scalability

With great data, comes great resources. Using the use of massive amount of data can help collaborative filtering to improve reliability. However, when the information grows exponentially, so does the price and the result becomes inaccurate.

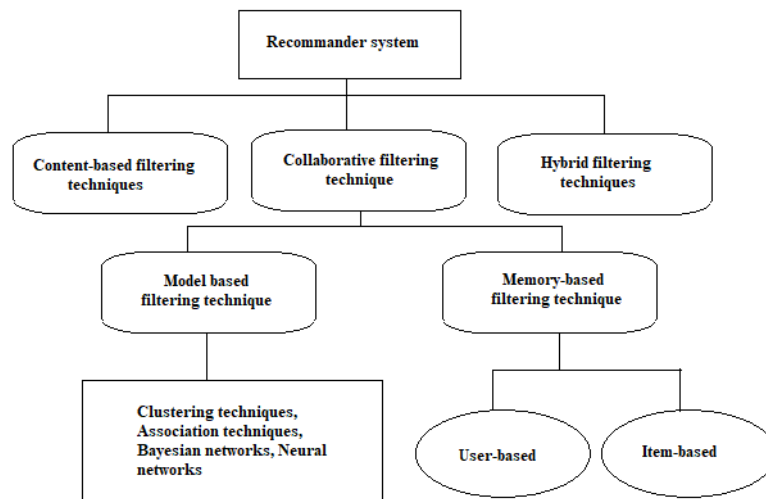


Figure 1. Taxonomy of recommendation system

### 3. PROPOSED RESEARCH METHOD

The most used technique for recommendation systems in both academic and commercial fields is the collaborative filtering (CF). The idea behind this great technique (Figure 1) is to make the recommendations based on the user's ratings of a product. There are two types of rating: explicit or implicit. The first type is done by directly asking the user to express their opinion about a product, while the second one is understood after the purchase or the mention of an item in an article as it is considered an expression of appreciation. This second method is easier to collect. However, it implies adding some noises to the information that have been collected [13], [14].

#### 3.1. KNN algorithm

In the recommendation world, one of the most appreciated techniques is the k-nearest neighbor (KNN) [15]-[18]. More often than not, techniques of this kind are based Netflix [19] being the most known streaming services nowadays, made the matrix factorization approach popular as it is used by them to handle massive amount of data while staying meticulous and nimble. By containing algorithms that are easy to implement and quite efficient, these techniques are observed as being the state of the art for a static ranking forecasting duty.

Matrix Factorization uses vectors of factors contained from item evaluation models to designate users and items, then bases the recommendation on the match between factors of both items and users. The input data used for recommendation systems is of different types and categories placed in a matrix representing users in one dimension and items of interest in the other dimension. In order to identify latent semantic factors, singular value decomposition (SVD) approach is used in information retrieval, while it is crucial to factorize the user-element evaluation matrix in the collaborative filtering situation. These matrices often face lack of values and are considered incomplete, which leads to sparsity. One must be careful in handling such matrices, otherwise overlearning can be induced.

In this work, we will be suggesting a recommendation system for movies based on that second approach, collaborative filtering, in addition to KNN which computes the similarity between movies as a means to generate movie recommendation for a specific user, and we determine which are the Top-K. On either binary data or real-valued data, benefit some principles of rules of association and generalize them on object comparison. Although more appreciated in item-to-item recommendations, these techniques experience lack of scalability so the time consumed by the algorithms increases noticeably with the number of items. The three components most present in collaborative filtering approaches using KNNs are the following: measure of similarity, function that uses the measure of similarity to capture the neighborhood and prediction function build on the evaluations of neighbors.

### 3.2. Matrix factorization

Recommender systems can be defined in several ways, given the diversity of classifications proposed by these systems [2], but there is a general definition by Burke [12] who defines them as follows “systems capable of providing personalized recommendations to guide the user to interesting and useful resources within a large data space”. Every recommendation system has two basic entries: the user and the item. The user is the person who is targeted by the recommendation system and uses it. They give their opinion on a variety of items and gets new recommendations from the system in return. The item is the recommendation provided by the system to the user. Each filtering algorithm has a specific kind of input data to be used for the recommender system. More often than not, they are three categories, the estimation, the demographic data and also content data.

