



# A contextual-bandit approach for multifaceted reciprocal recommendations in online dating

Tulika Kumari<sup>1</sup> · Ravish Sharma<sup>2</sup> · Punam Bedi<sup>3</sup>

Received: 29 December 2021 / Revised: 29 March 2022 / Accepted: 30 March 2022 /  
Published online: 25 June 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Recommender Systems (RS) provide an effective way to deal with the problem of information overload by suggesting relevant items to users that the users may prefer. However, many online social platforms such as online dating and online recruitment recommend users to each other where both the users have preferences that should be considered for generating successful recommendations. Reciprocal Recommender Systems (RRS) are user-to-user Recommender Systems that recommend a list of users to a user by considering the preferences of both the parties involved. Generating successful recommendations inherently face the exploitation-exploration dilemma which requires predicting the best recommendation from the current information or gathering more information about the environment. To address this, we formulate reciprocal recommendation generation task as a contextual bandit problem which is a principled approach where the agent chooses an action from a set of actions based on contextual information and receives a reward for the chosen action. We propose SiameseNN-UCB algorithm: a deep neural network-based strategy that follows Siamese architecture to transform raw features and learn reward for the chosen action. Upper confidence bound type exploration is used to solve exploitation-exploration trade-off. In this algorithm, we attempt to generate reciprocal recommendations by utilizing multiple aspects such as multi-criteria ratings of a user, popularity-awareness, demographic information, and availability of users. Experimental studies conducted with speed dating data set demonstrate the effectiveness of the proposed approach.

---

✉ Tulika Kumari  
tulika.singh@rajguru.du.ac.in

Ravish Sharma  
ravish.sharma@pgdav.du.ac.in

Punam Bedi  
pbedi@cs.du.ac.in

<sup>1</sup> Shaheed Rajguru College of Applied Sciences for Women, University of Delhi, Delhi, 110096, India

<sup>2</sup> P.G.D.A.V. College, University of Delhi, Delhi, 110065, India

<sup>3</sup> Department of Computer Science, University of Delhi, Delhi, 110007, India

**Keywords** Reciprocal recommender systems · Contextual bandits · Online dating · Deep neural network · Siamese neural network

## 1 Introduction

Recommender systems (RS) act as an essential tool to assist users to deal with information overload by predicting their future preferences in terms of generated recommendations and suggesting relevant items from the enormous information available online. Reciprocal Recommender System (RRS) (Pizzato et al., 2010) is a user-to-user recommender system where people are recommended to each other such as job recommendation, mentor-mentee matching, and online dating. In such systems, a recommendation is successful only if mutual preferences are satisfied. Some of the important features of reciprocal recommender systems which distinguish them from traditional RS are listed below:

- **Reciprocity:** Agreement of preferences between the user receiving a recommendation and the user being recommended is important for a successful match.
- **Limited availability of users:** In RRS, users receiving recommendations as well as users who are being recommended have limited availability. Recommending a person to too many users should be avoided in RRS whereas items can be recommended to any number of users provided they are available (in the stock) in traditional RS.
- **Users in traditional RS typically continue to engage in the system if they receive successful recommendations. However, in RRS, users normally stop using the system even after receiving successful recommendations as they may no longer need it. For example, in online dating, once a user finds a suitable partner, he or she most likely stop using the system. Likewise, in online recruitment, a person once finds a desirable job position, he/she may no longer require the system.**

Many real-world situations present dilemmas on choices, that is, whether to select the best decisions given current information (exploitation) or gather more information about the environment (exploration). This is known as exploitation vs. exploration trade-off which requires balancing reward maximization based on the knowledge that has already been acquired by the agent and attempting new actions to further add knowledge. RRS faces an exploitation-exploration dilemma while generating recommendations as it involves two choices: exploit the known user preferences or explore other preferences the user may have. In essence, they need to exploit their knowledge about the previously chosen users as a potential match for any user, while also exploring new choices for the user when generating recommendations in online dating. Multi-Armed Bandit (MAB) problem is a classic problem that well demonstrates the exploitation vs exploration dilemma (Slivkins, 2019). It is inspired by a gambling scenario where a gambler faces several slot machines or one-armed bandits, that appear identical, but yield different rewards or payoffs. A gambler pulls a lever arm and receives rewards at some expected rate. The gambler has two fundamental choices, either pull a new lever arm to acquire new information (exploration), or to make optimal decisions based on the available information (exploitation). The Multi-armed bandit problem refers to the challenge of constructing a strategy for pulling the levers in the absence of prior knowledge of the reward for any of the levers which require creating a balance between exploitation and exploration. In recent years, the MAB framework has garnered attention in applications such as recommender systems and information retrieval (Bouneffouf & Rish, 2019). Multi-armed bandit algorithms often interact with agents which are defined as

computer systems that are situated in an environment and perform autonomous action in that environment to meet its objectives. Contextual multi-arm bandit is a version of MAB where at each iteration, the agent uses contextual information and rewards of the arms played in the past to select the next arm or action (Li et al., 2010). Contextual bandits (Auer et al., 2002; Langford & Zhang, 2007) are sometimes known as associative reinforcement learning (Barto & Anandan, 1985). In a contextual bandit problem, an agent chooses an action based on observed contextual information/context that usually comprises known properties of user profiles and features of actions, if any. In return, a numerical “reward” is computed for the chosen action. The behaviour of an agent in contextual bandits can be described as a policy which is a function that maps past observations of an agent and the contextual information to actions. In reciprocal recommender systems, users are mostly willing to provide detailed information about themselves and their preferences due to the reciprocal nature of the recommendations. We utilized this contextual information about the users and their preferences to model the problem of generating reciprocal recommendations as contextual bandits approach.

In this paper, we formulate reciprocal recommendation generation task as a contextual bandit problem to address partner selection in online dating which is influenced by several factors such as:

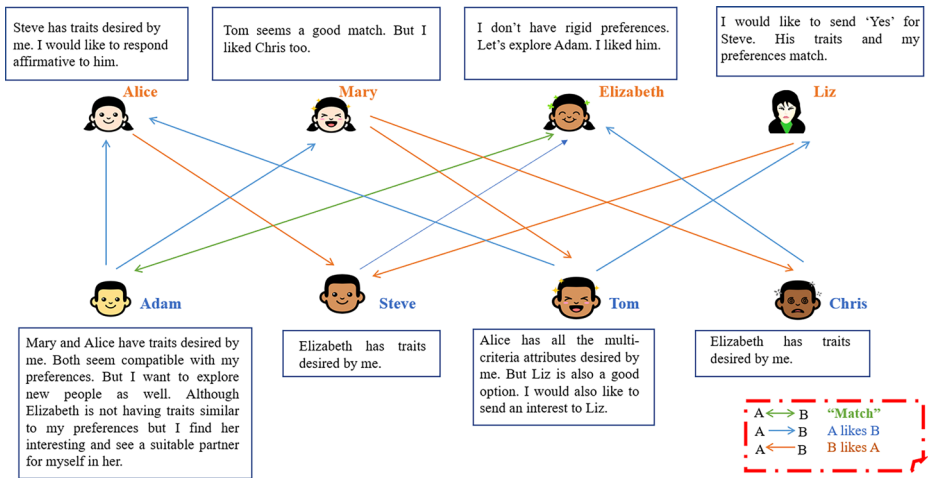
- Multi-criteria ratings of user’s self-assessment for various criteria namely attractiveness, sincerity, intelligence, fun, and ambition.
- Multi-criteria ratings of user’s preferences for attractiveness, sincerity, intelligence, fun, ambition, and shared interests in the desired partner.
- Demographic attributes of a user.
- Willingness of users to explore new choices.

