A Safer Approach to Build Recommendation Systems on Unidentifiable Data

Kishor Datta Gupta[1], Akib Sadmanee[2][a] and Nafiz Sadman[2][b]

[1] *University of Memphis, TN, U.S.A.*

[2] *Silicon Orchard Ltd, Bangladesh*

Keywords: Machine Learning, Deep Reinforcement Learning, Behavior Modeling, Human Activity Recognition, Recommendation Systems, Privacy, Security.

Abstract: In recent years, data security has been one of the biggest concerns, and individuals have grown increasingly worried about the security of their personal information. Personalization typically necessitates the collection of individual data for analysis, exposing customers to privacy concerns. Companies create an illusion of safety to make people feel safe using a mainstream word, "encryption". Though encryption protects personal data from an external breach, the companies can still exploit personal data collected from users as they own the encryption keys. We present a naive yet secure approach for recommending movies to consumers without collecting any personally identifiable information. Our proposed approach can assist a movie recommendation system understand user preferences using the user's movie watch-time and watch history only. We conducted a comprehensive and comparative study on the performance of three deep reinforcement learning architectures, namely DQN, DDQN, and D3QN, on the same task. We observed that D3QN outperformed the other two architectures and achieved a precision of 0.880, recall of 0.805, and F1 score of 0.830. The results show that we can build a competitive movie recommendation system using unidentifiable data.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Personalized recommendations typically need the compilation of individual data for analysis, making users vulnerable to privacy infringement problems. Recent research exhibits that the user's personal data are most beneficial for recommendation systems (Wang et al., 2018). However, data security is a serious concern in recent times, and people, in general, have become more conscious about their personal data. (Jeckmans et al., 2013) studied several privacy issues in existing recommendation systems, which includes breach of information. The data can be traded to third parties without any user consent. Another privacy issue is that cyber-attacks can expose platforms and users' data, which is a pretty common occurrence. (Shyong et al., 2006) discussed exposure and bias in recommendation systems in three domains: the risk factor that comes with recommendation systems, bias in recommendation systems due to personalized attacks, or shilling attacks, and the issues in peer-to-peer recommendation systems. The authors address how recommendation systems are crucial for consumers and businesses to assist the decision-making process. However, the amount of personal data used for this purpose is alarming.

Despite such "bad news", there are several ways a user can now protect their privacy. Users need awareness about their situation. Browsers can digitally inform this, and consent can be taken from users. This also includes having a privacy policy in place to let the user be fully disclosed about the results of sharing their private data. Therefore, users can willingly opt-out from sharing data. Anonymization (Jeckmans et al., 2013) is another smart win-win approach for both users and businesses, where any identifiable information is anonymized while retaining only required information. A simple solution would be to use incognito mode provided by some browsers such as Google Chrome, Mozilla Firefox, etc.

There are algorithmic improvements (Cissée and Albayrak, 2007; Chen and Williams, 2010) that can also counter such privacy issues. For example, (Zhan et al., 2010) mention two approaches to preserve privacy: Homomorphic encryption, which computes the

[a] https://orcid.org/0000-0002-7591-2659

[b] https://orcid.org/0000-0001-6784-0029

Gupta, K., Sadmanee, A. and Sadman, N.
A Safer Approach to Build Recommendation Systems on Unidentifiable Data.
DOI: 10.5220/0010868000003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 589-596
ISBN: 978-989-758-547-0; ISSN: 2184-433X

Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

multiplicative homomorphism, where the two ciphertexts multiplication equals the encryption of the multiplication of the plain texts, and Scalar Product approach, where the information from two parties can be protected, and the computation of similarities can be accurately computed. However, current methods to recommend user content/product/items are unsuitable due to the constraints mentioned above. The provider can not track user choices as users can opt out from provider tracking systems. Furthermore, the provider can not use similar user data to suggest content to new users as users are now more conscious about sharing their demographic, economic, or educational information. These restrictions make it hard to recommend appropriate content to the user. Moreover, the process suffers from a cold-start problem.

Companies these days collect and sell large amounts of user data without the consideration of user privacy. It is claimed that the data is used for a personalized recommendation system. In this paper, we propose a safer approach to utilizing deep reinforcement learning algorithms that can understand human preferences and devise a movie recommendation system without collecting any identifiable user data.

