Jurnal CoreIT, Vol. 6, No. 2, Desember 2020
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

Mesin Rekomendasi menggunakan Algoritma *Alternating Least Square* (ALS) pada Goodreads

Stanley Martin[1], Jay Idoan Sihotang[2], Bern Jonathan[3]
[1,2]Fakultas Teknologi Informasi, Universitas Advent Indonesia
Jl. Kolonel Masturi No.288, Cihanjuang Rahayu, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat 40559 Telp. +628112311949
[3]Departemen Teknologi & Data, Female Daily Network
Lantai 4 Wisma Prima, Jl. Kapten Tendean No.34, RT.2/RW.1, Mampang Prpt., Kec. Mampang Prpt., Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12720 Telp. (021) 79182121
1781009@unai.edu[1], jay.sihotang@unai.edu[2], bernjonath@gmail.com[3]

***Abstrak* – Sistem rekomendasi merupakan sistem yang dapat memberikan rekomendasi berupa prediksi *rating* pada suatu *item*. Ada banyak cara dalam merekomendasikan suatu *item* kepada pengguna, salah satunya adalah *Alternating Least Square*. Goodreads adalah situs web buku yang menjelaskan sinopsis dan memberi peringkat untuk buku, dan Goodreads membagikan peringkat pengguna mereka ke *item* di Kaggle untuk dianalisis. Oleh karena itu, penulis ingin merancang, mengimplementasikan, menguji serta ingin mengevaluasi mesin rekomendasi buku Goodreads mulai dari *rating* pengguna hingga *item* sebagai alternatif pemecahan masalah dari rekomendasi peringkat buku di Goodreads sekarang ini. Terdapat 981.756 data yang diolah menggunakan *Alternating Least Square* dengan 80% data latih dan 20% data uji. Hasil prediksi dievaluasi dengan *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE). Hasil penelitian ini menunjukkan bahwa *Root Mean Square Error* 0.67 dan *Mean Absolute Error* 0.52 dan dapat dikatakan mesin yang dibuat memiliki *error varians* yang kecil karena memiliki kesalahan di bawah 1.00.**

Kata Kunci – Collaborative Filtering, Alternating Least Square, Mesin Rekomendasi

PENDAHULUAN

Pada era ini teknologi telah berkembang pesat dalam kehidupan manusia. Di Industri 4.0 sekarang, semuanya bergantung pada internet. Apapun yang manusia butuhkan semua bisa disediakan oleh internet itu sendiri, termasuk informasi tentang apa yang manusia suka atau minati. Salah satu teknologi yang membantu manusia dalam memberikan informasi tentang apa yang disukai atau minati adalah sistem rekomendasi.

Sistem rekomendasi adalah proses analisis dinamis, yang dilakukan secara keseluruhan mengenai suatu produk dan juga tentang pelanggan yang terkait dengan produk tersebut [1]. Ada 3 bagian dari sistem rekomendasi itu sendiri, yaitu pemfilteran kolaboratif, pemfilteran berbasis konten, dan pemfilteran *hybrid* (kombinasi kolaboratif dan berbasis konten) [2]. Pemfilteran kolaboratif adalah salah satu metode untuk merekomendasikan sesuatu dengan peringkat dari banyak pengguna untuk sesuatu dan dengan peringkat secara kolaboratif dapat memprediksi satu peringkat yang tidak terisi [3]. Pemfilteran kolaboratif ini cukup penting karena akan menambah pengalaman pengguna dalam menggunakan aplikasi. Dengan demikian, ketika ada item yang disukai oleh seseorang atau diminati, orang tersebut akan lebih mudah mendapatkannya karena adanya *filtering*. Biasanya orang mendapatkan pemfilteran kolaboratif ini pada banyak aplikasi web atau belanja seperti sepeda motor, Tokopedia, Amazon, atau pada aplikasi atau web buku bacaan *online*, seperti Google Books, Bookrix, OpenLibrary, Goodreads, dan Smashwords.

Goodreads adalah situs yang mengkhususkan diri dalam pembuatan katalog buku sosial, yang mana di situs ini memungkinkan pengguna untuk memberikan penilaian terhadap buku atau komentar pada buku tersebut [4]. Penilaian pembaca dapat kita temukan melalui rating atau komentar yang dia berikan terhadap buku yang telah dia baca. Tujuan penulis adalah menyediakan mesin rekomendasi sehingga peringkat yang diberikan oleh pengguna tersebut dapat merekomendasikannya kepada pengguna lain.