For example, suppose Adam went for a date with Mary, the decision of Adam for Mary to select as a match is dependent on many factors. It can be influenced by Adam’s own multi-criteria attributes such as attractiveness, sincerity, intelligence, fun, ambition, or desirability of these attributes in a potential date, demographic attributes, interests/hobbies. Apart from these factors, the decision of Adam is also influenced by his interest to try/explore new choices. Likewise, the decision of Mary’s for Adam is influenced by her exploratory behaviour in addition to preferences, demographic attributes, and multi-criteria ratings. Thus, a successful match in online dating requires mutual agreement on the preferences of both the parties involved which can be interpreted in terms of their contextual information and exploratory aspects.

Figure 1 demonstrates a toy example that depicts exploitation-exploration performed by people while selecting their suitable partner in online dating.

We postulate that the task of generating reciprocal recommendations in online dating or other such reciprocal recommender systems can be modelled by contextual bandits. The main motivation is that the current user for whom we want to generate recommendations with a known “user profile” arrives in each round and our task is to select one of the users to recommend as a suitable match. Context is the known information of current users and actions that are essentially the choice of any user to be recommended as a potential match. The reward depends on the context and the chosen action.

This paper formulates reciprocal recommendation generation task in online dating as a contextual bandit problem and proposes a deep neural network-based strategy for generating reciprocal recommendations. The main motivation is that in RRS users often willingly provide detailed profiles of themselves and their preferences which can be used as



**Fig. 1** Toy example depicting Exploitation/Exploration used by people while selecting a suitable partner in online dating

contextual information or “context” to generate reciprocal recommendations. For example, in online dating or recruitment, users have evident motivation to provide detailed information about themselves and their ideal partner or desirable job position/candidate because they are aware that a successful match is also dependent on other persons involved. In online dating often our given preferences for partners during sign-up are influenced by our own attributes. But some users end up choosing partners who are superior to them on gender stereotypical attributes such as attractiveness and intelligence, others end up with partners who are inferior to them on these attributes and few also end up choosing partners having qualities different from what they have listed. Thus, it becomes important to consider other aspects as well in addition to the user’s expectations. It is also critical to consider popularity awareness and limited availability of users. Popularity bias arises when the RS tends to favour a few popular users while not giving equal and fair attention to others. It can negatively impact both the users in RRS. For example, in online dating, if a popular user is recommended to too many people, he/she may become overwhelmed by the attention and stop responding. People who are trying to initiate contact with him/her may also become discouraged as he/she is not replying or replies negatively. To make our recommendations fair we have penalized popular users. Popularity penalty for a user  $u$  denoted by  $pp_u$  is computed as the ratio of users on whose matches  $u$  occur to the total matches. As people have limited availability, so dating habits (measured by the attributes representing his/her frequency to go on dates and primary goal for date) of a user are also considered while generating reciprocal recommendations.

Contributions of this paper are manifolds which are summarized below:

- To the best of our knowledge, we present one of the first attempts to study RRS as a contextual bandit problem.
- We have captured multiple aspects such as demographical information, hobbies, exploratory behaviour, primary goal and frequency of dating in addition to a user’s given preferences for a partner while generating reciprocal recommendations for any user. Upper Confidence Bound type exploration that keeps track of our confidence in

our assessments of the estimated values of all of the arms is used for solving inherent exploitation vs. exploration trade-off.

- With the proposed multifaceted reciprocal recommendation algorithm SiameseNN-UCB we have addressed popularity bias and limited availability of users.
- Experiments conducted on speed-dating experiment data<sup>1</sup> demonstrate the effectiveness of our proposed SiameseNN-UCB over state-of-the-art approaches namely RECON Pizzato et al. (2010) and Zheng et al. (2018), COUPLET (Tay et al., 2018), DeepFM (Guo et al., 2017) and biDeepFM (Yıldırım et al., 2021).

In a variety of application domains, exploitation-exploration dilemma occurs; highlighting the importance that if the agent is willing to explore a little and chooses suboptimal actions in the short run, it might discover optimal long-term strategy. It is critical for an agent to solve the trade-off between exploitation and exploration to maximize its long-term performance.

Our proposed approach can be applicable to domains where reciprocity and exploitation-exploration trade-off is important for generating successful and new/informative recommendations such as

- Online recruiting systems that try to find suitable matches for its users (job seeker(s) and recruiter(s)). Job seeker(s) often want to explore new and exciting roles/professions that may vary according to the state-of-the-art and recent trends in the industry whereas recruiter(s) may be in continuous search of best eligible candidate(s).
- Socializing, skill sharing, academic/scientific collaboration are some other domains where exploitation-exploration dilemma can also be resolved with the proposed contextual bandit-based approach while generating reciprocal recommendations. For example, mentor-mentee matching systems could be one such system where often people want to collaborate with people from different fields of work with different skills and expertise for multidisciplinary work.

We can utilize contextual information of the users and their preferences to model the problem of generating reciprocal recommendations in various domains as contextual bandits approach. Recommendation can be an action/arm and reward can be observed if the action is accepted by both the parties. With the proposed SiameseNN-UCB approach, we tried to combine the predictive power of deep neural networks and reinforcement learning which involves learning from the environment to determine an optimal (or nearly optimal) policy for the environment. Our proposed approach can also handle cold start problem as users are not required to provide any input apart from their profile before receiving recommendations and are therefore able to engage with the system right from the start. Empirical evaluation of SiameseNN-UCB conducted on speed dating experiment data set demonstrates its effectiveness. The experimental results demonstrate that SiameseNN-UCB outperforms the state-of-the-art algorithms in terms of several evaluation criteria thus validating the importance of considering exploratory aspects along with the contextual information for generating successful reciprocal recommendations.

The rest of the paper is organized as follows. Section 2 covers related work with focus on RRS and contextual-bandit problem. In Section 3, we present basic concepts followed by our proposed approach for the generation of reciprocal recommendations in Section 4. Section 5 describes dataset, implementation and experimental results. Section 6 concludes the paper followed by references.

<sup>1</sup><https://www.kaggle.com/annavictoria/speed-dating-experiment>

## 2 Related work

In this section, we first discuss Reciprocal Recommender Systems and the classic methods proposed by researchers to solve it, then we introduce how recommender systems use Contextual-bandit based methods to achieve the exploitation-exploration balance.