Non-personal data, in its most basic form, is any set of data that does not contain personally identifiable information like names and contact addresses[1]. It is believed to be impossible to build a personalized recommendation system without the use of any identifiable personal data. However, we experimented with only two unidentifiable features from custom-made non-personal session data using three deep reinforcement learning algorithms, namely: Deep Q Network (DQN)(Mnih et al., 2015), Double DQN (DDQN)(Van Hasselt et al., 2016), and Dueling Double DQN(D3QN)(Wang et al., 2016), which results in high accuracy and a high F1 score. Thus it proves that it is indeed possible to build a competitive recommendation system using just unidentifiable non-personal data.

The main contribution of our paper encompasses a comprehensive and comparative study of deep reinforcement learning algorithms for devising a movie recommendation system using unidentifiable data and inaugurating a new paradigm of recommendation systems that respect user privacy.

The organization of our paper is as follows. Section II provides a brief overview of the technical foundation of our findings. Section III contains a thorough review of the literature. Section IV discusses the methodology and the setup of our experiments and the generation of our dataset. Section V presents and analyzes the outcomes of the three deep reinforcement methods. Finally, our article concludes with prospective future work and limitations of the existing version in section VI.

[1]https://indianexpress.com/article/explained/non-personal-data-explained-6506613

2 PRELIMINARIES

In this section, we will briefly touch technical literature involved in this paper.

2.1 Session-based Recommender Systems

Session-based recommender systems tend to capture short-term but dynamic user preferences based on user interactions on a particular session. These interactions include clicks, purchases, cart histories, etc.

2.2 Unidentifiable Data

Unidentifiable data, also interchangeably used as non-personal data, is defined by the absence of identifiers within a data. Typical identifiers are name, address, contact numbers, locations, passwords, etc. Data without such kinds of identifiers are often less valuable and unsaleable. Hackers cannot misuse non-personal data to blackmail and leverage personal ill motives.

2.3 Q-learning

Q-learning is another approach to solve reinforcement learning problems that analyses and improves the quality of the action taken by an agent to transit into another state. It lets agents learn to act optimally in Markovian domains by experiencing the decisions. Q-learning can be formulated by Equation 1, where α is the learning rate that controls the agent's time to adapt to the random changes in the environment. Moreover, the equation presents the temporal difference in the Q-values, which helps to capture the random changes in the environment. (Sutton and Barto, 2011; Watkins and Dayan, 1992).

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma max Q(S_{t+1}, a) - Q(S_t, A_t)] \quad (1)$$

2.4 Deep Q-learning

Deep Q-learning (DQN) is a derivative of Q-learning in deep reinforcement learning, which uses a neural network layer to approximate the Q-value function. Deep Q-learning works best when the action space becomes larger, and computations become sparse. Deep Q network, also commonly known as Deep Q learning, is a deep neural-based Q learning approach to solving RL problems. Deep Q learning helps deal with many action spaces and states by inferring Q-values with neural networks.

2.5 Double Deep Q-learning

The idea of Double Deep Q-learning (DDQN) originates from the motivation to counter the overestimation of action spaces by the DQN algorithm. It is done by decoupling the selection of action evaluation.

2.6 Dueling Double Deep Q-learning

In the dueling variant of the DQN, namely D3QN, an intermediate layer in the Q-Network estimates the state value and the state-dependent advantage function. One may use the predicted Q-Value as the state value with which the advantage estimate is added and then demeaned. This state-independent and state-dependent factorization supports disentangling learning over actions and provides better results.

3 RELATED WORK

The recommendation system is a well-defined problem in Computer Science. Previously matrix factorization (Salakhutdinov et al., 2007; Koren et al., 2009) was a generic method to build recommendation systems. The fundamental goal of matrix factorization is to factorize a user-item rating matrix into two low-rank matrices, each of which reflects a user's or item's latent components. It is unsuitable for a session-based recommendation system since the user preference is supplied only by a few affirmative clicks. Then Markov chains were introduced to anticipate users' next behavior. (Shani et al., 2005) used Markov decision processes (MDPs) to solve the sequential optimization problem of recommendation generation. The authors in the paper (Chen et al., 2018b) proposed to frame the recommendation problem as one of (model-based) reinforcement learning. The two main innovations were: 1) a model and objective for learning the environment and reward models; 2) a cascaded DQN framework for reasoning about a combinatorial number of actions (i.e., which subset of items to recommend user). Another work (Yu, 2018) introduced a sample efficiency issue of reinforcement learning, which is the learning of the RL agent on sample data. The author discussed possible ways to reduce the sample cost of the domain.

