Sistem Rekomendasi *Board-games* Berbasis Collaborative Filtering Menggunakan Matrix Factorization

1st Dicky Permata Putra
Telkom University
Fakultas Informatika
Bandung, Indonesia
dickypermatap@student.telkomuniversity.ac.id

2nd Agung Toto Wibowo
Telkom University
Fakultas Informatika
Bandung, Indonesia
agungtoto@telkomuniversity.ac.id

***Abstrak* — Sistem rekomendasi telah menjadi komponen penting dalam berbagai platform digital saat ini, terutama dalam membantu pengguna menemukan konten yang sesuai dengan preferensi mereka. Salah satu pendekatan yang paling sering digunakan adalah *collaborative filtering* karena mampu memprediksi item yang mungkin disukai pengguna. Pada penelitian ini bertujuan untuk membangun sistem rekomendasi dengan menggunakan tiga model algoritma *matrix factorization*, yaitu Singular Value Decomposition (SVD), SVD++, dan Non-negative Matrix Factorization (NMF), dengan dataset yang digunakan dari situs BoardGameGeek(BGG). Penelitian ini melakukan pengujian dengan menggunakan *cross validate* dengan skema pengujian 5 fold. Evaluasi kinerja performansi dilakukan menggunakan metrik *Root Mean Squared Error* (RMSE) dan *Mean Absolute Error* (MAE). Berdasarkan hasil pengujian, didapatkan nilai rata-rata terbaik RMSE 1.1809 MAE 0.8801 pada algoritma SVD.**

***Kata kunci*— Sistem Rekomendasi, Collaborative Filtering, Matrix Factorization, Cross Validation, BoardGameGeek**

I. PENDAHULUAN

Dalam beberapa tahun terakhir, industri *game* mengalami pertumbuhan yang terus berkembang pesat dalam skala global karena seiring dengan kemajuan teknologi dan platform digital. Sementara itu, permainan papan (*board games*) juga meraih popularitas yang sangat pesat. Penelitian sebelumnya, mencatat bahwa industri *board games* mengalami peningkatan penjualan sebesar 25% setiap tahunnya [1]. BoardGameGeek (BGG) adalah salah satu forum online terbesar untuk *board games*, forum ini berisi lebih dari 100.000 permainan dan 2 juta pengguna terdaftar. Namun, jumlah permainan yang sangat besar justru menimbulkan tantangan bagi pengguna dalam menentukan permainan yang sesuai dengan preferensi mereka [2]. Oleh sebab itu, sistem rekomendasi menjadi peran penting guna memberikan saran permainan yang sesuai dengan minat dan preferensi pengguna.

Sistem rekomendasi adalah salah satu algoritma yang digunakan untuk memprediksi item yang relevan kepada pengguna berdasarkan pola perilaku dan kesamaan preferensi dengan pengguna lain [3]. Salah satu pendekatan yang paling banyak digunakan saat ini yaitu *collaborative filtering* (CF) karena kesederhanaanya dalam penyediaan data. CF merekomendasikan secara akurat berdasarkan interaksi antara pengguna dan item [6].

Beberapa penelitian sebelumnya mengenai sistem rekomendasi menggunakan *collaborative filtering* (CF) telah banyak di implementasikan. Penelitian yang dilakukan oleh Luekhong et al.[13] pada sistem rekomendasi buku dengan menggunakan algoritma *Singular Value Decomposition* (SVD) untuk menghasilkan performa prediksi yang lebih efektif terutama pada kondisi *sparsity* data. Zalewski et al.[4] mengembangkan sistem rekomendasi berbasis klasterisasi hasilnya menunjukkan bahwa sistem rekomendasi mampu memberikan rekomendasi dengan baik, tetapi tidak optimal dalam perilaku pengguna seperti pola rating dan preferensi sebelumnya.

Dalam penelitiannya, Ion et al.[1] membandingkan berbagai pendekatan seperti, *collaborative filtering*, *content-based*, dan *hybrid*. Dengan hasil menunjukkan bahwa terdapat peningkatan yang signifikan pada seluruh metrik evaluasi. Sementara itu, penelitian lain menyebutkan bahwa *matrix factorization* dan *deep learning* dapat meningkatkan akurasi model antar item dan pengguna [5]. Namun, perlu dilakukan penelitian lebih lanjut untuk menemukan metode sistem rekomendasi yang lebih efektif dan efisien.

Penelitian ini bertujuan untuk mengimplementasikan *collaborative filtering* (CF) sebagai pendekatan utama sistem rekomendasi, CF merekomendasikan secara akurat berdasarkan interaksi antar pengguna dan item [6]. Penelitian ini juga menerapkan metode *matrix factorization* (MF), serta membandingkan performa algoritma SVD, SVD++, dan NMF dalam memprediksi rating pengguna. Perbandingan berbagai metode dilakukan untuk menunjukkan berbagai macam hasil dari penilaian pengguna sehingga dapat meningkatkan efektivitas sistem rekomendasi. MF digunakan karena telah terbukti efektif dalam mengidentifikasi faktor semantik laten [9]. Untuk mengevaluasi kinerja performa ketiga model tersebut, penelitian ini menggunakan metrik evaluasi *root mean squared error* (RMSE) dan *mean absolute error* (MAE).

II. KAJIAN TEORI

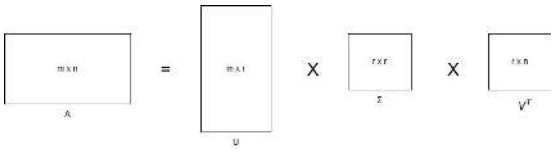
A. Sistem Rekomendasi

Menurut Zhang et al. (2019), Sistem rekomendasi merupakan sistem yang berperan penting dalam mengatasi permasalahan kelebihan informasi yang di alami oleh pengguna. Tujuan utama sistem rekomendasi yaitu membantu memilih informasi yang relevan dengan cara memprediksi preferensi pengguna dan menyarankan item berdasarkan interaksi ataupun historis sebelumnya.