### 2.1 Reciprocal recommender system

RECON: a content-based approach for generating people-to-people recommendations in online dating was proposed by Pizzato et al. (2010). To compute unilateral preference scores for user  $x$ , the attributes of all users with whom  $x$  initiated contact and all users whose messages were positively replied to by  $x$  were considered. Reciprocal compatibility score was calculated as the harmonic mean of the two unilateral preference scores based on the preferences and attributes of a user. Akehurst et al. (2011) proposed CCR: content collaborative reciprocal recommender approach by combining the predictive capabilities of content-based and collaborative strategy in online dating. Users similar to the target user were computed by exploiting user preferences using content-based approach. The interaction history of similar users was used in the collaborative filtering part to generate possible reciprocal recommendations. MEET was proposed by Li and Li (2012) as a framework for a reciprocal recommender system. A bipartite graph based on the relevance between users of different sets was constructed and mutual relevance score between users as the product of two unilateral relevance scores was computed in this work. Alanazi and Bain (2016) developed a reciprocal recommender using hidden Markov models. In Zheng et al. (2018) authors used multicriteria ratings and defined utility as the similarity between user expectation and dating experience. COUPLET (Tay et al., 2018) was a deep learning model with Gated Recurrent Units and coupled attention layers to model the interactions between two users for solving the problem of relationship recommendation. LFRR (Neve & Palomares, 2019) for heterosexual online dating gathered latent attributes from a preference matrix using matrix factorization. Matrix factorization was used to train two latent factor models, one for each preference matrices representing preference of female users towards male users and male user's preference for female users. Yu et al. (2016) used communities of like-minded users to learn from experienced users to produce successful recommendations for new users. Zang et al. (2017) used supervised machine learning techniques and users' profiles and dating social networks were used to acquire information and preferences of users. The authors argued that profile-based features and graph-based features can be used to predict reciprocal match. BlindDate (Rodríguez-García et al., 2019) combined Content-Based and knowledge-based recommendations. Semantic similarity between users having similar search preferences was computed by comparing their preferences and the weights assigned by them.

In addition to online dating, online recruitment (Malinowski et al., 2006; Wenxing et al., 2015), mentor-mentee matching (Li, 2019), social media platforms (Neve & Palomares, 2020), online learning environments (Potts et al., 2018) are some other applications where RRS can be used to yield successful reciprocal recommendations.

### 2.2 Contextual-bandit in recommender systems

Pilani et al. (2021) presented a comparative analysis of LinUCB, Hybrid-LinUCB and CoLin. Altulyan et al. (2021) formulated a recommendation system to help Alzheimer's patients using contextual bandit framework to tackle dynamicity in human activity patterns.

Yang et al. (2020) proposed a contextual bandit algorithm for online recommendation via sparse interactions between the users and the recommender system using positive and negative responses to build the user preference model, ignoring all non-responses. Intayoad et al. (2020) used past student behaviours and current student state as contextual information to provide learning objects to learners in online learning systems. Xu et al. (2020) considered two reward models with disjoint and hybrid payoffs with the assumption that the preferences of users towards items are piecewise-stationary, i.e., the reward model may undergo abrupt changes but between two consecutive change points, the model remains fixed. He et al. (2020) proposed UBM-LinUCB (LinUCB with User Browsing Model) to address the position bias and pseudo-exposure issue that arises due to the limited screen size of mobile devices while recommending a list of commodities to users. Santana et al. (2020) proposed meta-bandit where each option maps to a pre-trained, independent recommender system which is selected according to the context. Yang et al. (2020) focused on dynamic resource allocation in the contextual multi-armed bandit problems based on the assumption of linear payoff function like LinUCB. Zhang et al. (2020) incorporated conversational mechanism into contextual bandit framework for arm selection strategy and key-term selection strategy through conversations and arm-level interactions with the user. Liu et al. (2018) proposed a transferable contextual bandit policy for cross-domain recommender systems using Upper Confidence Bound (UCB) principle. In (Liu et al., 2018) dependent click model is formulated using contextual information and Upper Confidence Bound is calculated for the attraction weight which denotes the probability with which a user is attracted to an item for web page recommendation. In Zhang et al. (2018) user feedback and prior information of contexts were used to improve Click-Through Rate which is commonly used to measure the payoff of a recommendation. Balakrishnan et al. (2018) introduced Behaviour Constrained Contextual Bandits (BCCB) for online movie recommendations with behavioural constraints. The agent was trained with samples of context and subsequent arm pulled to learn desired constraints during the teaching phase. Contextual Thompson Sampling Algorithm was then used to learn a policy from these examples during the online recommendation phase. Shen et al. (2018) proposed a bandit approach using deep learning that tracks historical events using “memory cells” which learn user preferences and it was based on  $\epsilon$  greedy policy.

All the works described above use Contextual Bandits in traditional recommender systems. To the best of our knowledge, none of the work in RRS literature has used contextual bandits focussing on exploitation-exploration dilemma to generate reciprocal recommendations. In this paper, we define the reciprocal recommendation generation task as a contextual bandit problem in order to address the exploitation-exploration dilemma.

### 3 Preliminaries

In this section, we provide a brief introduction of contextual bandits and Siamese neural network.

#### 3.1 Contextual bandits

Bandit problems were introduced by Thompson (1933) in 1933 for the application of clinical trials. A bandit problem is a sequential interaction between an agent and an environment over  $T$  rounds. In each round  $t \in T$ , the agent first chooses an action from a given set

of actions and the environment then reveals a reward. Actions are often referred as ‘arms’. In the literature, the terms “ $k$ -armed bandits” and “multi-armed bandits” are used interchangeably. Usually, former is used when the number of arms is  $k$  and the later term is used when the number of arms is at least two and the actual number is irrelevant to the discussion (Lattimore & Szepesvari, 2020). Multi-armed bandits are primarily used to address the conflicting objectives of choosing the arm with the best reward history and choosing a less explored arm to learn its reward so that better actions can be taken in future. Contextual bandits are a variant of multi-armed bandits, in which rewards in each round depend on a context observed by the agent prior to making a decision. An agent knows the context and arm/action set but does not know the true reward. Formally contextual bandit problem is defined as follows Slivkins (2019):

At each round (iteration)  $t = 1, 2, 3 \dots, T$

1. The agent observes the current user  $u_t$  and a set  $A$  of arms or actions together with their context  $x_{t,a}$  for  $a \in A$ . Context,  $x_{t,a}$  includes information of both the user  $u_t$  and arm  $a$ .
2. Agent picks an arm  $a \in A$  based on observed rewards or payoffs in previous trials and receives reward  $r_{t,a}$  whose expectation depends on both the user  $u_t$  and the arm  $a$ .
3. The agent then improves its arm-selection strategy with the observed context  $x_{t,a}$ , chosen arm  $a$  and reward received  $r_{t,a}$ .

In linear contextual bandits each arm is characterized by a feature vector  $x_{t,a}$  and the expected reward is linear in this vector for some fixed but unknown vector  $\theta$ . Context can be interpreted as a static context, i.e., a context that does not change from one round to another (Slivkins, 2019).

One of the most straightforward algorithms for solving exploitation-exploration trade-off is  $\epsilon$ -greedy. In each trial  $t$ , the algorithm estimates the average reward of each arm and chooses the greedy arm (arm having highest reward) with probability  $1 - \epsilon$  or a random arm with probability  $\epsilon$  (Lattimore & Szepesvari, 2020).