(Belacel et al., 2020) predicts recommendations based on a user's previous interests. Though their main contribution lies in the introduction of k Closest resemblance classifier for Content Based Filtering recommender systems, it is a fact that the algorithm is based on a user's identifiable data. (Yodogawa and Kuwabara, 2020) simulates a round-table negotiations for recommending an item to the users. While the idea is brilliant, its complexity is rather unclear as the program runs until all the negotiators agree on a product. (Argente et al., 2017) took a rather unique approach to the privacy issue. They focus on raising awareness among people about public exposure of data. A hybrid neural-network and higher order Markov model (Chen and Lin, 2019) is proposed by the author to build recommender systems based on temporal data. They used user information, set of items used, time, and ratings. The authors in the paper (Ikemoto and Kuwabara, 2019) proposes a method for improving knowledge about items in a database for use in an interactive recommender system. The proposed method is integrated into a recommender system and is triggered when the system detects a problem with the item database based on user feedback about recommended items. The proposed method collects information from a user through interactions that are similar to those of a recommendation process. Another work (Walter et al., 2007) take a different approach to build a recommendation system using graphs where each nodes are termed as agents. Each agents aim is to recommend an item to the other on basis of preference. Thus each agent forms a 'relation of trust' with each other. This trust-based recommendation system performs optimally. However, each agent can be identified due to the personal information it holds.

There are several applications (Rohde et al., 2018; Mahmood et al., 2008; Nawrocka et al., 2018) of reinforcement learning in recommendation systems deployed on social sites, medical websites, e-commerce sites, travel sites, etc. Apart from traditional reinforcement learning approaches (i.e, Markov-decision process) (Sutton and Barto, 2011)), there are deep learning implementations that accounts for the dynamic nature of environment in applications (Chen et al., 2018a; Huang et al., 2021). (Xin et al., 2020) proposed two frameworks, namely Self-Supervised Q learning (SQN) and Self-Supervised Actor-Critic (SAC). The frameworks are based on an approach that augments standard recommendation models with two output layers. One layer is for self-supervised learning, and the other one is for RL. The last layer drives the supervised layer to emphasize certain rewards, and the self-supervised layer with cross-entropy loss provides strong gradient signals for parameter updates. (Zhao et al., 2018) introduced a novel concept of integrating negative feedback (what a user does not like) in recommender systems using deep reinforcement learning. While they devised separate states and reward functions for negative feedback, our approach considers the negative feedback within the pattern of recognizing preference priorities. Recommender systems installed in various sites usually keep backlogs. (Chen et al., 2019) in their paper described how logs from recommender systems in YouTube could be leveraged, and the bias can be addressed using top-K-off policy correction algorithm from YouTube. Some effort has already been put to improve the reliability of group recommendations (Rossi and Cervone, 2016; Alabdulrahman et al., 2019; Rossi et al., 2017). However, they don't solve the issue with data security

(Wang et al., 2019) in their paper discussed many folds of session-based recommender systems encapsulating diverse datasets used and the challenges faced by such systems. Session-based recommender systems learn from the interactions of users on sessions. They discussed recommender systems based on anonymous data and mentioned that session-based recommendation systems are mostly dependent on user data in the session, and their particular interactions like purchases and assumed data are orderly. However, despite attempting to address privacy issues in collecting user data, none of the research considers an alternate route to constructing efficient recommendation systems that will not collect identifiable user data. The "privacy-preserving" techniques consider building robust databases (Liu et al., 2021) for users that will cut off data leakage, or smart data encryptions (Zhan et al., 2010; Park et al., 2020). Nevertheless, some identifiable data is still collected. We aim to make a change to the existing practices by offering consumers with useful suggestions by "not collecting" user information rather than "protecting" the information after it has been collected.

