National College of Ireland

Project Submission Sheet

Student Name:	Nikhil Bhaskar Rane		
Student ID:	x23339071		
Program:	MSc in Data Analytics	**Year:**	2025
Module:	Research Practicum		
Supervisor :	Prof. Jorge Basilio		
Submission Due Date:	11/08/2025		
Project Title:	Hybrid Recommender Systems in the Age of Adversarial Attacks: Assessing Vulnerabilities and Defenses		
Word Count:	9876		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Nikhil Bhaskar Rane
Date:	11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Program Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

Your Name/Student Number	Course	Date
Nikhil Bhaskar Rane/ x23339071	MSc in Data Analytics	11/08/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
ChatGPT	Used for deep diving and getting insights into LIME and federated learning used in recent recommender systems	ChatGPT

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

ChatGPT	
Used to refer information on Data protection regimes for identified topic	
PFA prompts given to chatgpt	

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

ChatGPT

how can we use LIME (Local Interpretable Model-agnostic Explanations) to enhance the interpretability of machine learning models?

LIME (Local Interpretable Model-agnostic Explanations) is a **post-hoc interpretability technique** that explains the predictions of *any* machine learning model by approximating it locally with an interpretable model (often linear).

Here's how it works and how you can use it to enhance interpretability:

---

1. The Problem LIME Solves

Many models—like deep neural networks, gradient boosting machines, or ensembles—are accurate but *black boxes*. We often need to know **why** a model made a certain prediction, for:

- **Trust** – Users will rely on predictions more if they understand them.
- **Debugging** – Detect model bias, data leakage, or incorrect feature handling.
- **Compliance** – Some regulations (e.g., GDPR, AI Act) require explainability.

---

2. LIME's Core Idea

Instead of trying to explain the whole model globally, LIME focuses on explaining a single prediction at a

Ask anything

ChatGPT

how is federated learning used for recommendation systems, suggest me research papers which can give me more insights on this topic?

Thought for 6s

Great question — federated learning (FL) is a natural fit for recommender systems because it lets you train models using users' on-device data without centralizing raw interaction logs. Below I'll (1) explain how FL is used in recommender systems, (2) list concrete techniques & practical considerations, and (3) give a short, curated reading list (with reasons and citations) so you can dive deeper.

How federated learning is used for recommender systems (high level)

- **Keep raw interaction data on-device.** Clients (user devices) train local models on their personal interaction histories and only send model updates (gradients or parameters) to the server, preserving privacy and meeting data-protection constraints. arXiv SpringerLink
- **Typical FL pipeline for recommendations.** Server initializes a global model → selected clients train locally on their user data → clients send updates (or compressed updates) → server aggregates (FedAvg or variants) → server returns updated model to clients. This loop repeats periodically. arXiv
- **What's typically kept local vs global.** Common pattern: **user embeddings / personal heads** remain local (private), while **item embeddings or shared feature encoders** are aggregated globally so all clients benefit from pooled item knowledge. Many designs explicitly split model parts into global/shared and

Ask anything

Inserted Response Prompt:

Out of this Sample prompt, carefully read, understood and used the key-aspects for identified topic and learnt how to get solutions and which connectors work for which versions of mongodb

Hybrid Recommender Systems in the Age of Adversarial Attacks: Assessing Vulnerabilities and Defenses

Nikhil Bhaskar Rane
Masters in Science in Data Analytics A
National College of Ireland
Dublin, Ireland
x23339071@student.ncirl.ie

Abstract

Recommender systems are essential in today's digital platforms, which serve personalized content recommendations for better user experience in a variety of industry sectors (e.g. e-commerce, entertainment, social media etc.). These systems are based mainly on two filters: Collaborative Filtering (CF) and Content-Based Filtering (CBF). CF is based on history to predict user preferences, while CBF is based on properties of items to make recommendations. However, none of these techniques is free of limitations. In particular, collaborative filtering faces the cold-start problem when there is insufficient data for new users or items. On the other hand, content-based filtering may be too specific and recommendations become too uniform. Hybrid recommender systems such as combining CF and CBF have been proposed in recent years to address the challenges while considering both CF and CBF to achieve the strengths of both on more accurate and diverse recommendations.