Explore First also known as explore-then-commit or explore-then-exploit is another algorithm used for solving exploitation-exploration trade-off which selects random arms for a fixed number of times and then exploits by choosing the arm that appeared best during exploration (Lattimore & Szepesvari, 2020). In contrast to  $\epsilon$ -greedy and explore-then-exploit, upper confidence bound algorithms (Auer et al., 2002) use a smarter way to balance exploitation and exploration. Upper Confidence Bound (UCB) algorithm which is based on “optimism in the face of uncertainty” principle assigns a value to each arm known as upper confidence bound which is an overestimate of the unknown reward (Lattimore & Szepesvari, 2020). UCB algorithms estimate both the mean reward  $r_{t,a}$  of each arm  $a$  as well as a corresponding confidence interval value  $c_{t,a}$  so that  $|r_{t,a}^* - r_{t,a}| < c_{t,a}$  holds with high probability where  $r_{t,a}^*$  is actual reward. Then arm having highest upper confidence bound:  $a_t = \operatorname{argmax}_a (UCB_t(a))$  is selected where  $UCB_t(a) = r_{t,a} + c_{t,a}$ .

Although there are many versions of the UCB algorithm, Linear Upper Confidence Bound (LinUCB) is one of the widely used and efficient algorithms for linear contextual bandits (Abbasi-Yadkori et al., 2011) which computes confidence interval efficiently in closed form when the reward model is linear by assuming that the expected reward of an arm  $a$  is linear in its  $d$ -dimensional feature  $x_{t,a}$  with some unknown coefficient vector  $\theta$  (Li et al., 2010). At each round  $t$ , LinUCB chooses action by the following strategy:

$$a_t = \operatorname{argmax}_{a \in A} \{x_{t,a}^T \theta_t + \alpha_t \|x_{t,a}\|_{H_t^{-1}}\} \quad (1)$$



where  $H_t$  is a matrix based on the historical context-arm pairs defined as

$$H_t = \lambda I_d + \sum_{i=1}^t x_{i,a_i} \cdot x_{i,a_i}^T \tag{2}$$

with some  $\lambda > 0$  and  $\alpha_t > 0$  is a tuning parameter that controls the exploration rate in LinUCB.

### 3.2 Siamese neural network

Siamese Neural Network (SNN) was first introduced in early 1990s for signature verification (Bromley et al., 1993). Over the years, SNN has garnered attention from researchers and is mostly used to find relationship or compare two objects. SNN is also known as twin neural network which consists of two or more identical networks with shared weights. These twin networks accept pair of inputs and produce embedding vectors of its respective inputs. These networks are used to find the similarity of the inputs by comparing their feature vectors. For same or similar input patterns, SNN is trained to output a high similarity value and for dissimilar input patterns it is trained to output a low similarity value. Figure 2 illustrates architecture of a SNN.

## 4 Proposed work

In this section, we formulate and define proposed SiameseNN-UCB algorithm for generating reciprocal recommendations in online dating.

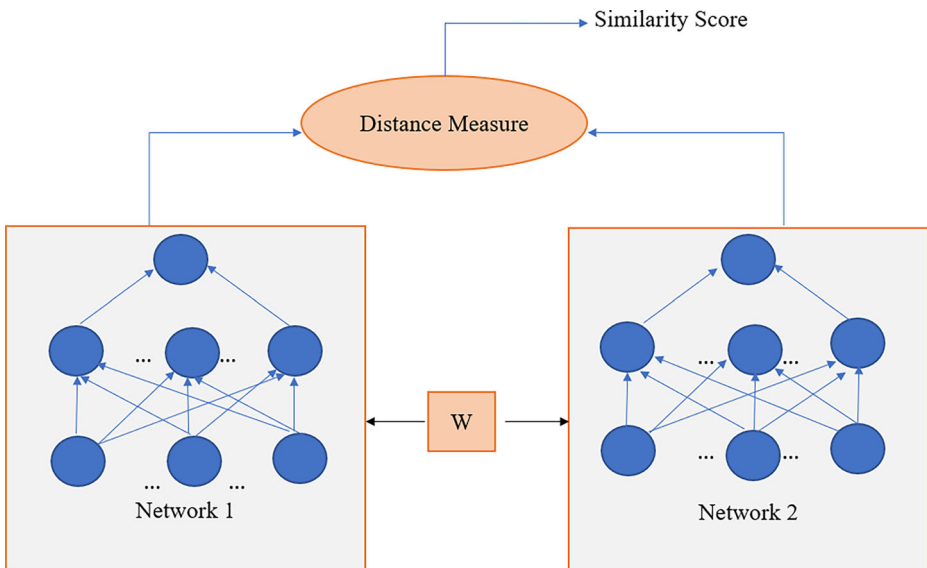


Fig. 2 Architecture of a siamese neural network

#### 4.1 Problem setting and formulation

*Agent:* Reciprocal Recommender System algorithm (SiameseNN-UCB for heterosexual online dating).

*Environment:* Set of users,  $U$  that can be partitioned into two disjoint sets  $U_m$  and  $U_f$  of male and female users respectively.

$$U = \{U_m \cap U_f\} \quad (3)$$

*Arm/action:* At each round  $t$ , choice of a user  $a$  among a finite, but possibly large number of arms i.e.,  $a \in A$  to be recommended as a potential match for user  $u_t$  is an arm or action. Set of arms  $A$  is defined as:

$$A = \left\{ \begin{array}{l} U_f \text{ if } u_t \in U_m \\ U_m \text{ if } u_t \in U_f \end{array} \right\} \quad (4)$$

*Context:* Context refers to multi-criteria ratings of user's self-assessment for various criteria namely attractiveness, sincerity, intelligence, fun and ambition, multi-criteria ratings of user's preferences for attractiveness, sincerity, intelligence, fun, ambition and shared interests in a desired partner, demographic attributes such as age, race, income, country/place of origin and intended profession, dating habits (primary goal and frequency for going on date) and hobbies of users. Context includes this information of both the user as well as the arm.

*Reward/payoff:* A numerical reward of 1 is chosen if the users are a reciprocal match and 0 otherwise.

At round  $t = 1, 2, 3, \dots, T$

1. The agent observes the current user  $u_t$  ( $\in U$ ) and context  $x_{t,u_t}$  of  $u_t$ .
2. Set of arms  $A$  and the context  $x_{t,a}$  for each arm  $a \in A$  are observed by the agent. Information of both the user  $u_t$  and arm  $a$  will be referred as a context  $s_t \in S$  where  $S$  is the context space.
3. Agent picks an arm  $a \in A$  based on observed rewards in previous rounds and receives reward  $r_{t,a}$  whose expectation depends on both the user  $u_t$  and the arm  $a$ .
4. The agent improves its arm-selection strategy with the observed context  $s_t$ , chosen arm  $a$  and reward received  $r_{t,a}$ .

At round  $t$ , for a given set of users  $U$ , we can observe user  $u_t$  ( $\in U$ ) with a  $d$ -dimensional feature vector. Each arm or action  $a \in A$  is associated with a  $d$ -dimensional feature vector  $\langle x_{t,a} \rangle$ . The agent recommends an arm  $a$  to the user  $u_t$  and computes a reward  $r_{t,a}$ . Our goal is to minimize the following regret:

$$R_T = E \left[ \sum_{t=1}^T (r_{t,a}^* - r_{t,a}) \right] \quad (5)$$

where  $r_{t,a}^*$  is actual reward and  $r_{t,a}$  is computed reward at round  $t$ .