4 METHODOLOGY

4.1 Hypothesis

A watch-time based movie recommendation system is the system of predicting a user-preferred movie based on his/her current sequential movie interaction data. Here we formulate the movie recommendation based on the movie watch-time problem. Let a movie watching sequence S be $[x_1, x_2, ..., x_{n-1}, x_n]$ of a user U, where $x_i \in (1 \leq i \leq n)$ is a recent movie watched by U. We train a RL agent A to recommend a user-preferred genre U_y based on S. A sequential model M_S maximizes the reward function F_r of an agent A based on the outcome $O(s,a)$ from taking action a on a state s taken by A.

Our hypothesis states that a RL agent can learn to recommend a set of movies to a user based on only the user's movie watch-time. Here we formulate our hypothesis to solve the movie recommendation based on the movie watch-time problem. Let, a domain specific dataset D have n classes where d_1, d_2, ... d_n are subsets of D representing data points for class 1,2,...,n respectively. The agent A for the RL algorithm can learn to recommend a set of data from D on the basis of time spent on d_1, d_2, ... d_n only. Our reward R can be formulated as:

$$R(A) = \begin{cases} 2, & \text{for } optimal\ action \\ 1, & \text{for } 2^n d\ most\ preferred\ action \\ -1, & \text{otherwise} \end{cases} \quad (2)$$

4.2 Algorithms

We applied three different well-known deep reinforcement learning algorithms to our dataset. On top of each algorithm, a sequential model is constructed with 3 Dense layers to process our movie watch time and output an action, i.e., the recommendation. These algorithms are DQN, DDQN, and D3QN. We use the hubber loss (L) function to learn from the penalty incurred by the estimation procedure. The hubber loss is formulated using the following equation:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2, & \text{for } |a| \leq \delta \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise} \end{cases}$$

This function is quadratic for values smaller than a certain threshold and linear for values larger than the threshold with equal values and slopes of the different sections at the two points where —a— = δ .

The process of integrating these algorithms into our task is shown in Algorithm 1. We created our environment using open source OpenAI Gym[2]. A sequential model processes the environment, and upon that, the algorithms are applied. The reward mechanism is computed to guide the agent in taking the right course of action. We provide the agent a range for movie watch time that can be considered optimal states.

4.3 Experimental Setup

A reinforcement learning setup is broken into components for interpretability (Sutton and Barto, 2011). These are environment and observation, states, agent and its actions, and rewards. Each term is explained according to our task-specific setup.

[2]https://gym.openai.com/

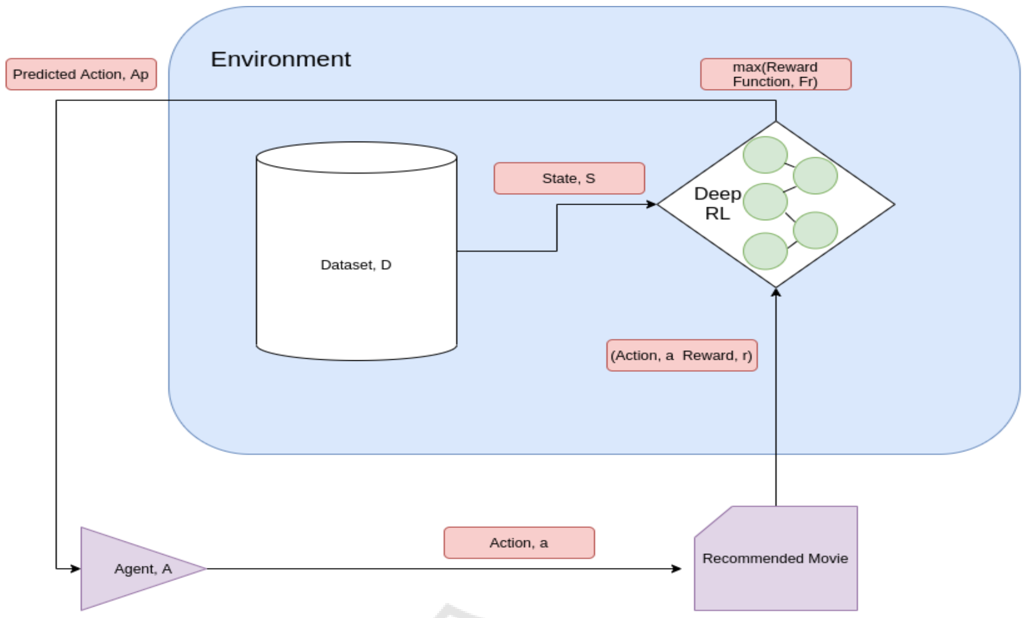


Figure 1: Environment of our reinforcement learning problem. The goal of the agent is to recommend correct list of movies. The action taken by the agent is evaluated by the deep reinforcement learning which aims to maximize the reward.