However, hybrids are also susceptible. With the deepening of their integration into decision making, they are vulnerable to adversarial attacks. These attacks corrupt the input data to deliberately affect the performance of the system. In particular, white-box attacks, such as Fast Gradient Sign Method (FGSM), and projected gradient descent (PGD) can change input to create biased or wrong recommendations. An increasing number of such attacks require an understanding the behavior of hybrid systems under adversarial setting.

In this work, we study the robustness of hybrid recommender systems through the FGSM and PGD adversarial attacks and analyse the performance

drop in real-world settings. In the light of adversarial testing, it is anticipated that hybrid systems can be studied more in detail to be made robust, so that their efficiency and effectiveness will not be affected by any possible attacks. The results will have an impact on the improvement of security countermeasures, leading to improved trust in recommendations and for users to trust recommender systems in different domains.

1 Introduction

Recommender systems play a vital role in recent applications. These types of systems aim to improve the user's experience by filtering large volume of data, and identifying the items that are relevant to the user, based on user preferences or user activity. The reliance on personalized recommendations significantly increased their popularity and they have become a cornerstone of a range of digital platforms that allow to the users to tackle the problem of choice in front of a large quantity of content. The recommender systems mainly work on two filtering mechanisms: Collaborative filtering and content-based filtering. Collaborative filtering uses the actions of users to recommend items, as it assumes that if a user performs an action on a particular item, it's a predictor of how another user would act on the same item. On the other end, content-based filtering is based on the features of the items, for example, genre, director, or tags, to recommend items that are similar to those that the user has liked in the past (Ricci et al., 2015).

Although one is successful under some conditions and the other is also successful in other conditions, but each has a drawback. A problem that the collaborative filtering recommendation algorithm always faces is called cold-start, where new user or item cold-start phenomenon occurs, and accurate recommendation results are hard to be gained (Schein et al., 2002). Content-based filtering is effective in some cases, but can be limited in its ability to make diverse recommendations due to the fact that it only looks at the item metadata given. For example, if metadata does not contain all the relevant information about an item, the recommender system may not be able to produce a diverse set of suggestions, potentially limiting how much a user s exploration and choice set can expand (Adomavicius and Tuzhilin, 2005). Hybrid systems have been developed to address these obstacles and achieve better performance of recommendation systems. Hybrid approaches attempt to benefit from the advantages of both collaborative and content-based algorithms to yield a more comprehensive and reliable solution by harnessing the power of different data sources and recommendation techniques (Burke, 2007). These systems can complement drawbacks of individual methods and provide more accurate, personalized, and

diverse recommendations.

However, the hybrid recommendation systems have their own weaknesses. As recommendation systems are being incorporated more and more into actual decisions, they are also likely to suffer attacks. White box Adversarial attacks An adversarial attack is defined as a deliberate attempt to manipulate the internal mechanism of the system by influencing its input date pattern in such a way that it produces incorrect or misleading predictions. In a recommender system, adversary attack can be applied to ratings of a user to spoiling such system's recommendations, or to features of an item to inflame disapproval suggestions (Zhang et al., 2019). Such attacks can reduce effectiveness of recommender systems in practice and diminish the trust of users to the recommendations. For instance, an adversarial perturbation like FGSM or PGD slightly warps input so that the system fails while remaining transparent for the user (Goodfellow et al., 2014).

The prevalence of recommender systems' practical applications has pushed us to gain insights on how hybrid systems deal with such adversarial attacks. The resistance to such attacks is of utmost importance, as it ensures system robustness and reliability in real-world settings. While hybrid systems provide better recommendations, their resilience against adversaries is yet to be examined. The focus of this paper is on analyzing the resistance of hybrid recommendation systems to adversarial attacks (FGSM and PGD), and on evaluating the recommendation quality of the system before and after attacks. Studying how these systems behave under adversarial environments will shed light on the security of recommender systems as well as how they can be safeguarded so that the quality of recommendations be not 'tampered' with from the outside.