Given the feature vectors  $\langle x_{t,u_t} \rangle$ ,  $\langle x_{t,a} \rangle$ , reward  $r_{t,a}$  for any round  $t$  is defined as

$$r_{t,a} = f(\cdot; \langle x_{t,u_t} \rangle, \langle x_{t,a} \rangle, W) \quad (6)$$

which is drawn from an unknown distribution  $f(\cdot; \langle x_{t,u_t} \rangle, \langle x_{t,a} \rangle, W)$  where  $W = (w_1, w_2, \dots, w_l)$  is an unknown weight matrix representing the preferences of users towards other users in the feature space.

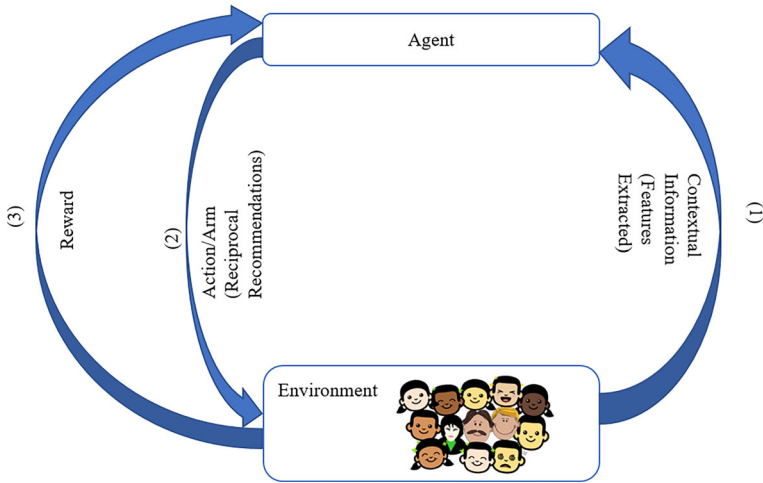


Fig. 3 Agent-environment interaction in contextual bandits adapted for online dating

### 4.2 Proposed multifaceted RRS: SiameseNN-UCB

In this section, we propose SiameseNN-UCB: a deep neural-network based approach that follows Siamese architecture to transform raw features and uses upper confidence bound type exploration to solve exploitation-exploration trade-off for generating reciprocal recommendations in RRS. We have incorporated multiple aspects like multi-criteria ratings of a user, popularity-awareness, demographic information and availability of users. Figure 3 shows a pictorial illustration of agent-environment interaction in contextual bandits based online dating reciprocal recommendation system.

The framework of the proposed multifaceted RRS is depicted in Fig. 4.

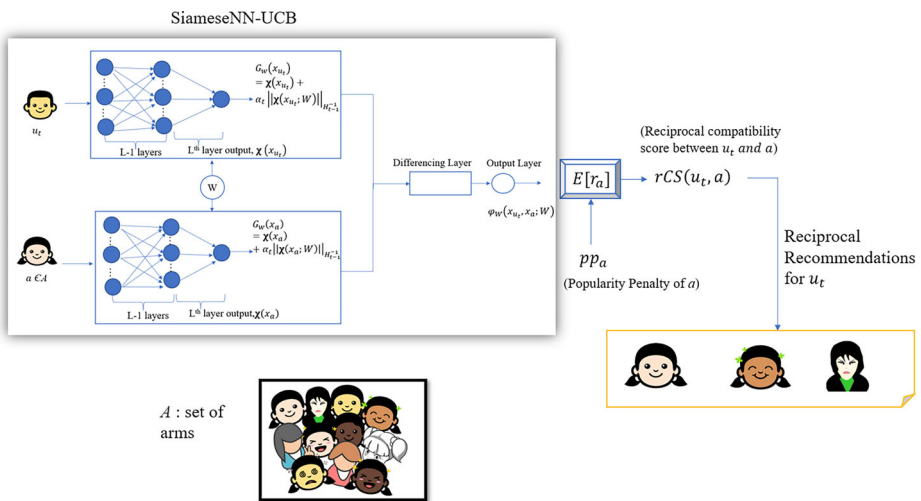


Fig. 4 The framework of proposed multifaceted RRS

Over the years deep neural networks have emerged as a powerful technique to approximate any arbitrary function. Therefore, in this paper we are trying to learn reward generating function (eq. 6) using a deep neural network. In order to learn the reward function  $f(\cdot)$ , we propose to use a deep Siamese neural network so that distance metric learning can be used to drive a similarity metric (Zhang et al., 2016) to be small for potential matches, and large for pairs who are not a potential match. We assume that the reward function  $f(\cdot)$  can be expressed using a feature vector and an exploration parameter. The hidden layers of deep neural network are used to represent the features and the Upper Confidence Bound (UCB) type exploration is performed in the last layer of the neural network.

At round  $t$ , consider a pair of users  $u_t$  and  $a$  such that  $u_t \in U$ ,  $a \in A$ ,  $a \neq u_t$ . Let  $W$  be the shared weight in the Siamese architecture which needs to be learnt. Output of the last hidden layer of Siamese deep neural network is used to transform the raw feature vectors and then UCB-type exploration is performed in the last layer of the neural network. We define  $\varphi_W(x_{u_t}, x_a; W)$  as the output of the Siamese neural network consisting of  $L$  fully connected layers.

$$\varphi_W(x_{u_t}, x_a; W) = (G_w(x_{u_t}) - G_w(x_a)) \quad (7)$$

$\varphi_W(x_{u_t}, x_a; W)$  denotes difference between  $G_w(x_{u_t})$  and  $G_w(x_a)$  where  $G_w(x_{u_t})$  and  $G_w(x_a)$  represents output from two identical subnetworks in Siamese architecture for users  $u_t$  and  $a$  respectively obtained after adding UCB-type exploration defined as:

$$G_w(x_{u_t}) = \chi(x_{u_t}) + \alpha_t \|\chi(x_{u_t}; W)\|_{H_t^{-1}} \quad (8)$$

$$G_w(x_a) = \chi(x_a) + \alpha_t \|\chi(x_a; W)\|_{H_t^{-1}} \quad (9)$$

$\chi(\cdot)$  is the output of the  $L^{th}$  hidden layer of SiameseNN-UCB.

In round  $t$ , our proposed SiameseNN-UCB arm-selection strategy is defined by the following equation:

$$a_t = \operatorname{argmin}_{a \in A} \{ \varphi_W(x_{u_t}, x_a; W) \} \quad (10)$$

which gives the best arm or selected partner for a given user  $u_t$ .  $\|\cdot\|_H$  denotes Mahalanobis distance defined as:

$$\|x\|_H = \sqrt{x^T H x} \quad (11)$$

$\alpha_t > 0$  is an algorithmic parameter controlling the exploration and  $H_t$  is a matrix based on historical transformed features defined as:

$$H_t = \lambda I_d + \sum_{i=1}^t \chi(x_{i,u_i}; W_{i-1}) \cdot \chi(x_{i,a_i}; W_{i-1})^T \quad (12)$$

where  $\lambda > 0$  and  $I_d$  is  $d \times d$  identity matrix.