Dalam penelitian ini, penulis telah merancang, mengimplementasikan, menguji, serta mengevaluasi mesin yang akan dibuat. Mesin yang dibangun lebih mengarah kepada pembuatan rekomendasi buku dengan mengambil contoh data kasus buku-buku yang ada di Goodreads, sehingga ini dapat juga nantinya diimplementasikan ke *platform* buku lainnya. Alasan penulis memilih situs Goodreads oleh karena Goodreads adalah salah satu situs buku yang sangat besar, dengan data buku, *user*, dan *rating* yang sangat banyak, dan juga data-data tersebut dapat diperoleh dengan cukup mudah. Penulis juga menggunakan algoritma untuk dapat membuat mesin rekomendasi sebagai pengganti buku. Penulis menggunakan algoritma *Alternating Least Square* (ALS), di mana fungsi yang diperkirakan menyimpang sesedikit mungkin dari f(x).

Indah dalam penelitiannya yang berjudul "Mesin Rekomendasi Film Menggunakan Metode Kemiripan

Genre Berbasis *Collaborative Filtering*", menggunakan algoritma ALS-WR dan *cosine similarity* dengan menggunakan mesin Mlib Apache Spark. Indah kemudian membagi data menjadi 3 bagian, yaitu data latih, uji, dan validasi, namun hasil evaluasi yang dilakukan dengan RMSE masih mendekati angka 1 [5]. Setelah itu Fadlil dengan Wayan dalam penelitiannya yang berjudul "Pembuatan Sistem Rekomendasi Menggunakan *Decision Tree dan Clustering*", menggunakan klasifikasi *Decision Tree* dan algoritma *K-Means Clustering* dan mendapatkan kesimpulan bahwa perbedaan antara keduanya kurang dari angka 1, namun metode ini akan berpengaruh apabila *cluster* tidak memiliki anggota, sehingga penulis harus membagi anggota dari *cluster* yang memiliki anggota terbanyak [6]. Kemudian Eko, Siti, dan Abdul dalam penelitiannya yang berjudul "Penerapan Algoritma Squeezer untuk Memberikan Rekomendasi Pilihan Lagu Berdasarkan Daftar Lagu yang Dimainkan pada Pemutar Mp3 Android", menggunakan algoritma Squeezer. Mereka membagi data latih sebanyak 300 data dan data uji sebanyak 50 data, sehingga dihasilkan tingkat akurasi sebesar 84%. Namun penulis mengatakan bahwa dengan menggunakan data riwayat lagu yang sama tetapi dengan konfigurasi yang berbeda akan menghasilkan rekomendasi yang berbeda juga [7].

METODE PENELITIAN

Alur pemodelan mesin rekomendasi yang digunakan adalah dengan tahap pertama memecah data menjadi 80% *data training* dan 20% *data testing*. 80% *data training* yang sudah dipecah tadi selanjutnya akan dipelajari oleh algoritma *Alternating Least Square* (ALS), sehingga dengan pembelajaran dari algoritma terhadap data tersebut menghasilkan suatu model mesin rekomendasi. Model mesin rekomendasi tersebut selanjutnya akan memprediksi sisa 20% data test tadi. Pada tahap terakhir, hasil prediksi model terhadap *data test* akan di evaluasi dengan menggunakan 2 *evaluator*, yaitu *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE) dengan 0 sebagai *best value*. Secara ringkas alur dapat dilihat pada Gambar 1 di bawah ini.

Gambar 1. Alur Metode Penelitian

A. Pemfilteran Kolaboratif

Pemfilteran Kolaboratif didasarkan pada ide sederhana — pengguna akan lebih memilih *item* jika direkomendasikan oleh teman yang berpikiran sama. Misalkan ada restoran baru di kota. Jika seorang teman, yang kebetulan memiliki minat yang sama dengan pengguna, memuji hal itu — pengguna mungkin juga menyukainya. Idenya masuk akal secara intuitif. Pemfilteran kolaboratif memanfaatkan wawasan ini ke dalam algoritma yang memprediksi preferensi pengguna atas suatu item dengan menemukan kelompok pengguna yang kebetulan memiliki preferensi yang sama dengan pengguna. Preferensi atau peringkat yang diprediksi untuk suatu *item* dapat disimpulkan oleh peringkat yang diberikan oleh kelompok pengguna serupa [8].