## 4. RESULTS AND DISCUSSION

Recommender systems can be evaluated in different measures in order to determine their accuracy and their performance. Our approach is based on two metrics which are mean absolute error (MAE) and root mean square error (RMSE). The MAE measures the errors rate between paired different observations based on scale-dependent accuracy measure. The RMSE gives as a predicted values based on some quantitative unstructured data. The MAE and RMSE errors divulgate the performance of using hybrid logic for movies recommendation datasets.

### 4.1. Mean absolute error (MAE)

$MAE = (\frac{1}{|R|}) \sum_{u,i,r \in R} |\hat{r}_{u,i} - r_{u,i}|$ . Here,  $\hat{r}_{u,i}$  is the predicted rating for user  $u$  on item  $I$ ,  $r_{u,j}$  is the actual rating, and  $N$  is the total number of ratings on the item set. Accuracy of the predictions made by the recommendation engine for user rating goes up when the MAE goes down [20].

### 4.2. Root mean square error (RMSE)

Larger absolute error is emphasized by the root mean square error (RMSE) [21]:

$$RMSE = \sqrt{\frac{1}{|R|} \sum_{u,i,r \in R} (\hat{r}_{u,i} - r_{u,i})^2}$$

the lower of the RMSE, the better accuracy of the recommender system. Over the last handful of decades, film recommendations using multiple techniques have been largely studied, including the alternating least square (ALS) algorithm [22]-[24] which is based on the weighting technique and the item collaborative filtering based on similarity. Previous information on the evaluations of the films generated by the user is required by these techniques that essentially use movie lens data sets in order to achieve evaluation. Nonetheless, accuracy of these systems is still yet to be proven and there is on-going research for real-time improving performance of these systems.

### 4.3. Datasets

As mentioned above, the movie lens dataset [25], which is collected and managed by the group lens organization, is the one most used for movie recommendation systems. Otherwise, experiments are done on public and standard datasets. Figure 2 shows the properties of the movie lens datasets.

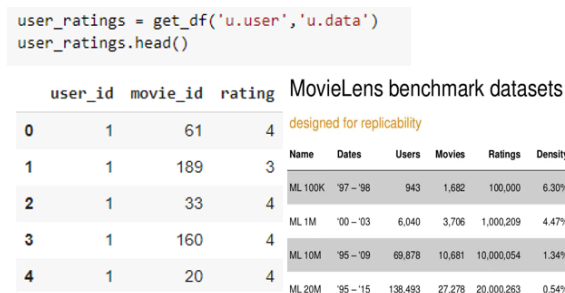


Figure 2. MovieLens dataset

Dataset is a measurement unit for information published in a public open data storehouse, which is used by researchers to perform experiments. Datasets contains one or more database tables where columns express variables and rows express a given record from the data. Datasets are divided into two sections: training set and test set. This division allows them to achieve the aimed result. The model uses the training set to run, which is then compared to the larges result. Estimation of the final model is provided by application of the test data. Movie datasets are of considerable amount and widely available. Some of the examples of such datasets are: MovieLens 100K, MovieLens-10M, and so on. Fields found in these datasets are user ID, item ID, and others. All models were implemented using the scikit-learn and surprise framework in Python. First, the data was imported into the Pandas data framework so that the matrices could be created and used by all models. Figure 3 gives details about our used framework by listing all used packages.

```
[ ] import pandas as pd
import numpy as np
from surprise import Reader
from surprise import Dataset
from surprise.model_selection import cross_validate
from surprise import KNNBasic
from surprise import KNNWithMeans
from surprise import SVD
from surprise import BaselineOnly
from surprise import SVDpp
from surprise import NMF
from surprise.accuracy import rmse
from surprise import accuracy
from surprise.model_selection import train_test_split

import matplotlib.pyplot as plt
%matplotlib inline
```