Algorithm 1 gives a detailed description of the proposed approach.

---

**Algorithm 1** Algorithm for SiameseNN-UCB.

---

**Input** Number of rounds  $T$ , a deep Siamese neural network architecture that defines a parametric function  $f(\cdot; x_{t,u_t}, x_a, W)$  and set of users  $U$ .

**Output** Reciprocal recommendations,  $RR$

- 1: Partition  $U$  in two disjoint sets  $U_m$  and  $U_f$ .
  - 2: **for**  $t = 1, 2, 3 \dots, T$  **do**
  - 3:     Extract  $u_t$ 's profile  $P_{u_t}$
  - 4:     **if**  $u_t \in U_m$  **then**
  - 5:         Extract  $a$ 's profile  $P_a \forall a \in A$  where  $A = U_f$
  - 6:         **for** each multi-criteria attribute  $m \in F_m$  list of multi-criteria attributes **do**
  - 7:             receive feature vector representations of  $u_t$  and  $a$ :
  - 8:              $x_{u_t} = \langle v_{u_t}^{(ms)}, v_{u_t}^{(mp)}, v_{u_t}^{(d)}, v_{u_t}^{(dg)}, I_{u_t} \rangle$
  - 9:              $x_a = \langle v_a^{(ms)}, v_a^{(mp)}, v_a^{(d)}, v_a^{(dg)}, I_a \rangle$
  - 10:             where  $v_{u_t}^{(ms)}, v_{u_t}^{(mp)}, v_{u_t}^{(d)}, v_{u_t}^{(dg)}$  and  $I_{u_t}$  denote  $u_t$ 's self assessment of  $m$ , preference ratings of  $m$ , demographic attributes, dating habits and ratings of hobbies respectively.  $v_a^{(ms)}, v_a^{(mp)}, v_a^{(d)}, v_a^{(dg)}$  and  $I_a$  denote  $a$ 's attributes.
  - 11:             **end for**
  - 12:             Feed the raw feature vectors  $(x_{u_t}, x_a)$  to two identical deep neural networks.
  - 13:             Upper confidence bound value is added in the last layer:
  - 14:              $G_w(x_{u_t}) = \chi(x_{u_t}) + \alpha_t \|\chi(x_{u_t}; W)\|_{H_{t-1}^{-1}}$
  - 15:              $G_w(x_a) = \chi(x_a) + \alpha_t \|\chi(x_a; W)\|_{H_{t-1}^{-1}}$
  - 16:             Reward of an arm computed as:
  - 17:              $\phi_W(x_{u_t}, x_a; W) = (G_w(x_{u_t}) - G_w(x_a))$
  - 18:             Update  $H_t$ :  $H_t = H_{t-1} + \chi(x_{t,u_t}; W_{t-1}) \cdot \chi(x_{t,a}; W_{t-1})^T$
  - 19:             Update the weight parameters of Siamese deep neural network
  - 20:             **end if**
  - 21:     **if**  $u_t \in U_f$  **then**
  - 22:         Extract  $a$ 's profile  $P_a \forall a \in A$  where  $A = U_m$
  - 23:         Perform steps 6-19.
  - 24:     **end if**
  - 25: **end for**
  - 26:  $popS \leftarrow$  ratio of users on whose matches  $a_t$  occur to the total matches.
  - 27:  $pp_a = \left\{ \begin{array}{l} 1 \quad \text{if } popS \in (0, 0.5) \\ popS \quad \text{otherwise} \end{array} \right\}$
  - 28: Reciprocal compatibility score  $rCS(u_t, a)$  that measures the mutual preferences between  $u_t$  and  $a$  is computed as
  - 29:  $rCS(u_t, a) \leftarrow pp_a * E[r_a]$  where  $E[r_a] = \phi_W(x_{u_t}, x_a; W)$
  - 30: Reciprocal recommendations are generated with top  $rCS$
  - 31: Return Reciprocal recommendations,  $RR$ .
-

## 5 Experimental study

In this section, we describe the data set used in our experiments and the performance evaluation of our proposed multifaceted reciprocal recommendation algorithm.

### 5.1 Dataset used

The speed-dating experiment data was combined by Ray Fisman and Sheena Iyengar (Fisman et al., 2006). It is available on Kaggle.com. The dataset has attributes related to demographic information, dating expectations and dating experience. There are 8,378 records in the dataset describing the information for each date.

### 5.2 Dataset analysis

The speed-dating experiment data has total 195 attributes in the dataset. We computed the percentage of missing values in an attribute column in order to prioritize the most important features. If the majority of the values of any attribute is missing, there would be no point in

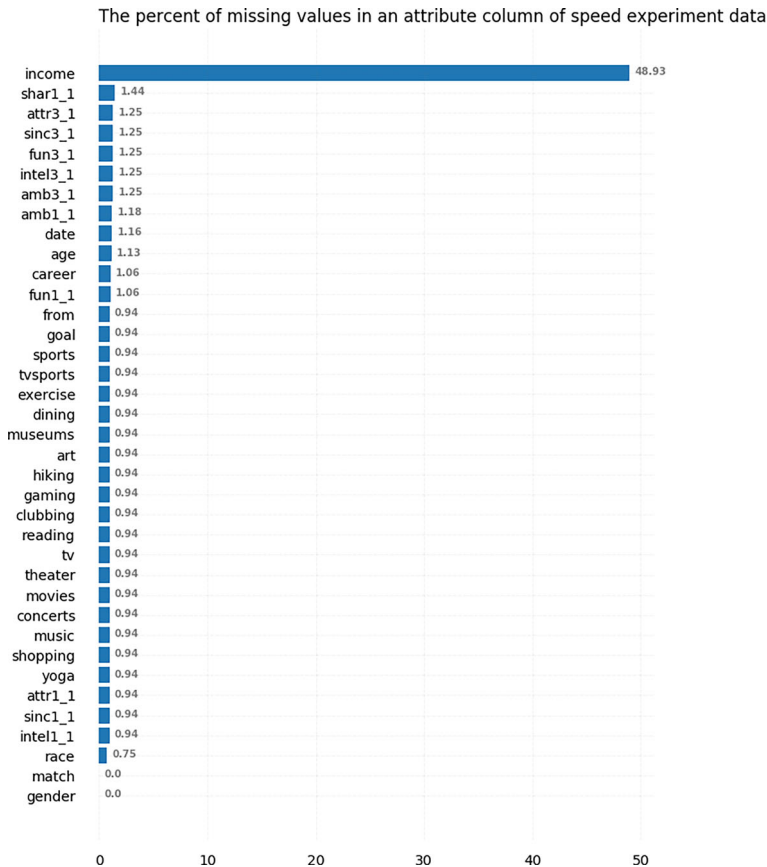
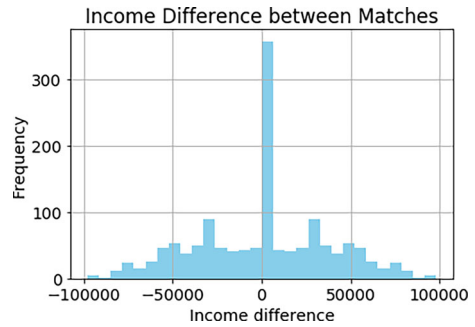