Algorithm 1: Our approach to constructing the recommender agent.

```
Initialize an environment
Action space = Three discrete values according to three classes
Observation space = List of possible timestamps from our dataset $D$
Initialize a sequential model with three dense layers with ReLU activation on the first two and Linear on the output
for number of epocs do
    for each steps do
        Train a deep reinforcement learning agent (algorithm) with the sequential model
        Check action with reward mechanism from Step Function
        Record and update action and reward history
    end for
end for
```

4.4 Environment and Observation

The environment consists of the dataset used for this experiment and the Deep Reinforcement learning algorithms that are built to maximize the reward function. The observation will be the output (recommendation) predicted by the algorithms.

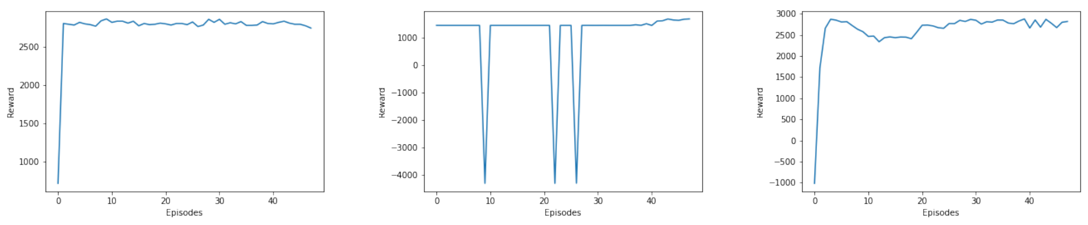
- **States:** A state contains a particular movie watch time from D.
- **Agent:** A program that has some set of actions it can take to change the states.
- **Actions:** The actions of an agent are to select grid arrangements or states that will be very close to the user preference.
- **Reward:** The reward function is dependent on the algorithm we will use. In general, the agent earns a reward after each state transition.

Figure 1 demonstrates our learning environment, which contains an agent that recommends one or more movie genres, a deep reinforcement learning algorithm that corrects the agent's output by looking at the action, reward, and state from the training data. The goal of this algorithm is to maximize compensation achieved by the agent and minimize the loss function. The selected action is then fed to the agent to teach the correct course of action.

4.5 Dataset

There are handful of publicly available datasets on movie recommendation systems on the Internet[3]. However, they consist of user-identifiable features, i.e. user id, ratings, clicks, etc. We aim to use an unidentifiable and unsaleable dataset to assert our hypothesis, but no such standard real datasets nor their features suited our purpose. There are also few state-of-the-art movie recommendation models (Kim and Suh, 2019). However, they are trained on identifiable features, which

[3]https://www.kaggle.com/netflix-inc/netflix-prize-data https://github.com/kishan0725/The-Movie-Cinema https://github.com/gauravtheP/Netflix-Movie-Recommendation-System



(a) DQN (b) DDQN (c) D3QN

Figure 2: Comparative analysis of Reward graphs generated from three deep reinforcement learning algortihm.

contradicts our purpose of research. Therefore, for this research, we manually constructed our dataset with the help of 20 human annotators. Each annotator was asked to watch 50 movies each from three genres from an anonymous streaming site. An annotator will watch 150 movies and 3000 movies from all the annotators in total. These genres were History, Action, and Comedy. We record the watch time of an annotator watching a particular movie. The watch time is started when the annotator starts the movie and ends when the annotator closes the movie. For reference, we call our dataset Session Specific Movie Data or SSMD in short. Table 1 summarizes the data statistics of SSMD.

Table 1: Session Specific Movie Data Statistics.