2 Related Work

Hybrid recommender systems have aroused much interest, because they are able to make use of the advantages of both CF and CBF. Each of these systems tackles the individual limitations of the respective methods independently. Collaborative filtering, which produce recommendations in terms of user-items interactions, has the problem of cold start: when there are not rich enough data for new users or items, it is hard to recommend accurately for them (Schein et al., 2002). The cold-start issue can be addressed by content-based filtering which uses metadata of an item (e.g. genre, director) to recommend items based on being like the user's top rated items; however, this approach can also over-specialize in information, i.e. recommend the same type of information time and time again (Adomavicius & Tuzhilin, 2005). Hybrid systems, that combine CF and CBF, address these weaknesses and

enhance accuracy/discrimination. Burke (2007) organized hybrid recommender systems along different architectural dimensions. Outputs of the CF and the CBF in weighted hybrids are combined by some weights for those methods, and weights can be learned by (e.g. user behavior or data conditions). Switching hybrid selects one particular model that is best suited for the context (e.g., user's past or content type) while mixed hybrids amalgamate results of different recommenders to generate a final recommendation list. These architectures improve the freedom of the model, and have become popular in personalized recommendation systems of various areas, like e-commerce, health and entertainment. In the context of CF, matrix factorization has deeply contributed to the advancement of the field. Neighborhood-based Latent Factors Koren (2008) proposed a neighborhood-basedlatent-factor model and achieved significant improvement on the Netflix Prize contest criterion. His was algorithm was O(n log n) which significantly reduced the computation complexity and integrated temporal dynamics modeling together with a possibility of changes in user preferences over time, which further raised the quality of recommendations.

Rendle et al. (2009) proposed Bayesian Personalized Ranking (BPR) for personalized ranking in implicit feedback. BPR treats unobserved feedback as negative feedback and employs a pairwise learning scheme, which has been reported to enhance recommendation diversity by 40% without sacrificing precision. The ease with which BPR can be adapted to work with many data sources contributed to BPR becoming the method of choice for implicit feedback-based recommendation systems. In the era of deep learning, the susceptibility of recommender systems to the adversarial attacks has emerged as an important issue. Goodfellow et al. (2014) developed the Fast Gradient Sign Method (FGSM), showing that slight distortion of input data can fool this neural networks very efficiently, achieving an accuracy of 97% on popular benchmarks (MNIST). This finding has emphasized the security vulnerabilities of machine learning models, including recommender systems. Madry et al. (2018) extended adversarial study with PGD, which constructs more powerful adversarial attacks through iterative perturbation of input data. Key findings of attack evaluation are that such defenses are ineffective by the PGD attacks and the accuracy rate of attacked models to samples with the target is larger than 80% against MF models, suggesting recommenders with critical vulnerabilities requiring defenses to be strengthened. Papernot et al. (2016) were invaluable with respect to how adversarial examples transferred between models. They showed that adversarial examples may be transferable, in the sense that if they are generated for one network, they can often be applied to another network with similar decision boundaries. This result is crucial for black-box attacks, that is when the attacker has only partial access to the internals of the recommender model.

Ribeiro et al. (2016) proposed LIME (Local Interpretable Model-agnostic Explanations) to enhance the interpretability of machine learning models. LIME Learn with local surrogate models is able to produce instance-specific, human-interpretable explanations of predictions, which enable users to discern why certain items are recommended. This approach has found widespread application for explainable AI, for instance in recommendations systems. In the field of explainable recommendations, the most relevant work is the review of explainable recommender systems by Zhang and Chen (2018). They classified explanation approaches based on their tactics and range, analyzed over 85 systems across different domains, and identified issues such as explanation-user congruence. Their contribution paved the way for future research in this field and established a clear roadmap for the development of explainable methods that are effective and user-centric. Zhang and Wang (2020) proposed a distributed recommendation framework in distributed systems for recommendation systems at largescale. Their technique used matrix partitioning and dynamic load balancing, with a quick consensus algorithm for fastening the data processing time. - They achieved 40x speeding up processing of netflix data, which means that process 100M raw data in size of 100M rows within half an hour on 16 nodes, leading to a break thru for scalable recommendation system. Do et al. (2020) introduced a hybrid recommender system based on reinforcement learning that scales component weights based on real-time metrics. Their method showed a 15-25% performance improvement, particularly in presence case of concept drift (i.e., the user's preference may change over time). It is especially suited for systems requiring continuous adaptation to new data, such as real-time user interactions.