Figure 3. Framework used

A short summary of the results using the open data-set MovieLens is presented. The results show that SVD++ produces the best results, but not by a large margin. In addition, SVD also produces good result and becomes the second-best model on this data-set. The figure also shows the computation time for all models. We can see here some interesting results. The most advanced model, SVD++, has a significantly longer learning time, longer than the SVD model. Additionally, on one hand, we can also see that all the matrix factorization models have slightly longer learning times while having short test times. On the other hand, the KNN model is the opposite, with a longer test time relative to its learning time. The objective of this paper was to evaluate different recommendation models based on machine learning. Traditional recommendation algorithms are based on neighborhood models such as k-nearest neighbor, while new state-of-the-art models are based on matrix factorization. Overall, the goal and challenge of this research was to find the best performing recommendation model on the movie dataset. Figure 4 gives details about RMSE and MAE MovieLens representation.

Algorithm	test_rmse	test_mae	fit_time	test_time
<b>SVDpp</b>	0.902478	0.707675	66.588792	3.876131
<b>SVD</b>	0.921117	0.724371	2.712908	0.223839
<b>KNNWithMeans</b>	0.925014	0.728794	0.166343	4.214886
<b>BaselineOnly</b>	0.925766	0.731707	0.074889	0.162912
<b>NMF</b>	0.936850	0.736709	2.897876	0.194182
<b>KNNBasic</b>	0.946700	0.746637	0.144948	3.977553

Figure 4. RMSE and MAE for MovieLens

From the experiments performed in this research on the MovieLens dataset, it is clear that matrix factorization is the superior technique. In the following figure, the results of the RMSE and MAE error measures

show that the SVD and SVD++ matrix factorization techniques consistently produced results with fewer errors than the nearest neighbors. Figure 5 shows the results related to SVD evaluation:

```

Evaluating RMSE, MAE of algorithm SVD on 3 split(s).
RMSE (testset)  Fold 1  Fold 2  Fold 3  Mean  Std
MAE (testset)  0.9144  0.9226  0.9263  0.9211  0.0050
Fit time       0.7192  0.7276  0.7263  0.7244  0.0037
Test time     2.74   2.70   2.70   2.71   0.02
Test time     0.18   0.18   0.31   0.22   0.06
    
```

Figure 5. SVD evaluation

in the following Figure 6, we represents the results related to the KNN evaluation:

```

RMSE (testset)  Fold 1  Fold 2  Fold 3  Mean  Std
MAE (testset)  0.7443  0.7481  0.7476  0.7466  0.0017
Fit time       0.13   0.15   0.15   0.14   0.01
Test time     3.90   3.98   4.05   3.98   0.06
    
```

Figure 6. KNN evaluation

in Figure 7, we have represented the global prediction and actual sum of the SVD evaluations. Also the Figure 8 gives us an idea about estimated ration for the user id=9:

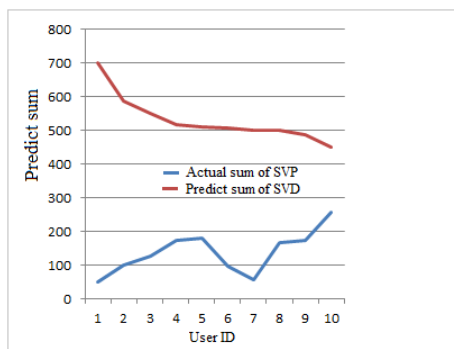


Figure 7. Actual and predicted sum of evaluations-SVD

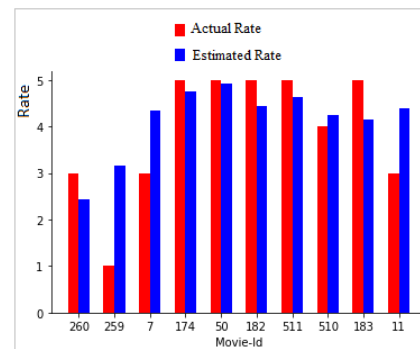


Figure 8. Actual and estimated rating for user with ID equals 8

in the Figure 9, we have represents the actual and predicted sum of evaluations-KNN and especially in Figure 10, we gives the rating of the actual and estimated order for a special user with ID equals 8.