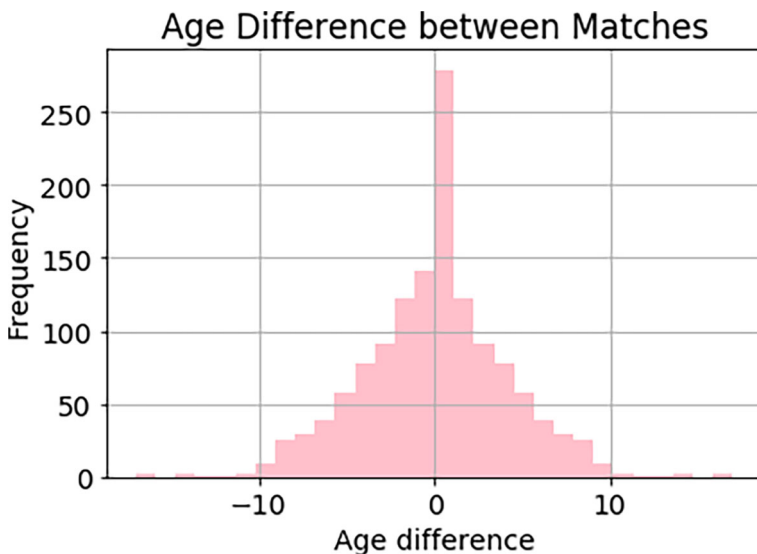
Fig. 5 Percentage of missing values in an attribute column of speed dating experiment data

**Fig. 6** Income difference between matches in speed dating experiment data

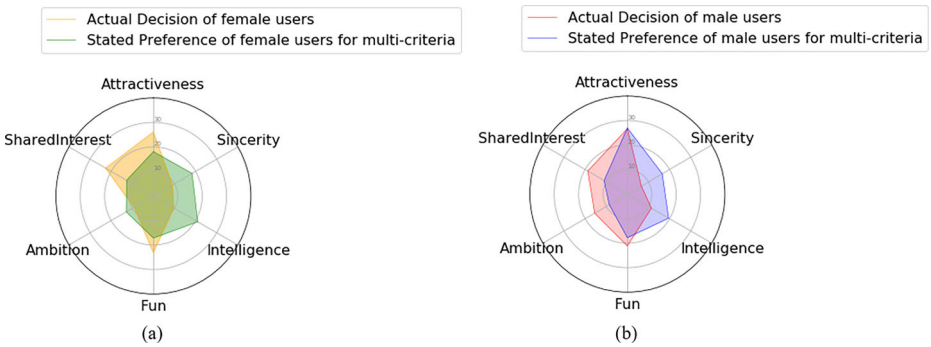


trying to impute values as it increases the likelihood of inaccuracy. Therefore, we only used attributes shown in Fig. 5 as the percentage of missing values for these attribute columns except ‘Income’ as depicted in the figure is negligible. Although ‘Income’ attribute column has about 48.93% missing values, we chose to use it because in the speed-dating experiment data people preferred less difference in income as well as age between matches as illustrated in Figs. 6 and 7.

We also analysed user’s stated preference for multi-criteria and their actual decision to examine the difference between what people state they look for and what actually influence their decision in choosing a match. Figure 8a suggests female user’s stated preferences were drastically different from what actually influenced their decisions in most attributes. Figure 8b shows that male user’s stated preference on the attractiveness and fun matches their actual decision but user’s stated preference on the shared interest, ambition, sincerity and intelligence show considerable deviation from their actual decisions. It suggests that there is a huge gap between what a user stated for multi-criteria attributes during sign up and what they really want. This has encouraged us to consider various aspects such as demographic attributes, dating habits (primary goal and frequency for going on date), hobbies



**Fig. 7** Age difference between matches in speed dating experiment data



**Fig. 8** Attribute comparison of (a) female user’s stated preference for multi-criteria and their actual decision (b) male user’s stated preference for multi-criteria and their actual decision

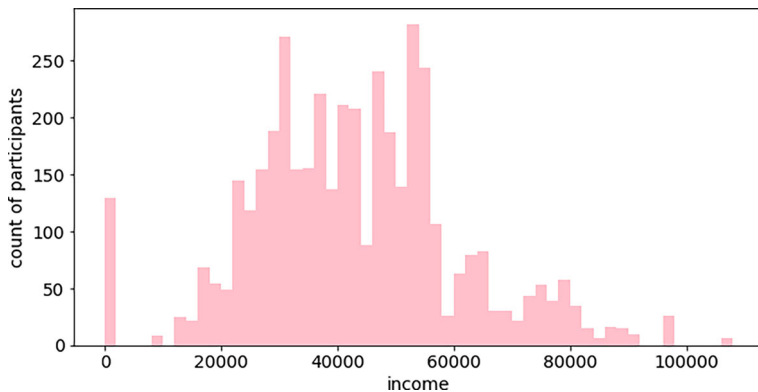
of users, their willingness to explore new choices in addition to self-assessment and preference ratings for multi-criteria namely attractiveness, sincerity, intelligence, fun, ambition and shared interests.

### 5.3 Dataset pre-processing

#### 5.3.1 Handling missing data and discretization

The speed-dating experiment dataset has a very large number of variables (195) and many of them have missing values. Before determining important features that can be fed into the proposed approach, we examined each column to make sure it has enough available data to be used for further analysis. If a feature has mostly missing values, then that feature itself can also be eliminated. We identified features having only a reasonable percentage of values which are missing (shown in Fig. 5). Income attribute column has about 48.93% missing values and it is slightly right skewed distribution as depicted in Fig. 9.

We examined k-nearest neighbors regression, logistic regression, Lasso model fit with least angle regression, kernel ridge regression, Support Vector Regression (SVR) and Multi-layer Perceptron (MLP) regressor to assess the performance of these regression models for



**Fig. 9** Distribution of income of participants in speed-dating experiment data



predicting ‘Income’ attribute values. We used three hidden layers with MLP regressor and RELU activation function to enhance the learning capability of neural network. The number of neurons were tuned to {96,64,32}, L2 penalty (regularization term) parameter was set to 0.00001 and ‘lbfgs’ was used as solver for weight optimization as better performance was achieved with these parameters. In our experiments for SVR, the Radial Basis Function (RBF) kernel exhibited better performance and learning capability. Evaluation for selection of the optimal predictive regression model was done using 10-folds cross-validation where data was randomly partitioned into 10 disjoint subsets of approximately equal size. Here, “fold” refers to the number of resulting subsets. The model was trained using 9 folds which, together, represent the training set. Then, the model was applied to the remaining subset, which is known as the test set, and the performance was measured. This procedure was repeated until each of the 9 subsets was used as test set. The average of the 9 performance measurements on the 9 test sets is the cross-validated performance. We found that predicting missing ‘Income’ attribute column values using scikit-learn’s MLP regressor was the best strategy on the dataset used with carefully tuned parameters. Comparison results of the Mean Absolute Error (MAE) of these regression models are shown in Table 1. As it can be observed from the Table 1, MLP regressor outperforms other regressors. We have also plotted scatter plot of true vs predictive values using these regressors in Fig. 10 for data visualization. Ideally, all the points should be close to a regressed diagonal line. We observed that the MLP regressor predicted values are closest to this diagonal line with very few outliers compared to other regressors.