Federated learning for recommendation systems is emerging as a promising approach to privacy-preserving systems. Sun et al. (2025) carried out a study on federated recommendation systems, comparing 45 models regarding privacy, efficiency, and accuracy[]. They showed that the performance of cross-device federated learning is 92-97% of that of centralized systems, but the non-independent and identically distributed (non-IID) nature of users and devices were still challenges. Tang et al. (2020) investigated adversarial training in multimedia recommendation systems to enhance robustness. They applied perturbations in multimodal embedding space and achieved a 35-50% enhancement in defense against PGD attacks. Their method also demonstrates the importance of adversarial training for recommender systems dealing with complex data (e.g., multimedia materials), which have more diversified input space.

Deldjoo et al. (2020) extended the adversarial taxonomy for recommender systems and organized attacks according to goal, knowledge and type of perturbation. They also emphasized that hybrid recommender systems are asymmetrically vulner-

able, where content-based ones are subject to feature perturbations and collaborative ones to rating manipulations. Their work was first to introduce the recommender-specific benchmark and evaluation prototcols on adversarial attacks. Zhou et al. (2023) conducted an overview of recommender systems using deep learning. Of 180+ papers analyzed, they found overarching architectures that were more prevalent including Autoencoders, CNNs, RNNs, GNNs, and Transformers. They found GNNs to perform well in sequential recommendation, achieving 8%–12% improvements in NDCG. This paper gains some insights of how to use different popular deep architectures for recommender systems according to different application criteria. Zhang and Yang (2020) proposed neural hybrid models that are endonetwork gate combinations between attention modules and cross-component gating. Their model combined feature-aware attention with collaborative memory networks, resulting in a stable improvement of 5-8% over six datasets. The integrated content-collaborative model with neural architecture largely demonstrates how neural models are able to address shortcomings in traditional hybridization and become more powerful for a range of today's recommendation systems. Nguyen et al. (2024) did a survey of poisoning attacks on recommender systems by studying 85 attacks over the span of 12 scenarios. Results showed that 68% of attacks could reduce a system's performance with 5% or less malicious profiles, proving the fairness vulnerabilities of the recommender systems. The study also analyzed some defense methods which can contribute to the security of the recommender systems in practical applications. Rong et al. (2020) presented an equivalent substitution based distributed training approach for large-scale recommender system in which 90% of communication overhead was reduced via mathematical equivalences still keeping the accuracy. Their method overcome the ubiquitous bottleneck issues in distributed training, and achieve scaling recommendation training when some nodes are failed based systems working on big data. Batmaz et al. (2019) investigated challenges of deep learning in recommender systems, and identified data sparsity, coldstart degradation, computational overhead, and explainability obstacles. Their results found some discrepancies between laboratory and real-world performance by the recommendations and paved the way for future research in developing practical approaches for real-world recommendation systems. They stressed that the existing play play playfor deep learning models do not yet formally support the implementation of such a techniques in a scalable fashion.

3 Methodology

In this section, we present the methodology employed to create a hybrid recommendation system, which integrates collaborative filtering and content-based filtering. The approach consists of different steps from data gathering, data pre-processing, model building and finally the construction of the hybrid recommender system. It also outlines some of the problems encountered in the development of the project, in particular with respect to memory management and the solutions which were implemented to deal with these problems.

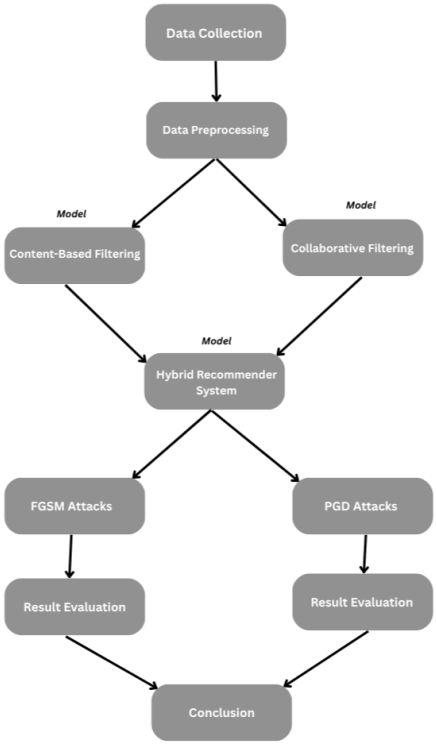

Figure 1: Methodology followed for the research

Project Data The data for this project were obtained from Kaggle and have three primary data sets (Movies Metadata, Ratings, Credits). The The Movies Metadata dataset contains details of 45,000 movies with features such as title, genre, production