Sistem rekomendasi memiliki beberapa metode yang biasanya digunakan yaitu, *collaborative filtering*, *content-based filtering*, dan *hybrid filtering*. Pada metode ini, *collaborative filtering* memberikan hasil rekomendasi berdasarkan interaksi antara historis pengguna dan item yang memiliki kesamaan preferensi.[5]

B. Collaborative Filtering

Saifudin dan Widyaningtyas (2024), pada penelitiannya menunjukkan bahwa, C*ollaborative filtering* (CF) adalah salah satu metode sistem rekomendasi dengan memanfaatkan interaksi pengguna sebelumnya antara pengguna dan item untuk menghasilkan rekomendasi[10]. CF memanfaatkan data *rating* pengguna sebagai masukan, yang kemudian direpresentasikan dalam bentuk *user-item rating*.



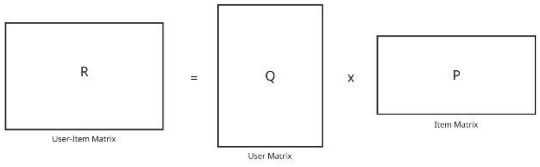
Gambar 2.1 Alur Kerja CF

Metode *collaborative filtering* (CF) merekomendasikan item berdasarkan minat preferensi antar pengguna atau mengidentifikasi kemiripan antar item berdasarkan penilaian pengguna sebelumnya. Secara umum, CF dapat dikategorikan ke dalam dua kategori yaitu *Memory-based* dan *Model-based* [12]. *Memory-based* melakukan pendekatan dengan menggunakan keseluruhan data pengguna dengan secara langsung untuk membuat prediksi, sedangkan *Model-based* memanfaatkan pembelajaran mesin dalam menghasilkan model yang digunakan untuk rekomendasi (Mu, 2018).

C. Matrix Factorization

Pada studi yang dilakukan oleh Duan et al. (2022) menjelaskan bahwa, *Matrix Factorization* (MF) adalah teknik yang umum digunakan dalam mengatasi masalah *sparsity* data pada metode *collaborative filtering*. Matriks ini bekerja dengan memecah matriks berukuran besar menjadi lebih kecil sehingga dapat merepresentasikan kedalam faktor laten pengguna dan item. Metode ini mempelajari faktor laten pengguna dan item yang tersembunyi sehingga menghasilkan dimensi yang lebih rendah untuk membantu mendapatkan sistem rekomendasi yang lebih baik [14].

Menurut (Isinkaye, 2021), Salah satu teknik yang paling efektif untuk menghasilkan sistem rekomendasi yaitu *matrix factorization*, MF memungkinkan memecah matriks penggunda dan item menjadi dua matriks berdimensi rendah yang mampu mengatasi *sparsity* data dan merekomendasikan item bagi pengguna dengan lebih efektif [17].



Gambar 2.2 Alur Matrix Factorization

Pada gambar tersebut, matriks factorization bekerja dengan membagi menjadi dua matriks yang berdimensi kecil yaitu matriks user dan item sehingga memungkinkan untuk menemukan fitur laten yang merepresentasikan preferensi pengguna dan karakteristik item.

D. Singular Value Decomposition (SVD)

Menurut Aciar et al. (2021), Singular Value Decomposition (SVD) adalah salah satu metode *matrix factorization* yang bekerja dengan mereduksi dimensi ruang vektor dari komponen matriks [3]. Secara matematis, rumus SVD dapat ditulis sebagai berikut:

$$A = U \cdot \Sigma \cdot V^T \tag{2.1}$$

Dimana A merupakan matriks awal, *U* sebagai matriks orthogonal, Σ sebagai matriks diagonal dan V^T adalah matriks transpos orthogonal. Dari persamaan(2.1), SVD mampu untuk mereduksi dimensi data sehingga dapat meningkatkan hasil akurasi sistem rekomendasi yang lebih akurat.

E. SVD++

Jiang et al. (2020) pada penelitiannya menjelaskan bahwa, SVD++ merupakan pengembangan lebih lanjut dari algoritma SVD, algoritma ini memanfaatkan *implicit feedback* dan informasi pengguna [16]. *Implicit feedback* adalah informasi yang secara tidak langsung menggambarkan preferensi interaksi pengguna dan item [18]. SVD++ dapat direpresentasikan dalam bentuk persamaan berikut:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p^u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \tag{2.2}$$

Dimana, setiap notasi yang digunakan dalam rumus dapat dijabarkan sebagai berikut:

- $\hat{r}_{ui}$ adalah prediksi *rating* oleh pengguna
- μ adalah rata-rata *rating* global
- b_u adalah matriks bias oleh pengguna
- b_i adalah matriks bias item
- q_i^T adalah matriks transpos faktor laten item
- p_u adalah matriks faktor laten pengguna
- $|N(u)|^{-\frac{1}{2}} \Sigma_{j \in N(u)} y_j$ adalah matriks *eigen vectors* dari *rating* pengguna

F. Non-negative Matrix Factorization (NMF)

(Isinkaye, 2021) *Non-negative Matrix Factorization* (NMF) adalah teknik dekomposisi matriks dengan memecah suatu matriks menjadi dua matriks faktor yang lebih kecil juga non negatif [17]. NMF mengasumsikan bahwa preferensi pengguna terhadap item dapat direpresentasikan oleh faktor laten yang tidak dapat diamati secara langsung [19]. Pendekatan NMF telah banyak digunakan karena terbukti efektif untuk meningkatkan akurasi prediksi *rating* dan mengatasi *sparsity* data, Liu dan Zhao (2023). Secara matematis, rumus dari NMF dapat ditulis sebagai berikut:

$$A \approx W \cdot H \tag{2.3}$$

Dengan demikian, setiap variabel dapat dijelaskan sebagai berikut:

- A merupakan matriks *rating* pengguna dan item berukuran $m \times n$
- W merupakan matriks representasi pengguna dalam faktor laten berukuran $m \times k$
- H merupakan matriks representasi item dalam faktor laten berukuran $k \times n$
- k merupakan jumlah faktor laten

G. Metrik Evaluasi

Berdasarkan penelitian sebelumnya yang dilakukan oleh Jayalakshmi et al. (2022) Pada umumnya, sistem rekomendasi memerlukan pengujian kinerja evaluasi. Oleh karena itu, berbagai aspek pada sistem rekomendasi harus diuji selain metrik akurasi untuk memastikan bahwa model yang telah dibangun dapat membuat prediksi yang efektif dan tepat [20]. Metode evaluasi dilakukan dengan cara membandingkan nilai prediksi model terhadap nilai *rating* aktual.