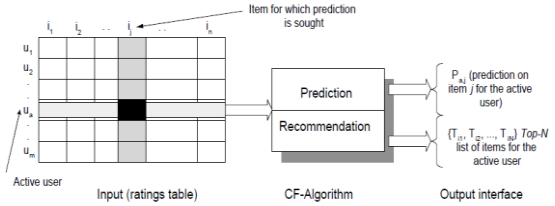
Collaborative filtering atau yang biasa disebut dengan *crowd-wisdom* atau pemfilteran berbasis pengguna adalah 1 dari 3 metode pemfilteran yang menggunakan *data rating* dari pengguna untuk menghasilkan kontribusi. Pemfilteran kolaboratif kemudian digunakan untuk memprediksi *item* yang mungkin menarik bagi pengguna. Pada dasarnya, pemfilteran kolaboratif adalah peringkat yang tidak ditentukan atau item tanpa peringkat dapat dihitung karena perbedaan antara pengguna dan *item*. Berikut adalah keluarga luas dari fungsionalitas sistem CF yang umum:

1) Merekomendasikan item. Tampilkan daftar item kepada pengguna, dalam urutan seberapa berguna *item* tersebut. Seringkali ini dideskripsikan sebagai memprediksi apa yang pengguna akan menilai *item*, kemudian memberi peringkat item berdasarkan peringkat yang diprediksi ini. Namun, beberapa algoritma rekomendasi yang berhasil tidak menghitung nilai peringkat yang diprediksi sama sekali. Misalnya, algoritma rekomendasi Amazon mengumpulkan *item* yang mirip dengan pembelian dan penilaian pengguna tanpa pernah menghitung peringkat yang diprediksi [9]. Alih-alih menampilkan peringkat prediksi yang dipersonalisasi, antarmuka pengguna mereka menampilkan peringkat pelanggan rata-rata. Akibatnya, daftar rekomendasi mungkin tampak rusak sehubungan dengan nilai peringkat rata-rata yang ditampilkan. Dalam banyak aplikasi, memilih beberapa item teratas dengan baik sangatlah penting; menghasilkan nilai prediksi adalah sekunder.

2) Memprediksi *item* tertentu. Dengan mempertimbangkan *item* tertentu, hitung peringkat prediksi nya. Perhatikan bahwa prediksi bisa lebih menuntut daripada rekomendasi. Untuk merekomendasikan *item*, sistem hanya perlu disiapkan untuk menawarkan beberapa alternatif, tetapi tidak semua. Beberapa algoritma memanfaatkan hal ini agar lebih dapat diskalakan dengan menghemat memori dan waktu komputasi [9], [10]. Untuk memberikan prediksi untuk item tertentu, sistem harus siap untuk mengatakan sesuatu tentang item yang diminta, bahkan yang jarang dinilai. Bagaimana sistem memutuskan bagaimana pengguna tertentu akan menilai item yang diminta

jika sangat sedikit pengguna - apalagi pengguna yang serupa dengan pengguna tertentu - telah menilai *item* tersebut? Prediksi yang dipersonalisasi mungkin menantang, jika bukan tidak mungkin.

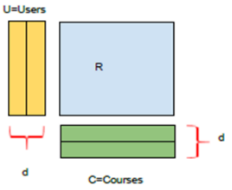
3) Rekomendasi terbatas: Rekomendasikan dari satu set item. Diberikan set tertentu atau batasan yang memberikan satu set *item*, merekomendasikan dari dalam set itu. Sebagai contoh: "Pertimbangkan skenario berikut. Keponakan Mary yang berusia 8 tahun berkunjung untuk akhir pekan, dan dia ingin mengajaknya ke bioskop. Dia ingin film komedi atau keluarga dengan rating tidak "lebih tinggi" dari PG-13. Dia lebih suka film tersebut tidak berisi seks, kekerasan atau bahasa yang menyinggung, berdurasi kurang dari dua jam dan, jika memungkinkan, ditayangkan di teater di lingkungannya. Akhirnya, dia ingin memilih film yang mungkin dia sukai." [11]