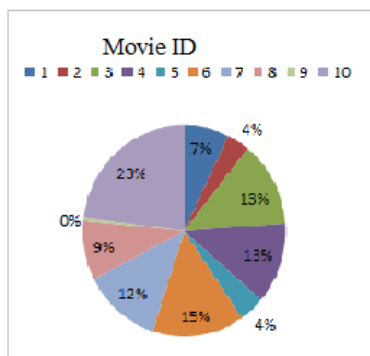


Figure 9. The % of actual and predicted sum of evaluations-KNN

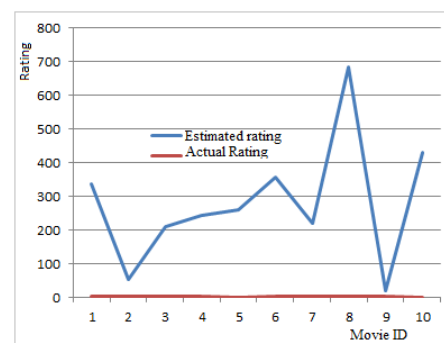


Figure 10. Actual and estimated rating for user with ID equals 8




## 5. CONCLUSION

The objective of this project was to evaluate different recommendation models based on machine learning. As mentioned earlier, traditional recommendation algorithms are based on neighborhood models and new ones are based on matrix factorization. Therefore, several models were evaluated using both of error and precision (accuracy). Overall, the goal and challenge has been to find the best performing recommendation model on movie-lens datasets. Thus, returning to the original research question which was: how does matrix factorization compare to neighborhood models using standard metrics to generate specific content recommendations? From the experiments done on the MovieLens dataset, it is clear that matrix factorization is the superior technique and it gives a good result with a high score of precision.