H. Root Mean Squared Error (RMSE)

Menurut Koren et al. (2022) *Root Mean Squared Error* (RMSE) adalah salah satu metrik evaluasi yang paling banyak digunakan dalam mngevaluasi akurasi *rating* yang diprediksi [9]. Metode evaluasi secara umum dilakukan dengan membandingkan nilai prediksi model terhadap nilai *rating* aktual, RMSE menghitung akar dengan mengambil rata-rata kuadrat. Secara matematis, rumus RMSE dapat ditulis sebagai berikut:

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}} \tag{2.4}$$

Dimana, $RMSE$ menunjukkan nilai akar rata-rata, N menunjukkan total *rating* yang akan di uji, $y_{(i)}$ menunjukkan nilai *rating* aktual pengguna terhadap item (i), sedangkan $\hat{y}(i)$ menunjukkan *rating* prediksi pengguna terhadap item (i).

I. Mean Squared Error (MSE)

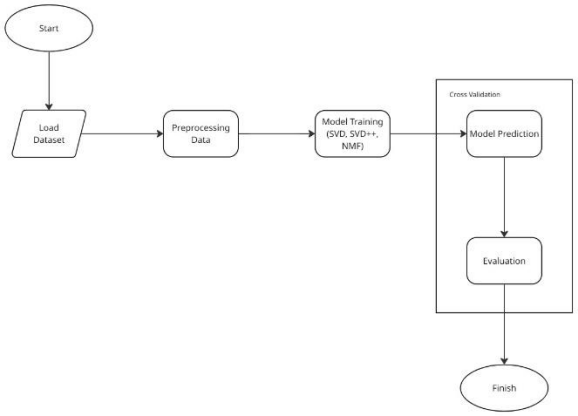
Raghuwanshi dan Pateriya (2019), *Mean Absolute Error* (MAE) adalah metrik evaluasi yang digunakan untuk menguji dan mengukur akurasi suatu model dalam melakukan *rating* rekomendasi [21]. MAE menghitung rata-rata selisih antara nilai prediksi dan nilai aktual. Dengan demikian, rumus MAE dapat dijabarkan sebagai berikut:

$$MAE = \frac{1}{N} \Sigma_{i=1}^{N} \|y_i - \hat{y}_i\| \quad (2.5)$$

Dimana, MAE merupakan nilai rata-rata dari nilai asli dan nilai prediksi pengguna, N merupakan jumlah data pengujian, y_i merupakan nilai asli pengguna terhadap (i), dan $\hat{y}_i$ merupakan nilai prediksi pengguna terhadap item (i).

III. METODE

Pada tahap perancangan sistem ini, sistem dirancang untuk membangun sistem rekomendasi dengan pendekatan *collaborative filtering* menggunakan metode *matrix factorization*, dan melakukan perbandingan model SVD, SVD++, dan NMF.



Gambar 3.1 Desain Sistem

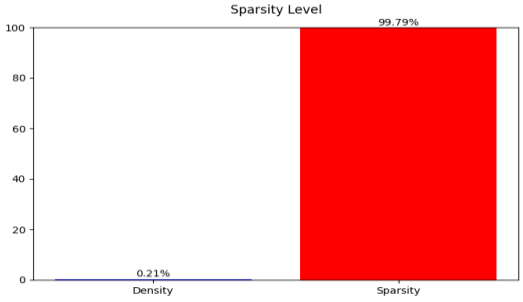
A. Dataset

Dalam konteks penelitian ini, penulis menggunakan dataset *BoardGameGeek*(BGG) dengan berfokus pada *user_ratings.csv* dan dataset dapat di unduh pada website *Kaggle*. Pada dataset ini terdapat *attribute* dan *variable* yang diperlukan pada penelitian ini. Berikut terdiri dari 3 kolom data yaitu *BGGId*, *Rating*, dan *Username*. Dengan memiliki total baris data 18.942.215 baris untuk atribut *rating*, dan baris *BGGId* memiliki 21.925 baris kemudian *Username* memiliki 411.374 baris. Berikut data dapat direpresentasikan pada Tabel 3.1:

Nama Atribut	Jumlah Data
Game_Id	21.925
Rating	18.942.215
User_Id	411.374

Tabel 3.1

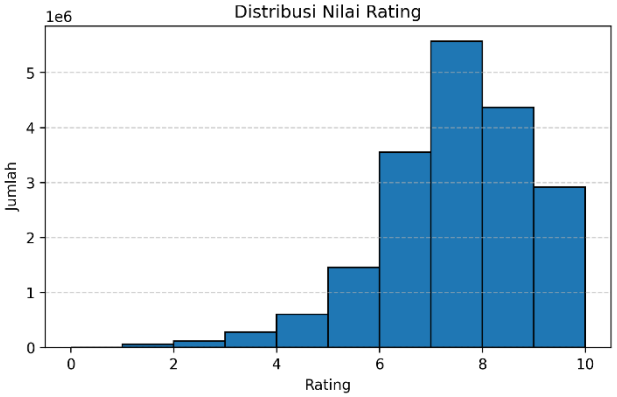
B. Sparsity Level



Gambar 3.2 Sparsity Level

Gambar 3.2 di atas menunjukkan bahwa tingkat *sparsity* data pada matriks rating pengguna dalam dataset memiliki nilai *sparsity* yang sangat tinggi, dimana ini menunjukkan sebagian besar pengguna hanya memberi *rating* pada sebagian kecil data yang tersedia. Berdasarkan pada gambar tersebut, diketahui bahwa hanya sebesar 0.21% pada data dari seluruh interaksi pengguna yang memiliki nilai *rating*, sedangkan 99.79% sisanya merupakan sel kosong atau belum memiliki *rating*.

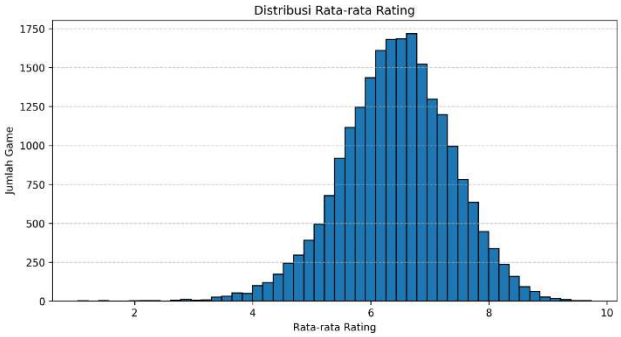
C. Distribusi Nilai Rating



Gambar 3.3 Distribusi Rating Pengguna

Selanjutnya, pada Gambar 3.3 grafik menunjukkan distribusi nilai *rating* yang bisa diberikan oleh pengguna dalam skala 1 hingga 10, terlihat bahwa mayoritas pengguna memberikan *rating* pada rentang 6 hingga 9 dengan frekuensi tertinggi terdapat pada *rating* 8. Sementara itu, *rating* dengan nilai terkecil relatif rendah, hal ini menunjukkan bahwa sebagian besar pengguna cenderung memberikan nilai positif terhadap *game* yang telah dimainkan oleh mereka.