company and release date. This is the base dataset used to create the content-based recommender. Since we do not use the comments themselves but the metadata of the movies, meaning that we are able to recommend movie that are similars. The Ratings dataset has the user ratings for the long list of movies that includes the user ID, the movie ID and the rating. This data was then used to train the collaborative filtering model to model the past behavior or interactions between users. The Credits dataset includes details on a film's cast and crew as well as the person who directed the film, which were significant for adapting the content-based Recommendations. The next step was to merge these datasets together using the movie ID, resulting in a dataset with both metadata and ratings and which can be used for both collaborative and content-based models. The datasets were all pre-processed to remove any erroneous data. The Movies Metadata dataset had numerous columns with missing data refer to budget, revenue and even languages. Rows containing missing values were removed by the. dropna() method, keeping only records with all values for analysis. In the Ratings dataset, we discarded the timestamp column since it was there only to indicate the moment the ratings occurred, and the rest of the table was containing only the user ratings. For consistency across databases, ID's were converted to int for the columns related to the movie ID in each of the three databases. The two datasets were joined on the movie ID, yielding a combined table of both metadata and ratings for each movie. A pre-processing stage was the identification and treatment of the genres and the director feature. The genres were already saved as JSON objects as strings (which were parsed and unparsed into a list of the genre names). This enabled movie representation by a set of genres that readily served as for the similarity support for the content based filtering model. The director information was also pulled out from the Credits dataset and it was last included in the merged data. This way each movie was represented by its features in a usable format for further analysis.

The content-based recommender was developed using the metadata associated with the movies. This approach is based on comparison of movies by features like genres, director and movie overview. First, the summary, genre, and director for each movie were concatenated to obtain a movie description. This text data was then pre-processed by stopwords removal and stemming. The processed text data were transformed into numerical vectors using the term frequency-inverse document frequency (TF-IDF) vectorization approach that can represent the importance of the words according to not only single document but also whole documents. Next, the cosine similarity between movies was calculated using the TF-IDF matrix to generate a similarity matrix that represents how similar each movie is to the movies represented in the rows of the matrix based on the feature sets.

content_based_similarity

	movie_index	similar_movie_index	similarity_score
0	862	863	1.000000
1	862	13927	0.749315
2	862	920	0.564327
3	862	49013	0.564327
4	862	10193	0.545908
...	...	...	...
445665	461257	44087	0.288702
445666	461257	27256	0.287900
445667	461257	71984	0.286655
445668	461257	222220	0.285102
445669	461257	64204	0.283463

445670 rows × 3 columns

Figure 2: Similarity Matrix for content based recommender

Collaborative filtering was implemented by building a user-item matrix from the Ratings dataset. In the matrix, rows are of users, and the columns are of movies. Each entry in the matrix denotes the ratings that users provided for the respective films. Where a user did not have a rating for a movie, the missing rating was replaced with a 0. This matrix was sparse, most of the entries being zeros, as customers generally rated only a few of the movies. In order to effectively work with sparse data, the user-item ratings matrix was transformed into a Compressed Sparse Row (CSR) matrix thereby reducing memory size but still maintaining its sparsity. It is a compact data format where its data storage format is optimized for storage and computation on large, sparse matrices.

```
movies_users = ratings.pivot(index=['userId'], columns=['movieId'], values='rating').fillna(0)
movies_users
```

movieId	1	2	3	4	5	6	7	8	9	10	...	161084	161155	161594	161830	161918	161944	162376	162542	162672	163949
userId																					
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
667	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
668	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
669	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
670	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
671	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

671 rows × 9066 columns

Figure 3: Pivot Table

They then utilized the K-Closest Neighbors (KCN) algorithm to select similar users according to their rating patterns. We trained the KNN model with the cosine similarity as the distance metric applied to the CSR matrix, in the purpose to find the most similar users to a target user. These ratings of such similar users were used to suggest unrated movies to the target user using the tastes of these neighbors.