Genre	**No. of Data-points**	**Average**	**High**	**Low**
Action	1000	105.296	149.965	60.038
History	1000	50.936	69.993	31.002
Comedy	1000	20.0596	29.990	10.016

In this experiment, we aim to architect an agent that can understand human preference, specifically, one or more movie genres, for our problem statement. We would also like to see if this can be done without using any identifiable data. Therefore, we construct our dataset with the movie watch time and the genres only. This alone cannot identify the user, nor can it interpret anything specific about the movies. Precisely, this is unsaleable data. However, in our Methodology, we show that our agent can capture human preferences and recommend movies based solely on the watch times. There are a few features that we also took into consideration like movie watching frequency. Though the feature may seem intriguing, our main focus is the data generated through a single activity session. It is highly unlikely that a person would watch the same movie twice in a single session. Therefore, we rejected that idea and went with only the movie watch-time and genres.

5 RESULT ANALYSIS

In this section, we experiment with our aforementioned deep reinforcement learning algorithms on the dataset that we created. For each algorithm, we record the precision, recall, and F-1 scores. These scores are computed against a set of truth values from a survey conducted among CS university graduates. The students were given a case study consisting of a movie watch pattern and then asked what movies they would recommend based on three criteria: Action, History, and Comedy. Answers with the most common criteria were chosen. The goal of our reinforcement learning agent is to learn the correct preference as best as possible. The three deep learning approaches we utilized to evaluate our suggested technique were constrained in terms of parameter tweaking and data. However, in our study, we demonstrated that the technique is feasible and accessible to researchers exploring other areas with the same notion utilizing different cutting-edge recommender systems. Figure 2 shows a comparative analysis of the three deep reinforcement learning algorithms. The reward graphs in Figure 1(a)DQN, (b)DDQN, and (c)D3QN algorithms respectively show that after an initial jump in reward, over the rest of the 47 iterations, the reward remains steady. As the algorithms start by assigning random rewards, we can notice rises and falls in the early stage of training. As the training progresses, the reward becomes more and more steady. However, in Figure 2(b), the DDQN algorithm starts to rise after 40 iterations. The algorithm could not get past the initial random rewards for the first 25 iterations while the other two algorithms, namely DQN and D3QN, got past that point at an early stage of the training, at the first five iterations to be precise. From the analysis, we can assert that the agent learns to take the correct course of action for our recommendation problem. In table II, we present a comparative study between 3 deep RL networks, namely, DQN, DDQN, and D3QN. We observe that D3QN has the highest precision, recall, and f1 score among the three algorithms. D3QN is the upgraded

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 2: Comparative analysis between DQN, DDQN, D3QN.

Movie recommended by Agent								
DQN			**DDQN**			**D3QN**		
Action	**History**	**Comedy**	**Action**	**History**	**Comedy**	**Action**	**History**	**Comedy**
10	3	2	9	3	3	9	4	2
Human Prefered								
Action			**History**			**Comedy**		
8			4			3		
Score								
DQN			**DDQN**			**D3QN**		
Precision	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
0.789	0.703	0.734	0.500	0.444	0.716	0.880	0.805	0.830

version of DDQN. Intuitively the D3QN algorithm can determine valuable states without learning the effect of each action at each state. This allows D3QN to perform well on the specific task of recommending movies based on limited data. It is transparent from the performance of the three deep learning algorithms that our research meets the study goal of developing a safer recommendation system that does not require any identifying data. Though in this research we proved that movie recommendation is doable with the user's non-personal data, it is also open to researchers who dwell in other domains of recommendation systems.

6 CONCLUSION

In this research, we propose a safer approach to use deep reinforcement learning algorithms that can understand human preference for movies without the need for any identifiable data from the user. Among the three algorithms used in our experiments, D3QN outperforms DQN and DDQN by a significant margin by achieving a precision of 0.880, recall of 0.805, and F1 score of 0.830. Therefore, we conclude that D3QN is by far the best algorithm to be used in recommender systems for unidentifiable data with a single feature. This research might pave the way for safer and more secure recommendation systems. Though the study is based on a small movie dataset, it has the potential to affect other recommendation systems as well. Our experiments prove that the approach works even if there are just two independent, distinct features that are not private data. Future work of this research includes but is not limited to comparing with existing recommendation systems and conducting applicability of our algorithms in commercial sites.

REFERENCES

Alabdulrahman, R., Viktor, H., and Paquet, E. (2019). Active learning and user segmentation for the cold-start problem in recommendation systems. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR, (IC3K 2019)*, pages 113–123. INSTICC, SciTePress.

Argente, E., Vivancos, E., Alemany, J., and Garcia-Fornes, A. (2017). Educating in privacy in the use of social networks. *Education in the Knowledge Society*, 18(2):107–126.