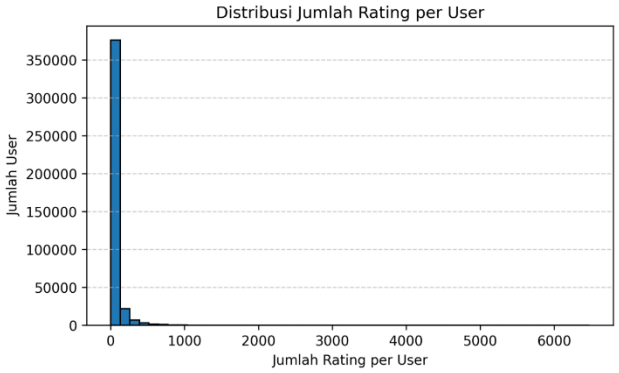
D. Distribusi Rata-Rata Nilai Rating



Gambar 3.4 Nilai Rating Rata-Rata

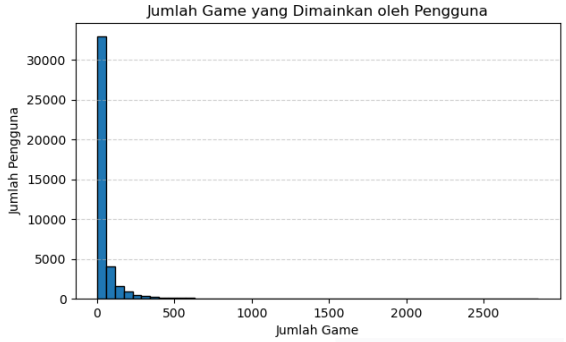
Gambar 3.4, menggambarkan rata-rata penilaian oleh pengguna dengan membentuk pola distribusi normal dengan puncak frekuensi pada perkiraan rentang 6 hingga 7, sedangkan hanya sebagian kecil yang memiliki *rating* dibawah 4. Hal ini dapat disimpulkan bahwa sebagian besar permainan mendapat nilai positif oleh pengguna dengan tingkat kepuasan yang sangat tinggi.

E. Distribusi Jumlah Rating User



Gambar 3.5 Jumlah User Yang Memberi Rating

Berdasarkan pada Gambar 3.5, grafik ini mengidentifikasi pola aktivitas setiap pengguna dalam memberikan *rating*. Dapat dilihat bahwa sebagian besar pengguna sangat pasif dalam memberi penilaian dan sebagian kecil lainnya cukup aktif dalam memberi penilaian terhadap *game* yang dimainkan. Dimana lebih dari 350.000 pengguna merepresentasikan kontribusi jumlah *rating* yang sangat kecil, sedangkan pengguna lainnya berkontribusi dalam memberikan rating sekitar 6000 *rating*.



Gambar 3.6 Distribusi Game

Pada gambar 3.6, pola aktivitas pengguna dalam bermain game menunjukkan bahwa sebagian besar game hanya dimainkan oleh sedikit pengguna dan sebagian pengguna yang aktif.

F. Statistik User



Gambar 3.7 Statistik User

Selanjutnya pada Gambar 3.7, pada grafik ini penulis hanya menggunakan 50 *user* teratas untuk menampilkan grafik distribusi statistik user yang memberikan *rating* terhadap *game*. Sebagian kecil pengguna aktif memberi *rating* dalam jumlah yang signifikan sedangkan mayoritas lainnya bersifat pasif dalam memberikan *rating*. Berdasarkan grafik tersebut menunjukkan pola cenderung menurun yang merepresentasikan bahwa data bersifat sangat jarang (*sparse*) dengan aktivitas yang tidak merata pada setiap user. Pola seperti ini menjadi tantangan utama dalam tahap permodelan karena berpengaruh terhadap pola preferensi dalam sistem rekomendasi.

G. Data Cleaning

Pada tahap ini, dilakukan pembersihan data untuk memastikan bahwa dataset yang digunakan pada penelitian ini dapat digunakan sehingga dapat menghasilkan model sistem rekomendasi yang lebih akurat dan efektif. Proses pembersihan data meliputi penghapusan *misssing value* dan menghapus data duplikat.

H. Data Normalization

Pada tahap ini, dilakukan perubahan nama atribut pada dataset yaitu *Game_Id*, *Rating*, dan *User_Id* agar memiliki struktur data yang konsisten dan sesuai dengan kebutuhan algoritma dalam penggunaan library *surprise*.

Selain itu, normalisasi data digunakan pada data ini untuk menyederhanakan nilai representasi *rating* sehingga dapat meningkatkan kinerja algoritma rekomendasi.

I. Random Sampling

Pada penelitian ini, pemrosesan data melakukan metode *random sampling* berdasarkan interaksi pengguna karena seluruh data pada dataset *BoardGameGeek*(BGG) memiliki 18.942.215 baris data. Penulis melakukan pengurangan data dengan mengambil 10% sampel data dari keseluruhan baris data, langkah ini dilakukan untuk meningkatkan efektivitas model dan mengurangi beban komputasi.

J. Skenario

Pada tahap ini, akan dijelaskan skenario pengujian dan rencana sistem rekomendasi yang telah dirancang. Proses pengujian model pada peneltian ini menggunakan pendekatan *collaborative filtering* dengan metode *matrix factorization*. Untuk mengukur performa model, pengujian dilakukan menggunakan mekanisme *cross-validation* dengan menggunakan teknik *k-fold* sebanyak 5 *fold*. Proses pengujian *cross-validation* dilakukan dengan cara membagi data menjadi data latih (*train data*) dan data uji (*test data*), dimana setiap data dibagi menjadi 5 bagian yang sama besar. Pada setiap pengujian 4 bagian data sebagai data latih (*train data*) dan 1 bagian lainnya digunakan sebagai data uji (*test data*), proses ini dilakukan secara bergantian sejumlah 5 kali hingga seluruh bagian data digunakan sebagai data uji (*test data*).