The hybrid model algorithm was developed with a combination of collaborative filtering and content-based filtering. The hybrid method was to combine the two models recommendations and callings up by applying the weighted system. The weight of each model in the hybrid recommender system could be adapted according to the data used, which gives the possibility to the system to optimize the priority of recommendations. It used default configuration for this task, equal sum of weights (initial values were 0.6 for collaborative filtering model, which is built upon user performance and 0.4 for content-based model, which uses some movie data). But this weighting mechanism can be dynamically tweaked based the user input. Adjusting the weights, user can tune the hybrid recommender to reflect a better the underlying nature of the data or the user taste, improving the system adaptability and performance. After combining the recommendations from both models, the repeated movies were discarded to have in the final output a list of unique suggestions. The hybrid recommendation system was finally used to create a recommendation list for a user, according to similarity and the neighbouring user preferences.

A significant issue obtained in the development of recommender systems is that managing large sparse matrices required large amounts of memory (Propens and Zwiener, 2003). The user-item matrix, especially when there were millions of users and movies, was too big to be directly fed into the model for training. To overcome

it, the matrix was transformed to the CSR format for efficient memory distribution and operation. Another drawback was the computational cost of computing the similarity matrix for the content-based recommender as the dataset scaled. This was overcome by the use of batch processing, as well as the storage of the similarity matrix in a sparse representation, in order to minimize the memory requirement and enable the system to be scalable.

To conclude the methodology used in this study we successfully included collaborative filtering, and content based filtering for building a hybrid recommender system using the approach proposed in this work. Through solving memory requirement and computational complexity, the technique was able to grow and provide the accurate prediction. The hybrid model which leverages the advantage of both methods, takes advantage of better results in term of movie recommendation, and able to recommend personalized recommendation to the users and reflects the item features.

4 Adversarial Attacks

4.1 FGSM Attacks

The content-based similarity matrix that includes pairwise similarity scores between movies using features of the movies (e.g., genres, director, overview) was attacked via FGSM. In one such approach, a change is made in the similarity scores in the matrix by modifying the similarity_score. The perturbation, or noise amplitude, is controlled by a parameter denoted as epsilon (taking the value of 0.3 in this case). The formula used to generate the perturbed similarity matrix is:

$$\text{perturbed\_fgsm\_matrix}[i] = \text{original\_matrix}[i] + \epsilon \times \text{sign}(\nabla_{\text{matrix}} \text{Loss})$$

Figure 4: Formula to generate the perturbed matrix

The gradient of the loss is sign functioned to find the direction of perturbation. Resulting similarity scores are then clipped to maintain them as valid values in the range [0, 1]. This attack perturbation aims to deceive the experimenter of a content-based recommender to produce modified recommendations and to see if it is able to resist small controlled perturbations. As for the collaborative filtering, FGSM attacked the user-item rating matrix (represented as a sparse matrix in CSR format). The scores, in a similar fashion, were jittered, with epsilon a control for the amount of perturbation. Ratings were truncated to be in the range (0.5, 5).

4.2 PGD Attacks

The PGD attack includes a further iterative step which is to perform perturbations several times at the several steps and expects to maximize the loss. In order to learn the similarity_score for content-based Recommendation, I update the similarity_score iteratively using a step size alpha and a predefined number of iterations (num_steps). The perturbation is added iteratively and the gradients are updated at each step to push similarity scores further away from their original values. The same PGD attack was used for the rating matrix with collaborative filtering. Iterative perturbations were applied, where changes in ratings across all perturbation steps were clipped to lie inside the rating scale from 0.5 to 5. This multiple step attack enables to perturbate the data more aggressively thereby allowing a greater challenge to the system's ability to cope with stronger adversarial perturbations. These attacks were performed to test the behavior of the hybrid recommendation system in the presence of adversarial changes of the system input and in order to understand its robustness and ease-of-manipulation.