Gambar 2. Proses Penyaringan Kolaboratif

B. *Alternating Least Square* (ALS)

Alternating Least Squares [12], [13], [14] adalah algoritma dalam paradigma penyaringan kolaboratif. ALS digunakan untuk menguraikan lebih banyak skor matriks menjadi produk matriks vektor eigen berdasarkan angka seperti peringkat, kemudian menggunakan ALS ini untuk menghitung vektor eigen pengguna atau item dan mengambil jumlah residual yang dihitung, serta peringkat yang dapat diprediksi dari pengguna suatu *item*. Singkatnya, ALS adalah algoritma yang digunakan dalam paradigma *Collaborative Filter* yang digunakan untuk menghitung jumlah kuadrat 1 antara nilai prediksi dan nilai valuasi aktual. Input ALS (pada Gambar 3) adalah *user sparse* dengan matriks item R yang berisi *rating* dari tiap *user*. Algoritma ini secara iteratif menghitung faktorisasi matriks peringkat rendah R = U ⇥ V di mana U dan V adalah matriks dimensi d. Fungsi kerugian didefinisikan sebagai kesalahan kuadrat [13], di mana tujuan pembelajarannya adalah untuk meminimalkan jumlah kesalahan kuadrat 1 antara nilai yang diprediksi dan nilai nyata dari kata-kata kasar. Kompleksitas dan biaya bergantung pada besarnya variabel tersembunyi d.



Gambar 3. Faktorisasi Matriks ALS

Algoritma ALS mahal secara komputasi, setiap iterasi berjalan pada persamaan (1), di mana m adalah panjang *item*, dan n adalah panjang pengguna [15], [16].

$$O(d^{-1}[Nr + m + n)\, r^2] + r^3) \qquad (1)$$

C. *Mean Absolute Error* (MAE)

MAE adalah pengukuran yang berguna, yang banyak digunakan dalam model evaluasi. MAE mencari perbedaan yang nyata, dimana perbedaan tiap penggunanya adalah sama. MAE digunakan dalam sistem rekomendasi untuk mendapatkan peringkat pengguna yang benar.

$$MAE = \frac{\sum_i^j = 1 \mid Pi - qi \mid}{\mathrm{N}}$$

Gambar 4. Rumus Mean Absolute Error

Gambar 4 menyatakan rumus untuk mencari MAE, dimana N adalah jumlah total peringkat nyata, *Pi* adalah peringkat prediksi pengguna dalam daftar, dan *qi* adalah peringkat sebenarnya dari pengguna yang paling mereka sukai berdasarkan preferensi pengguna. MAE adalah yang terbaik untuk mendapatkan hasil yang akurat atas nama peringkat pengguna [17].

D. *Root Mean Square Error* (RMSE)

RMSE merupakan salah satu metode alternatif untuk menentukan cara melakukan prediksi, yang digunakan untuk mengukur keakuratan hasil prediksi. RMSE ini merupakan nilai rata-rata dari penjumlahan kuadrat kesalahan, jadi jika nilai RMSE rendah berarti variasi nilai yang dihasilkan dari variasi pengamatan.

HASIL DAN PEMBAHASAN

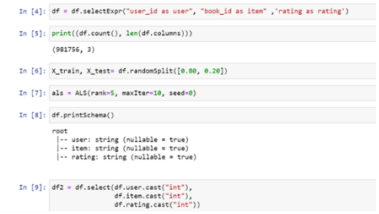
Implementasi mesin dilakukan dalam tahap sebagai berikut:

1) Mengambil *dataset* yang diperlukan. Gambar 5 memberikan gambaran secara rinci mengenai kode pengambilan *dataset* dari *file* .csv sampai memberikan tampilan awal dari *dataframe*.

Gambar 5. Proses Pengambilan *Dataset*

2) Melakukan *data cleaning* terhadap *dataset* yang ada. Pada Gambar 6 di bawah menjelaskan secara rinci

setiap kode untuk membuat data bersih dan siap digunakan.