Ada dua jenis evaluasi yang digunakan dalam penelitian ini untuk mengukur kinerja performa sistem rekomendasi yaitu dengan menghitung metrik evaluasi prediksi seperti *Root Mean Squared Error* (RMSE) dan *Mean Absolute Error* (MAE). Kedua metrik ini digunakan untuk mengukur tingkat kesalahan nilai *rating* dan nilai aktual. Pada penelitian ini dilakukan proses pengujian terhadap ketiga algoritma, setiap algortima diuji dengan menggunakan *5-fold cross validation* dengan variasi parameter *n_factors* (25, 50, 75, 100), kemudian parameter *learning rate* (0.001, 0.003, 0.005) dan *regularization* untuk menghindari overfitting (0.01, 0.03, 0.05). setiap parameter diuji untuk mengetahui seberapa efektif terhadap metrik evaluasi RMSE dan MAE.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini, penulis melakukan proses pencarian parameter dengan menggunakan *hyperparameter tuning* pada ketiga algoritma yaitu SVD, SVD++, dan NMF. Setiap algoritma diuji menggunakan *cross validation* dengan *k-fold* yang berjumlah 5, kemudian melakukan pengujian dengan n_epochs berjumlah 20. Pada penelitian ini juga penulis melakukan evaluasi menggunakan RMSE dan MAE.

Algoritma	n-factors	Fold					Average
		1	2	3	4	5	
SVD	25	1.1823	1.1840	1.1836	1.1847	1.1837	1.1836
	50	**1.1809**	**1.1844**	**1.1832**	**1.1825**	**1.1809**	**1.1823**
	75	1.1813	1.1840	1.1827	1.1824	1.1814	1.1824
	100	1.1811	1.1848	1.1826	1.1842	1.1817	1.1829
SVD++	25	1.1841	1.1881	1.1877	1.1873	1.1865	1.1867
	50	1.1851	1.1875	1.1881	1.1877	1.1861	1.1869
	75	**1.1846**	**1.1880**	**1.1868**	**1.1876**	**1.1846**	**1.1863**
	100	1.1858	1.1876	1.1880	1.1869	1.1851	1.1866
NMF	**25**	**1.3990**	**1.4082**	**1.4062**	**1.3796**	**1.4073**	**1.4001**
	50	3.0010	3.0101	3.0107	3.0015	3.0034	3.0053
	75	3.2275	3.2283	3.2330	3.2280	3.2277	3.2289
	100	3.2464	3.2467	3.2506	3.2463	3.2461	3.2472

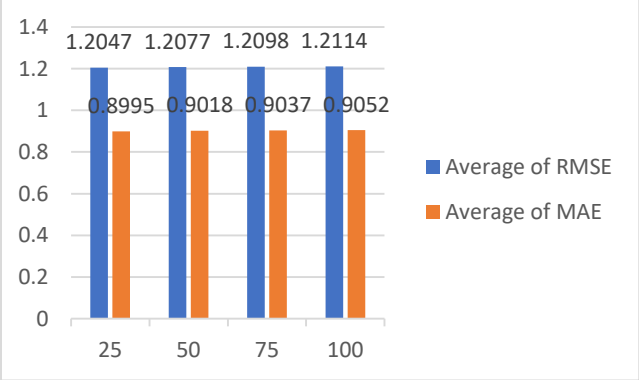
Tabel 4.1 Hasil Performansi Terbaik Algoritma berdasarkan RMSE

Algoritma	n-factors	Fold					Average
		1	2	3	4	5	
SVD	25	0.8819	0.8836	0.8824	0.8836	0.8834	0.8830
	50	**0.8801**	**0.8834**	**0.8816**	**0.8818**	**0.8810**	**0.8815**
	75	0.8808	0.8830	0.8812	0.8819	0.8817	0.8818
	100	0.8809	0.8838	0.8810	0.8828	0.8819	0.8821
SVD++	25	0.8838	0.8872	0.8861	0.8866	0.8864	0.8860
	50	0.8852	0.8871	0.8866	0.8871	0.8859	0.8864
	75	**0.8843**	**0.8875**	**0.8858**	**0.8870**	**0.8847**	**0.8858**
	100	0.8849	0.8869	0.8862	0.8863	0.8852	0.8859
NMF	**25**	**1.1062**	**1.1153**	**1.1129**	**1.0850**	**1.1148**	**1.1068**
	50	2.6705	2.6768	2.2680	2.6704	2.6730	2.5917
	75	2.8533	2.8527	2.8598	2.8534	2.8542	2.8547
	100	2.8654	2.8644	2.8711	2.8652	2.8661	2.8664

Tabel 4.2 Hasil Performansi Terbaik Algoritma berdasarkan MAE

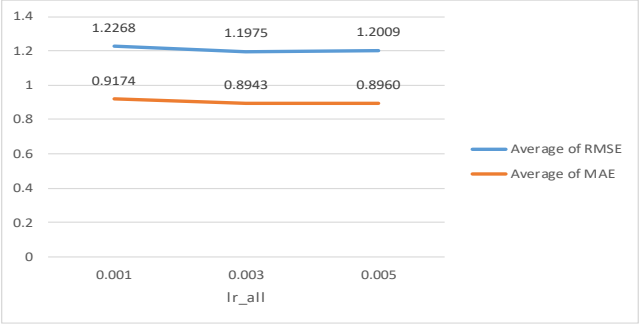
Pada tabel 4.1 dan 4.2, Berdasarkan hasil pengujian, didapatkan hasil performansi terbaik dari 3 algoritma yang digunakan dalam penelitian ini. Pengujian dilakukan dengan mengimplementasikan kombinasi parameter meliputi aspek jumlah faktor laten, parameter *learning rate*, serta parameter regularisasi dengan menggunakan *n-epochs* = 20. Pada algoritma SVD, kinerja terbaik didapatkan pada saat *n_factors* = 50, hasil menunjukkan bahwa kinerja algoritma SVD cenderung stabil pada kombinasi parameter tersebut. Hasil performansi pengujian untuk setiap algoritma. *n-factors* menunjukkan jumlah fitur laten untuk setiap algoritma, sementara itu *K-fold* menunjukkan seberapa banyak jumlah iterasi pada setiap pengujian algoritma. Pada algoritma SVD terlihat bahwa hasil menunjukkan menghasilkan performansi terbaik pada kombinasi parameter dengan *n_factors* = 50 dengan nilai RMSE sebesar 1.1823 dan nilai MAE sebesar 0.8815. Selain itu, SVD++ memiliki pola yang hampir serupa namun lebih kompleks karena algoritma ini menggunakan mekanisme *implicit feedback* sehingga mempengaruhi proses kinerja algoritma. Pada algoritma NMF terjadi peningkatan kinerja algoritma yang signifikan, hal ini karena algoritma menerapkan faktor laten *non negative* yang lebih besar untuk merepresentasikan data *rating* pengguna.