5 Results & Conclusions

5.1 Results of the Hybrid Recommender System

When a hybrid recommendation based on Jumanji was generated and the list of top movies are shown in the figure. These results also illuminate the operation of the hybrid system of both collaborative and content-based filtering. The recipient of recommendations were calculated using both the user behavior (Collaborative filtering) and attributes of the movies themselves (Content-based filtering). This method guarantees the recommended movies to be of the same type of the ones the user has interacted with and fit well into the content (in terms of genre, cast, and director).

```
Movie Selected:  Jumanji , Index:  1
Searching for recommendations.....
                                    Movie Title
0                                 Colonel Chabert
1  Brother Minister: The Assassination of Malcolm X
2                                 Batman Forever
3                                   Red Rock West
4                                Muriel's Wedding
5                                   The Rocketeer
6                                         Hidalgo
7                                    October Sky
8                                     The Wolfman
```

Figure 5: Recommender results without any attacks

5.2 Results of the Hybrid Recommender System with FGSM Attack

For the FGSM attack on the hybrid recommender system, only few movies had changed in the given recommendation list for Jumanji. Recommendations output of the recommender are given in figure. ***** On the other hand, the relatively small impact on FGSM attack indicates that, although the adversarial example was crafted to change similarity scores slightly or ratings slightly, the ultimate recommendation was not changed significantly. The resistance of the system to such small perturbations demonstrates robustness to low-level adversary intervention.

	Movie Title
0	Colonel Chabert
1	Brother Minister: The Assassination of Malcolm X
2	Batman Forever
3	Red Rock West
4	Muriel's Wedding
5	The Rocketeer
6	Hidalgo
7	October Sky
8	The Wolfman

Figure 6: Recommender results under FGSM Attack

5.3 Results of the Hybrid Recommender System with PGD Attack

Moreover, when the hybrid recommender system was subject to a PGD attack, the changes made to the best recommendations for Jumanji were more radical. The movie title suggestions in figure transformed as shown in the figure. This fact can be explained by the nature of the PGD attack and the possibility to apply iterative perturbations to the similarity scores and ratings. The appearance of completely new movie titles, such as The Amazing Panda Adventure and Nell, along with the total disappearance of some old ones namely, Colonel Chabert and Batman Forever, clearly exemplifies how the PGD attack can not only attack but control the recommender system more efficiently than FGSM.

	Movie Title
0	Brother Minister: The Assassination of Malcolm X
1	The Amazing Panda Adventure
2	Nell
3	Muriel's Wedding
4	Red Rock West
5	The Wolfman
6	Captain America: The First Avenger
7	Hidalgo
8	Jurassic Park III

Figure 7: Recommender results under PGD Attack

5.4 Conclusions

The experimental results of the hybrid recommender system on Jumanji indicate that the system can produce personalized movie recommendations by using a combination of collaborative and content filtering methods. Combining these two approaches, the hybrid system can include both user preferences and movie features and produce not only diversified but also relevant recommendations. But the resistance of such system to adversarial attacks was not uniform. Recommendations were not significantly

affected by the FGSM attack, which demonstrated that the hybrid system is robust to small perturbations. This is likely because the system aggregates together many streams of data—user preferences and item properties—leading to a diffusion of the impact of individual changes. That the output of the hybrid recommender remains unchanged assuming adversarial changes of this kind proves the robustness of the hybrid recommender model under simple adversaries. However for the PGD attacks the recommendations of the recommender system were more altered. The iterative process of the PGD attack allowed it to incrementally control the system and mix in new movies and leave some out. This indicates that the hybrid recommendation approach is robust to simple adversarial attack, but is sensitive to complex, iterative perturbation. This suggests that hybrid recommender systems are resistant to basic, while potentially vulnerable to more advanced, adversarial attacks. That adversarial reflections have an impact on recommendations should highlight the need for developing robust methodologies on these systems against a variety of attacks.