Gambar 6. Proses *Data Cleaning*

3) Membagi *dataset* yang sudah dilakukan cleaning ke dalam 2 bagian, *Data Training* (80%) dan *Data Testing* (20%). Gambar 7 memberikan kode bagaimana pembagian Data Latih sebesar 80% dari total data dan Data Uji sebesar 20% dari total data.

```
X_train, X_test= df2.randomSplit([0.80, 0.20])
```

Gambar 7. Proses Pembagian Data Latih & Data Uji

4) *Data Training* dimasukkan ke dalam algoritma *Alternating Least Square* (ALS) agar mesin dapat belajar dengan data tersebut. Gambar 8 menjelaskan ALS mempelajari Data Latih dan dimasukkan ke dalam Model.

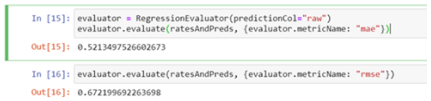
```
model = als.fit(X_train.select(["user", "item", "rating"]))
```

Gambar 8. Proses Pembelajaran Model

5) *Data Test* akan dilakukan prediksi dengan mesin yang sudah belajar tadi. Pada Gambar 9 dijelaskan lebih lanjut mengenai kode-kode dan memberikan hasil prediksi dari *Model* pada kolom *Raw*.

Gambar 9. Proses Prediksi

6) Setelah di prediksi dengan mesin, diambil MAE dan RMSE sebagai bagian dari hasil evaluasi dari tingkat kebenaran prediksi. Dari hasil *evaluator*, didapatkan MAE dan RMSE dibawah 1, mengartikan bahwa model yang telah dibuat sudah cukup baik (best value keduanya adalah 0.0). Gambar 10 berikut menyajikan kode dan hasil evaluasi MAE & RMSE.



Gambar 10. Proses Evaluasi MAE & RMSE

7) Dari baris kode yang dimuat dalam Gambar 11 di bawah, didapatkan hasil akhir yaitu 4 rekomendasi, yaitu:

Gambar 11. Proses Pemberian Hasil Rekomendasi

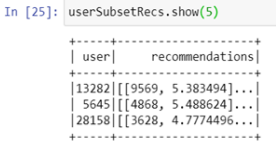
a) 10 rekomendasi buku teratas untuk setiap pengguna. Gambar 12 di bawah ini menampilkan ID User dan Rekomendasi Buku.

Gambar 12. Hasil Rekomendasi Poin A

b) 10 rekomendasi pengguna teratas untuk setiap buku. Gambar 13 di bawah ini (In [24]) menampilkan ID Buku dan Rekomendasi *User*.

Gambar 13. Hasil Rekomendasi Poin B

c) 10 rekomendasi buku teratas untuk sekelompok pengguna tertentu. Gambar 14 di bawah ini (In [25]) menampilkan beberapa ID User dan Rekomendasi Buku.



Gambar 14. Hasil Rekomendasi Poin C

d) 10 rekomendasi pengguna teratas untuk satu set buku tertentu. Gambar 15 di bawah ini (In [26]) menampilkan beberapa ID Buku dan Rekomendasi *User*.

```
In [26]: bookSubSetRecs.show(5)

+----+--------------------+
|item|     recommendations|
+----+--------------------+
| 296|[[45567, 6.780540...|
| 675|[[34886, 5.579472...|
| 467|[[41622, 6.168834...|
+----+--------------------+
```

Gambar 15. Hasil Rekomendasi Poin D

KESIMPULAN DAN SARAN

Dari 981.756 data yang ada, penulis membuat 80% data menjadi data latih dan 20% lainnya menjadi data uji. Model yang dibuat berhasil mempelajari 80% data yang ada, sehingga dapat menebak atau memprediksi data yang di uji. Penulis menggunakan algoritma *Alternating Least Square* (ALS) untuk membuat model mesin cukup bagus untuk memprediksi.

Akurasi prediksi *rating* menggunakan RMSE yaitu bernilai 0.6721996922637, sedangkan menggunakan MAE yaitu bernilai 0.5213497526602673. Jika diubah dalam bentuk presentase maka nilai RMSE & MAE yang didapat harus dibagi dengan nilai RMSE atau MAE maksimal yaitu 5 dan dikalikan dengan 100 sehingga didapatkan nilai *error* dari sistem rekomendasi adalah 13.444 % atau dapat dikatakan bahwa akurasi dari sistem rekomendasi adalah 86.556 % (RMSE) dan 89.573 % (MAE). Dapat dikatakan bahwa dari 20% data uji atau 196.351 data dapat diprediksi secara benar sebanyak 169.954 data (berdasarkan RMSE) atau sebanyak 175.877 data (berdasarkan MAE).