A. Singular Value Decomposition (SVD)



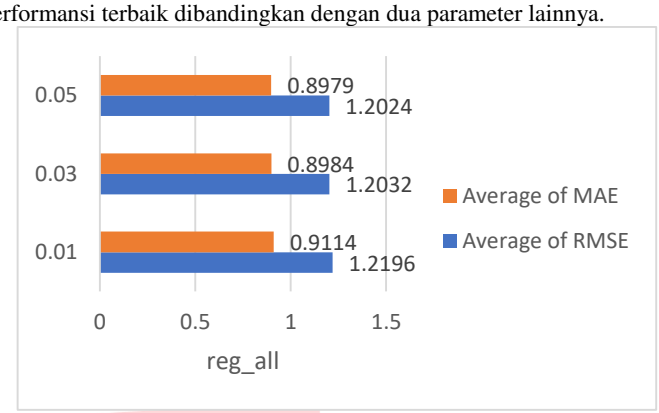
Tabel 4.3 Analisis Parameter SVD

Berdasarkan hasil pengujian, menunjukkan bahwa nilai RMSE dan MAE meningkat dengan dilakukan penambahan *n_factors*. Secara keseluruhan terlihat bahwa semakin besar nilai *n_factors* maka kinerja SVD cenderung menurun, dapat disimpulkan bahwa dengan melakukan penambahan jumlah *n_factors* tidak mempengaruhi akurasi semakin baik.



Tabel 4.4 lr_all

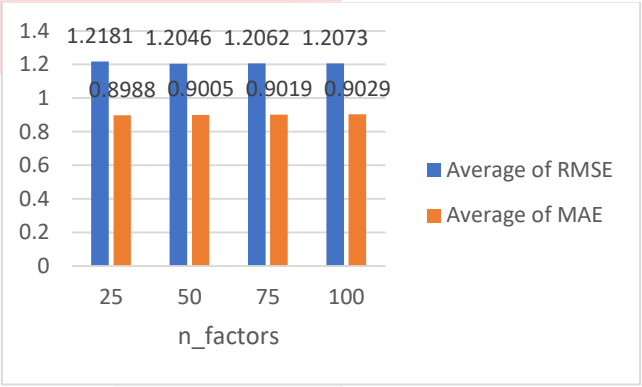
Pada parameter *lr_all*, terlihat bahwa parameter *lr_all* memberikan pengaruh yang cukup stabil *lr_all* = 0.03 menjadi performansi terbaik dibandingkan dengan dua parameter lainnya.



Tabel 4.5 reg_all SVD

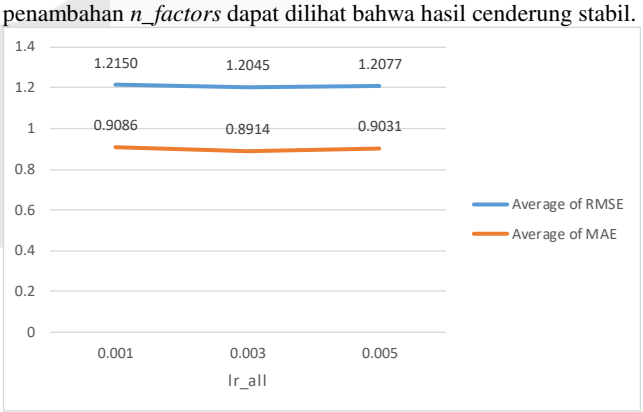
Kemudian, pengujian dilakukan menggunakan parameter *reg_all* untuk menghindari *overfitting* pada model. Hasilnya menunjukkan bahwa parameter *lr_all* = 0.05 merupakan performansi terbaik dari ketiga parameter uji.

B. SVD++



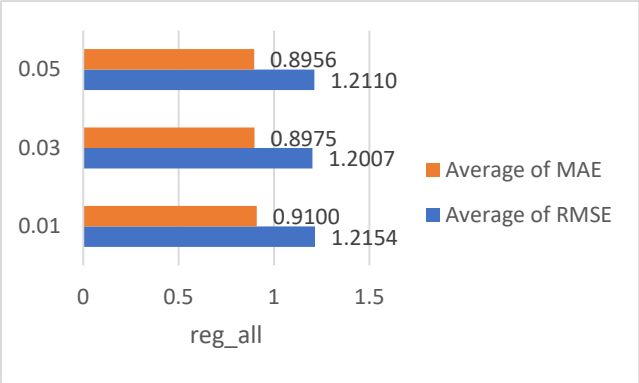
Tabel 4.6 Analisis Parameter SVD++

Berdasarkan hasil pada tabel 4.6, menunjukkan bahwa algoritma SVD++ cenderung serupa dengan algoritma SVD bahkan sedikit mengungguli pada kombinasi parameter tertentu. Dalam pengujian ini terlihat penambahan jumlah *n_factors* berpengaruh terhadap nilai RMSE dan MAE cenderung memburuk diawal namun seiring penambahan *n_factors* dapat dilihat bahwa hasil cenderung stabil.