6 Future Work

Although the experiments of this paper indicate that hybrid recommender systems can be robust under small adversarial perturbations, the existence of PGD attacks imply that the resulting system may be open to more sophisticated types of fraud. Under this premise, the natural question arises on how to defend hybrid recommenders from adversarial attacks. One solution might be to adversarially train the system, e.g. a model trained on clean data and data that is contaminated. This would enable the system to learn to detect and withstand adversarial perturbations by becoming more robust. Another interesting direction for further work would be to improve the defence mechanism in this case, so that our hybrid system would be able to cope better with PGD-type attacks. Work on methods like gradient masking or robust optimization to reduce the impact of perturbations at each iteration. Moreover, in the case of the hybrid recommender system, more sophisticated defense algorithms can also be developed, which can target on detecting and neutralizing adversarial attacks in real-time, this ensure the system can detect and neutralize dangerous manipulations in a timely manner.

Furthermore, this work could be extended in the future by running a wider set of adversarial attacks (Carlini-Wagner or DeepFool) to test the general resistance of hybrid recommender systems. By challenging the system with different attackers, it becomes feasible to have a deeper insight on its weak point requiring a particular strengthening. Another interesting direction of research might be to incorporate the use of other data sources and models in the hybrid recommender system, such

as neural networks or graph-based models, and study how the choice of various architectures affects the robustness of the system to adversarial attacks. It is possible to compare traditional hybrid systems and latterly developed, more sophisticated hybrid models to uncover which will provide the greatest resistance to adversarial manipulation.

Finally, analyzing the user experience after exposure to adversarially attacked recommendations would give insight into the impact of these changes on user trust and engagement. It is important to to interpret the wider effects of adversarial attacks on user satisfaction and decision-making in the design of robust and more user-oriented recommender systems that perform reliably in real-world scenarios.

7 REFERENCES

Schein, R. W., Popescul, A., Ungar, L. H., & Pennock, D. M., 2002. Methods and metrics for cold-start recommendations. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.253-260.

Adomavicius, G., & Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and future directions. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp.734-749.

Burke, R., 2007. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), pp.331-370.

Koren, Y., 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.426-434.

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L., 2009. BPR: Bayesian personalized ranking from implicit feedback. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp.452-461.

Goodfellow, I. J., Shlens, J., & Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. Proceedings of the International Conference on Machine Learning, pp.3910-3920.

Papernot, N., McDaniel, P., & Goodfellow, I. J., 2016. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07270.

Ribeiro, M. T., Singh, S., & Guestrin, C., 2016. "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1135-1144.

Zhang, Y., & Chen, X., 2018. Explainable Recommendation: A Survey and New Perspectives. arXiv preprint arXiv:1804.11192.

Deldjoo, Y., Di Noia, T., & Merra, F. A., 2020. A flexible framework for evaluating user and item fairness in recommender systems. User Modeling and User-Adapted Interaction, 31(3), pp.421–455.

Zhang, K., & Wang, C., 2020. Scalable collaborative filtering with distributed learning frameworks. IEEE Transactions on Big Data, 6(3), pp.653-665.

Do, H.-Q., Le, T.-H., & Yoon, B., 2020. Dynamic Weighted Hybrid Recommender Systems. Proceedings of the 22nd International Conference on Advanced Communication Technology, pp.644-650.

Sun, Z., et al., 2025. A Survey on Federated Recommendation Systems. IEEE Transactions on Neural Networks and Learning Systems, 36(1), pp.6-20.
Tang, J., et al., 2020. Adversarial Training Towards Robust Multimedia Recommender System. IEEE Transactions on Knowledge and Data Engineering, 32(5), pp.855-867.
Deldjoo, Y., Di Noia, T., & Merra, F. A., 2020. Adversarial Machine Learning in Recommender Systems (AML-RecSys). Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20), pp.869-872.
Zhou, H., Xiong, F., & Chen, H., 2023. A Comprehensive Survey of Recommender Systems Based on Deep Learning. Applied Sciences, 13(20), pp.11378.
Zhang, D., & Yang, Q., 2020. Hybrid recommendation systems using deep learning. IEEE Transactions on Knowledge and Data Engineering, 32(7), pp.1416-1429.
Nguyen, T., et al., 2024. Manipulating Recommender Systems: A Survey of Poisoning Attacks and Countermeasures. ACM Computing Surveys, 57(1), pp.39.
Rong, H., et al., 2020. Distributed Equivalent Substitution Training for Large-Scale Recommender Systems. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.911-920.
Batmaz, Z., et al., 2019. A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review, 52, pp.1–37.