Penelitian ini memiliki beberapa keterbatasan, sehingga penulis menyarankan untuk dapat menggunakan data yang lebih banyak dan lebih *update* lagi, sehingga dapat memberikan tingkat akurasi yang lebih baik dan lebih *real*. Kemudian penulis menyarankan untuk dapat menambahkan beberapa parameter dalam memberikan rekomendasi seperti pembuat buku, pemberi *rating*, dan lainnya. Penulis juga menyarankan untuk dapat menggunakan algoritma yang lebih terbaru dan lebih efisien, dengan demikian itu dapat memungkinkan hasil akurasi menjadi lebih meningkat.

REFERENSI

[1] F. Ricci, L. Rokach and B. Shapira, "Recommender Systems: Introduction and Challenges," in Recommender systems handbook, Second edition, New York, Springer Science+Business Media, 2015, pp. 1-34.

[2] P. B.Thorat, R. Goudar and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," International Journal of Computer Applications, pp. 31-36, 2015.

[3] J. B. Schafer, D. Frankowski, S. Sen and J. Herlocker, "Collaborative Filtering Recommender Systems," Springer-Verlag Berlin Heidelberg, pp. 291-324, 2007.

[4] M. Thelwall and K. Kousha, "Goodreads: A Social Network Site for Book Readers," Journal of the Association for Information Science and Technology, pp. 1-23, 2016.

[5] I. S. Wahyudi, "Big Data Analytic untuk Pembuatan Rekomendasi Koleksi Film Personal Menggunakan Mlib Apache Spark," Berkala Ilmu Perpustakaan dan Informasi, vol. 14, pp. 11-25, 2018.

[6] J. Fadlil and W. F. Mahmudy, "Pembuatan Sistem Rekomendasi Menggunakan Decision Tree dan Clustering," Kursor, vol. 3, pp. 45-66, 2007.

[7] E. W. Wibowo, S. Rochimah and A. Munif, "Penerapan Algoritma Squeezer untuk Memberikan Rekomendasi Pilihan Lagu Berdasarkan Daftar Lagu yang Dimainkan pada Pemutar MP3 Android," Jurnal Teknik Pomits, vol. 2, pp. A-111 - A-116, 2013.

[8] V. Kotu and B. Deshpande, "Collaborative Filtering," in Data Science: Concepts and Practice, Cambridge, Morgan Kaufmann Publishers, 2019, p. 353.

[9] G. Linden, B. Smith and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, pp. 76-80, 2003.

[10] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," WWW '01: Proceedings of the 10th international conference on World Wide Web, pp. 285-295, 2001.

[11] B. Schafer, J. Konstan and J. Riedl, "E-Commerce Recommendation Applications," Data Mining and Knowledge Discovery, pp. 115-153, 2001.

[12] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin and J. M. Hellerstein, "Distributed GraphLab: A Framework for Machine Learning in the Cloud," Proceedings of the VLDB Endowment (PVLDB), Vol. 5, pp. 716-727, 2012.

[13] Y. Zhou, D. M. Wilkinson, R. Schreiber and R. Pan, "Large-Scale Parallel Collaborative Filtering for the Netflix Prize," in Algorithmic Aspects in Information and Management, 4th International Conference (AAIM), Shanghai, 2008.

[14] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, Vol. 42, pp. 30-37, 2009.

[15] S. Schelter, C. Boden, M. Schenck and A. Alexandrov, "Distributed Matrix Factorization

with MapReduce using a series of Broadcast-Joins," in The 7th ACM Conference on Recommender Systems,, New York, 2013.

[16] R. Gemulla, E. Nijkamp, P. J. Haas and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, 2011.

[17] B. M. Khan, A. Mansha, F. H. Khan and S. Bashir, "Collaborative Filtering based Online Recommendation Systems: A Survey," in International Conference on Information & Communication Technologies, Karachi, Pakistan, 2017.