Tabel 4.7 lr_all SVD++

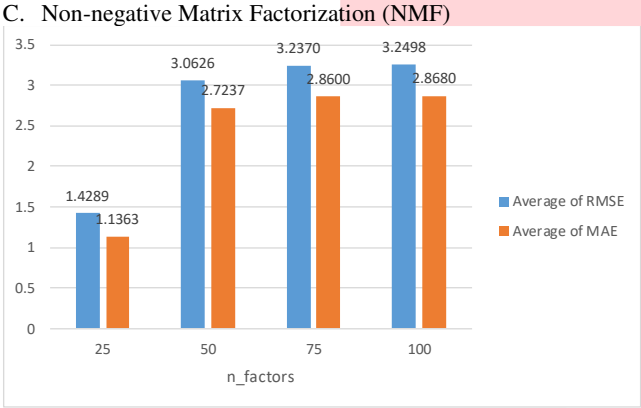
Sementara itu, pada parameter lr_all, nilai pada 0.003 menghasilkan nilai yang stabil dibandingkan dengan nilai lr_all = 0.001 yang cenderung memburuk diawal dan pada 0.05 memberikan hasil yang kurang stabil.



Tabel 4.8 reg_all SVD++

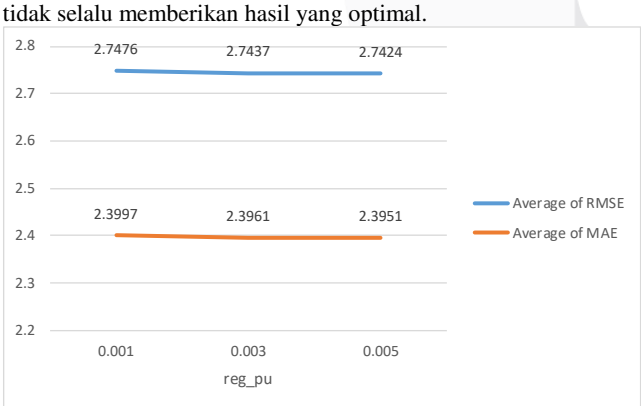
Berdasarkan hasil pada tabel 4.8, parameter *reg_all* mempengaruhi kinerja untuk menghidari *overfitting* pada model. Kombinasi pengujian parameter terlihat bahwa nilai 0.03 menghasilkan performansi yang paling stabil pada seluruh parameter pengujian.

C. Non-negative Matrix Factorization (NMF)



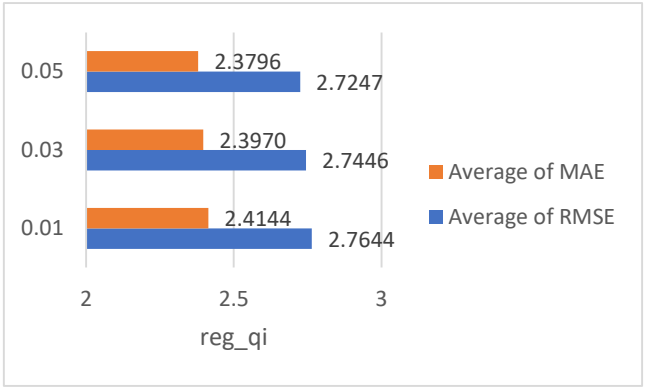
Tabel 4.9 Analisis Parameter NMF

Berdasarkan hasil pengujian algoritma NMF, menunjukkan bahwa hasil mengalami peningkatan yang cukup signifikan pada kombinasi parameter yang digunakan dibandingkan dengan kedua algoritma sebelumnya. Pada penambahan jumlah *n_factors* algoritma NMF menunjukkan pola yang cenderung semakin memburuk, dapat disimpulkan bahwa peningkatan jumlah *n_factors* tidak selalu memberikan hasil yang optimal.



Tabel 4.10 reg_pu NMF

Sementara itu, pada pengujian menggunakan parameter *reg_pu* cenderung memberikan hasil yang stabil pada setiap pengujian. Dapat dilihat bahwa hasil nilai RMSE dan MAE tidak mengalami penurunan maupun peningkatan secara signifikan pada setiap parameter uji.



Tabel 4.11 reg_qi NMF

Kemudian, parameter *reg_qi* menunjukkan hasil yang hampir serupa, terlihat pada nilai RMSE dan MAE yang cenderung stabil.

D. Top-N Recommendation

Peringkat	Game Id	Item	Prediksi Rating
1	342942	Ark Nova	9.40
2	63170	1817	9.10
3	55690	Kingdom Death: Monster	9.00
4	299659	Clash of Cultures: Monumental Edition	8.88
5	341169	Great Western Trail (Second Edition)	8.86
6	323947	Middara: Unintentional Malum â€“ Act 1 Version 1.1	8.83
7	246900	Eclipse: Second Dawn for the Galaxy	8.82
8	68820	Enemy Action: Ardennes	8.79
9	324856	The Crew: Mission Deep Sea	8.71
10	286749	Hansa Teutonica: Big Box	8.69

Tabel 4.12 Top-N Recommendation SVD

Pada tabel 4.12, bertujuan untuk memberikan hasil daftar item dengan menggunakan prediksi item tertinggi yang paling relevan dengan pengguna, data prediksi *rating* diurutkan dari yang paling tinggi hingga ke prediksi yang paling rendah untuk setiap pengguna.

Peringkat	Game Id	Item	Prediksi Rating
1	342942	Ark Nova	9.26
2	264220	Tainted Grail: The Fall of Avalon	9.10
3	323947	Middara: Unintentional Malum â€“ Act 1 Version 1.1	9.05
4	278292	Anachrony: Infinity Box	9.04
5	299659	Clash of Culture: Monumental Edition	9.02
6	246900	Eclipse: Second Dawn for the Galaxy	8.97
7	291572	Oath: Chronicles of Empire and Exile	8.83
8	63170	1817	8.80
9	169786	Scythe	8.79
10	324856	The Crew: Mission Deep Sea	8.75

Tabel 4.13 Top-N Recommendation SVD++

Berdasarkan tabel 4.13, menunjukkan hasil rekomendasi permainan untuk setiap pengguna, sistem rekomendasi ini dihasilkan setelah pengujian prediksi *rating* terhadap item kemudian diurutkan berdasarkan nilai prediksi tertinggi. Nilai dengan prediksi *rating* tertinggi dapat disimpulkan bahwa permainan tersebut memiliki kesesuaian terhadap preferensi pengguna berdasarkan pola interaksi sebelumnya.