Belacel, N., Wei, G., and Bouslimani, Y. (2020). The k closest resemblance classifier for amazon products recommender system. In *ICAART*.

Chen, H. and Lin, Z. (2019). A hybrid neural network and hidden markov model for time-aware recommender systems. In *ICAART*.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. (2019). Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 456–464.

Chen, S. and Williams, M.-A. (2010). Towards a comprehensive requirements architecture for privacy-aware social recommender systems. In *Conferences in Research and Practice in Information Technology Series*.

Chen, S.-Y., Yu, Y., Da, Q., Tan, J., Huang, H.-K., and Tang, H.-H. (2018a). Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1187–1196.

Chen, X., Li, S., Li, H., Jiang, S., and Song, L. (2018b). Neural model-based reinforcement learning for recommendation.

Cissée, R. and Albayrak, S. (2007). An agent-based approach for privacy-preserving recommender systems. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8.

Huang, L., Fu, M., Li, F., Qu, H., Liu, Y., and Chen, W. (2021). A deep reinforcement learning based long-term recommender system. *Knowledge-Based Systems*, 213:106706.

Ikemoto, Y. and Kuwabara, K. (2019). On-the-spot knowledge refinement for an interactive recommender system. In *ICAART*.

Jeckmans, A. J., Beye, M., Erkin, Z., Hartel, P., Lagendijk, R. L., and Tang, Q. (2013). Privacy in recommender systems. In *Social media retrieval*, pages 263–281. Springer.

Kim, D. and Suh, B. (2019). Enhancing vaes for collaborative filtering: flexible priors & gating mechanisms. *Proceedings of the 13th ACM Conference on Recommender Systems*.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Liu, X., Deng, R. H., Raymond Choo, K.-K., and Yang, Y. (2021). Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes. *IEEE Transactions on Emerging Topics in Computing*, 9(1):456–470.

Mahmood, T., Ricci, F., Venturini, A., and Höpken, W. (2008). Adaptive recommender systems for travel planning. In *ENTER*, volume 8, pages 1–11.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Nawrocka, A., Kot, A., and Nawrocki, M. (2018). Application of machine learning in recommendation systems. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 328–331.

Park, J., Kim, D. S., and Lim, H. (2020). Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures. *IEEE Access*, 8:203564–203579.

Rohde, D., Bonner, S., Dunlop, T., Vasile, F., and Karatzoglou, A. (2018). Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. *arXiv preprint arXiv:1808.00720*.

Rossi, S. and Cervone, F. (2016). Social utilities and personality traits for group recommendation: A pilot user study. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 38–46. INSTICC, SciTePress.

Rossi, S., Cervone, F., and Barile, F. (2017). An off-line evaluation of users' ranking metrics in group recommendation. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 252–259. INSTICC, SciTePress.

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798.

Shani, G., Heckerman, D., Brafman, R. I., and Boutilier, C. (2005). An mdp-based recommender system. *Journal of Machine Learning Research*, 6(9).

Shyong, K., Frankowski, D., Riedl, J., et al. (2006). Do you trust your recommendations? an exploration of security and privacy issues in recommender systems. In *International Conference on Emerging Trends in Information and Communication Security*, pages 14–29. Springer.

Sutton, R. S. and Barto, A. G. (2011). Reinforcement learning: An introduction.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Walter, F. E., Battiston, S., and Schweitzer, F. (2007). A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16:57–74.

Wang, C., Zheng, Y., Jiang, J., and Ren, K. (2018). Toward privacy-preserving personalized recommendation services. *Engineering*, 4(1):21–28.

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M., and Lian, D. (2019). A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864*.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

Xin, X., Karatzoglou, A., Arapakis, I., and Jose, J. M. (2020). Self-supervised reinforcement learning for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 931–940.

Yodogawa, D. and Kuwabara, K. (2020). Reaching agreement in an interactive group recommender system. In *ICAART*.

Yu, Y. (2018). Towards sample efficient reinforcement learning. In *IJCAI*, pages 5739–5743.

Zhan, J., Hsieh, C.-L., Wang, I.-C., Hsu, T.-S., Liau, C.-J., and Wang, D.-W. (2010). Privacy-preserving collaborative recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(4):472–476.

Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J., and Yin, D. (2018). Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1040–1048.