## REFERENCES

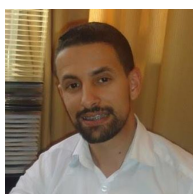
- [1] D. K. Behera, M. Das, S. Swetanisha, and P. K. Sethy, "Hybrid model for movie recommendation system using content k-nearest neighbors and restricted boltzmann machine," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 23, no. 1, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp445-452.
- [2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005, doi: 10.1109/TKDE.2005.99.
- [3] J. Daher, A. Brun, and A. Boyer, "A review on explanations in recommender systems," Technical Report LORIA -Université de Lorraine, Nancy, France, 2017.
- [4] G. Shani and A. Gunawardana, "Evaluating recommendation systems," In *Recommender systems handbook*, 2011, pp. 257-297, doi: 10.1007/978-0-387-85820-3\_8.
- [5] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowledge-Based Systems*, vol. 157, pp. 1–9, 2018.
- [6] F. Fkih, "Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison," *Journal of King Saud University-Computer and Information Sciences*, 2021, doi: 10.1016/j.jksuci.2021.09.014.
- [7] M. Brunato and R. Battiti, "A location-dependent recommender system for the web," In *Proceedings of the MobEA Workshop*, 2003.
- [8] J. Salter and N. Antonopoulos, "CinemaScreen recommender agent: combining collaborative and content-based filtering," *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 35-41, 2006, doi: 10.1109/MIS.2006.4.
- [9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.
- [10] T. M. Cover, P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.
- [11] F. Ricci, "Travel recommender systems," *IEEE Intelligent Systems*, vol. 17, no. 16, pp. 55-57, 2002.
- [12] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction volume*, vol. 12, no. 4, pp. 331–370, 2002, doi: 10.1023/A:1021240730564.
- [13] R. Burke, "Hybrid web recommender systems," *The Adaptive Web*, pp. 377–408, 2007, doi: 10.1007/978-3-540-72079-9\_12.
- [14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5-53, 2004, doi: 10.1145/963770.963772.
- [15] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research*, vol. 10, no. 2, 2009.
- [16] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient KNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774-1785, 2018, doi: 10.1109/TNNLS.2017.2673241.
- [17] A. A. Prasanti, M. A. Fauzi, and M. T. Furqon, "Neighbor weighted k-nearest neighbor for sambat online classification," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 12, no. 1, pp. 155-160, 2018, doi: 10.11591/ijeecs.v12.i1.pp155-160.
- [18] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Inter-Research Science Center*, vol. 30, no. 1, pp. 79-82, 2005.
- [19] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75-79, 2007, doi: 10.1145/1345448.1345465.
- [20] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014, doi: 10.5194/gmd-7-1247-2014.
- [21] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize," In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, 2008, pp. 337–348, doi: 10.1007/978-3-540-68880-8\_32.
- [22] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, "Matrix completion and low rank SVD via fast alternating least squares," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3367-3402, 2014.
- [23] F. M. Harper and J. A. Konstan, "The MovieLens datasets: history and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1-19, 2016, doi: 10.1145/2827872.
- [24] A. Ez-zahout, "A distributed big data analytics model for people re-identification based dimensionality reduction," *International Journal of High Performance Systems Architecture*, vol. 10, no. 2, pp. 57-63, 2021, doi: 10.1504/IJHPSA.2021.119147.
- [25] M.T. Nguyen, T. Nakajima, M. Yoshimi and N. Thoai, "Analyzing and predicting the popularity of online contents," In *Iii-WAS2019: 21st Int. Conf. on Information Integration and Web-based Applications and Services*, Dec. 2019, pp. 93–102, doi: 10.1145/3366030.3366047.




**BIOGRAPHIES OF AUTHORS**

**Abderrahmane Ez-Zahout**    is currently an assistant professor of computer science at department of computer science/faculty of sciences at Mohammed V University. His researches are in fields of computer sciences, digital systems, big data and computer vision. Recently, He works on intelligent systems. He has served as invited reviewer for many journal. Besides, he is also involved in NGOs, student associations, and managing non-profit foundation. He can be contacted at email: [abderrahmane.ezzahout@um5.ac.ma](mailto:abderrahmane.ezzahout@um5.ac.ma).






**Hicham Gueddah**    is currently an assistant professor of computer science at department of computer science/ENS, Mohammed V University. He is a TPC member of many international conferences on the fields of NLP and computer vision and also all applications of processing in arabic Language. He can be contacted at email: [h.gueddah@um5s.net.ma](mailto:h.gueddah@um5s.net.ma).






**Rabie Madani**    is currently a 1st year Ph. D student at faculty of sciences, Mohammed V University. He has a master degree on computer sciences and big data, he is actually an administrator of the information system on the institute of arabization and also the institute of african studies at Mohammed V University. He has launched his research cursus with publishing this paper in collaboration with us at IJEECS journal. He can be contacted at email: [rabie.madani@um5.ac.ma](mailto:rabie.madani@um5.ac.ma).



**Abir Nasry**    is currently a 1st year Ph. D student at faculty of sciences, Mohammed V University. She has obtained her master degree from faculty of sciences at 2021, she is master speciality is data engineering and computer sciences development. She is very active by patience at the field of big data and analytics. She can be contacted at email: [ennasri.a@gmail.com](mailto:ennasri.a@gmail.com).



**Fouzia Omary**    is the leader IPSS team research and currently a full professor of computer science at department of computer science/ENS, Mohammed V University. She is a full time researcher at Mohammed V University and she is the chair of the international conference on cybersecurity and blockchain and also the leader of the national network of blockchain and cryptocurrency. She can be contacted at email: [omary@fsr.ac.ma](mailto:omary@fsr.ac.ma).