Peringkat	Game Id	Item	Prediksi Rating
1	291951	The Everdeck	10.00
2	165694	The Cones of Dunshire	9.33
3	209322	Sabbat Magica	9.19
4	21730	Delta	9.16
5	20623	Wallenstein Big Box	8.74
6	17821	Connection Games	8.71
7	182982	Horse & Musket: Dawn of an Era	8.69
8	345976	System Gateway	8.67
9	103039	Beyond Waterloo	8.59
10	26138	Dungeon Twister Collectors Box	8.53

Tabel 4.14 Top-N Recommendation NMF

Tabel 4.14, dengan menggunakan algoritma NMF mendapatkan nilai rekomendasi yang tinggi ini mengindikasikan bahwa permainan yang direkomendasikan pada setiap pengguna memiliki tingkat relevansi yang sesuai. Item yang memiliki prediksi *rating* tinggi akan direkomendasikan kepada pengguna sebagai preferensi permainan yang disukai oleh setiap pengguna.

V. KESIMPULAN

Berdasarkan seluruh hasi pengujian pada penelitian ini, menghasilkan bahwa algoritma SVD memberikan hasil dengan performansi terbaik dibandingkan dengan 2 algoritma lain, yaitu SVD++ dan NMF. Hal ini dapat dilihat pada hasil dari nilai RMSE dan MAE bahwa algoritma SVD memiliki hasil yang optimal dengan nilai RMSE 1.1809 dan MAE 0.8801. Sementara itu, SVD++ hasilnya cenderung serupa bahkan dalam kondisi tertentu bisa lebih mengungguli namun memerlukan waktu komputasi yang cukup lama karena algoritma tersebut menggunakan *implicit feedback* yang memanfaatkan interaksi pengguna. Kemudian, hasil pengujian dari algoritma NMF cenderung menghasilkan nilai *error* yang lebih tinggi dibanding kedua algoritma lain, hal ini karena NMF menggunakan faktor *non-negative* sehingga kurang optimal dalam memprediksi preferensi pengguna. Penelitian ini masih dapat dikembangkan dengan cakupan yang lebih luas di penelitian selanjutnya. Dengan menggunakan metode *Deep Learning* sistem rekomendasi ini berpotensi menghasilkan sistem yang lebih kompleks dan optimal. Kemudian, metode evaluasi dapat dikembangkan menjadi lebih luas tidak hanya menggunakan RMSE dan MAE, dengan menambahkan metrik evaluasi lain seperti, *Precision@K*, *Recall@K*, *F1-score*, dan *NDCG*.

REFERENSI

[1] Ion, M., Sacharidis, D., & Werthner, H. (2020). Designing a Recommender System for Board Games. *Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing*, 1481–1484.

[2] Kim, J., Wi, J., Jang, S., & Kim, Y. (2020). Sequential Recommendations on Board-Game Platforms. *Symmetry*, *12*(2), 210.

[3] Aciar, S. V., Fabregat, R., Jové, T., & Aciar, G. (2021). Enhancing Recommender System with Collaborative Filtering and User Experiences Filtering. *Applied Sciences*, *11*(24), 11890, doi.org/10.3390/app112411890

[4] J. Zalewski, M. Ganzha and M. Paprzycki, "Recommender system for board games," *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 2019, pp. 249-254, doi: 10.1109/ICSTCC.2019.8885455.

[5] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, *52*(1), 1–38.

[6] Mohammed Fadhel Aljunid, Manjaiah D.H., "An Efficient Deep Learning Approach for Collaborative Filtering Recommender System," Procedia Computer Science, vol. 171, pp. 829–836, 2020.

[7] Ko, H., Lee, S., Park, Y., & Choi, A. (2022). A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*, *11*(1), 141, doi.org/10.3390/electronics11010141

[8] M. Fu, H. Qu, Z. Yi, L. Lu and Y. Liu, "A Novel Deep Learning-Based Collaborative Filtering Model for Recommendation System," in *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084-1096, March 2019, doi: 10.1109/TCYB.2018.2795041.

[9] Koren, Y., Rendle, S., Bell, R. (2022). Advances in Collaborative Filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds) Recommender Systems Handbook. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-2197-4_3

[10] I. Saifudin dan T. Widiyaningtyas, "Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets," IEEE Access, vol. 12, pp. 19827–19837, 2024. doi: 10.1109/ACCESS.2024.3359274.

[11] Duan, R., & Jiang, C. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decision Support Systems*, *158*, 113748. https://doi.org/10.1016/j.dss.2022.113748

[12] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," in *IEEE Access*, vol. 6, pp. 69009-69022, 2018, doi: 10.1109/ACCESS.2018.2880197.

[13] Luekhong, P., & Chansanam, W. (2023). *Advancing Book Recommendation Systems: A Comparative Analysis of Collaborative Filtering and Matrix Factorization Algorithms. Journal of Information Systems Engineering and Management.*

[14] Duan, R., Jiang, C., & Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. ***Decision Support Systems***, *156*, 113748. https://doi.org/10.1016/j.dss.2022.113748

[15] R. Barathy and P. Chitra, "Applying Matrix Factorization In Collaborative Filtering Recommender Systems," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020, pp. 635-639, doi: 10.1109/ICACCS48705.2020.9074227.

[16] S. Jiang, J. Li and W. Zhou, "An Application of SVD++ Method in Collaborative Filtering," *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, 2020, pp. 192-197, doi: 10.1109/ICCWAMTIP51612.2020.9317347.

[17] Isinkaye, F. O. (2021). Matrix Factorization in Recommender Systems: Algorithms, Applications, and Peculiar Challenges. *IETE Journal of Research, 69*(9), 6087–6100.

[18] Wang, S., Sun, G., & Li, Y. (2020). SVD++ Recommendation Algorithm Based on Backtracking. *Information*, *11*(7), 369. https://doi.org/10.3390/info11070369

[19] N. Liu and J. Zhao, "Recommendation System Based on Deep Sentiment Analysis and Matrix Factorization," in *IEEE Access*, vol. 11, pp. 16994-17001, 2023, doi: 10.1109/ACCESS.2023.3246060.

[20] Jayalakshmi, S., Ganesh, N., Čep, R., & Senthil Murugan, J. (2022). Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. *Sensors*, *22*(13), 4904. https://doi.org/10.3390/s22134904

[21] Raghuwanshi, S.K., Pateriya, R.K. (2019). Collaborative Filtering Techniques in Recommendation Systems. In: Shukla, R.K., Agrawal, J., Sharma, S., Singh Tomer, G. (eds) Data, Engineering and Applications. Springer, Singapore. https://doi.org/10.1007/978-981-13-6347-4_2