Numeric attributes namely income, age, race, country/place of origin and intended profession, dating habits (primary goal and frequency for going on date) were discretised. Discretization involves transforming a continuous-valued variable into a discrete one by creating a set of contiguous intervals that spans the range of the variable’s values. For example, age is discretised into groups, such as Age<20, Age20-25, Age25-30, Age30-35, Age35-40, Age>40 years. Speed dating experiment data suffers from skewed class proportions, the number of observations per class is not equally distributed, only about 16% of pairs resulted in a match. As illustrated in Fig. 11 we have a large number of observations for one class (“No Match”), and much fewer observations for other class (“Match”). Class balancing becomes imperative as we are more concerned with minority class (“Match”). To tackle skewed class proportions, we used Synthetic Minority Over-sampling Technique (SMOTE) which is a data-level method to balance the training data by creating synthetic minority samples (Johnson & Khoshgoftaar, 2019).

SMOTE restores the influence of the minority samples by boosting their numbers. For each minority sample, the algorithm finds its k-nearest-neighbors and generates synthetic

**Table 1** Comparison analysis of k-nearest neighbors regression, logistic regression, LassoLARS, kernel ridge regression, Support Vector Regression (SVR) and Multi-layer Perceptron (MLP) regressor

Regressor model	MAE
k-nearest neighbors regression	451.3226
Logistic regression	708.3011
Lasso model fit with least angle regression (LassoLARS)	5743.7778
Kernel ridge regression	7685.6216
Support Vector Regression (SVR) with RBF kernel	12461.2049
Multi-layer Perceptron (MLP) regressor	157.0388

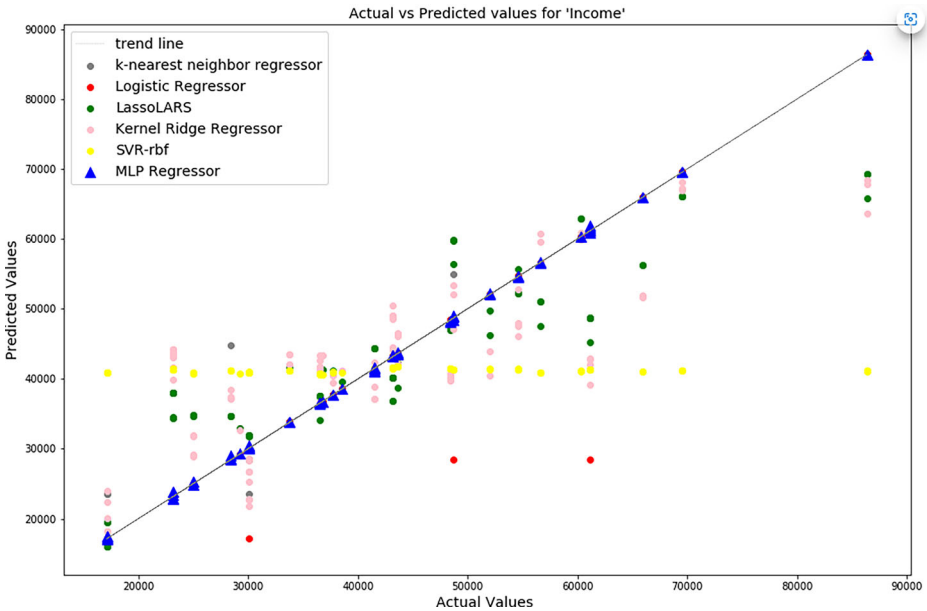


Fig. 10 Scatter plot of Actual vs Predicted values for 'Income' attribute

examples of the minority sample at a random location along the line segments joining any/all of the  $k$  minority class nearest neighbors. We used the SMOTE implementation provided by the imbalanced-learn Python library. The parameters denoting `sampling_strategy` and `number of nearest neighbours` were tuned to 0.3 and 5 respectively for the dataset as we obtained better performance with these values. We used 10-folds Cross Validation for model validation. SMOTE was used to balance the training data only. Test set follow the same distribution as the original dataset and its patterns were not considered in oversampling or training phases. Since, the patterns included in the test set are never oversampled, thus it allows us to have a proper evaluation of the model's capability to generalize from the training data. Test set was not used in oversampling to tackle/minimize the issues of

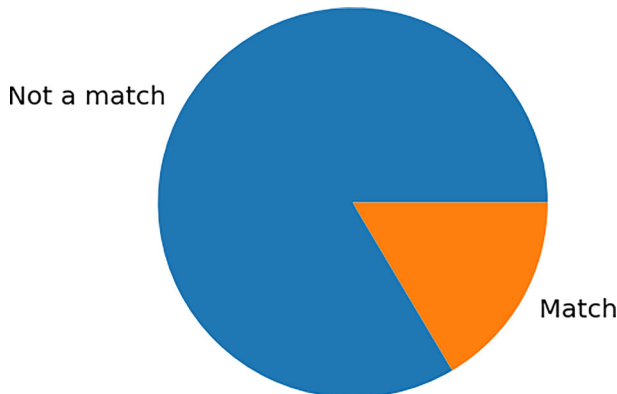


Fig. 11 Match distribution in speed dating experiment data

overoptimism and overfitting regarding the joint-use of Cross Validation and oversampling approaches. To address over-fitting, we performed cross-validation, used L2 regularization and dropout layers.

### 5.3.2 Normalization

Next step in the pre-processing phase involved normalizing the attribute values. The values of all the attributes were normalized to bring them in a uniform range of [0, 1]. In the dataset we have multi-criteria attribute ratings on a scale of 1-10 or 1-100. To normalize the values of multi-criteria ratings, we divided the values by highest score to get relative score in that category.

## 5.4 Experimental analysis

### 5.4.1 Evaluation metrics

To evaluate the performance of our proposed algorithm, we used accuracy, precision, recall and F1-measure. Accuracy is defined as the proportion of correctly predicted matches to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives are described as: True Positive cases occur when reciprocal match is correctly predicted. True Negative is an outcome where a pair of users which is not reciprocal match is correctly predicted as not a match. False Positive cases are those where a pair of users which is not reciprocal match is incorrectly predicted as a match and False Negative is an outcome where reciprocal match is incorrectly predicted as not a match.

Precision@K is defined as the proportion of retrieved recommended items in top-K set that are relevant. Recall@K is defined as the proportion of relevant items found in top-K retrieved recommendations. Precision@K and Recall@K are defined as follows:

$$Precision@K = \frac{|relevant \cap retrieved|}{|retrieved|} \quad (14)$$

$$Recall@K = \frac{|relevant \cap retrieved|}{|relevant|} \quad (15)$$

F1-score is defined as the harmonic mean of precision and recall and is computed using the formula:

$$F1 - score@K = \frac{2 * Precision@K * Recall@K}{Precision@K + Recall@K} \quad (16)$$

### 5.4.2 Experimental results

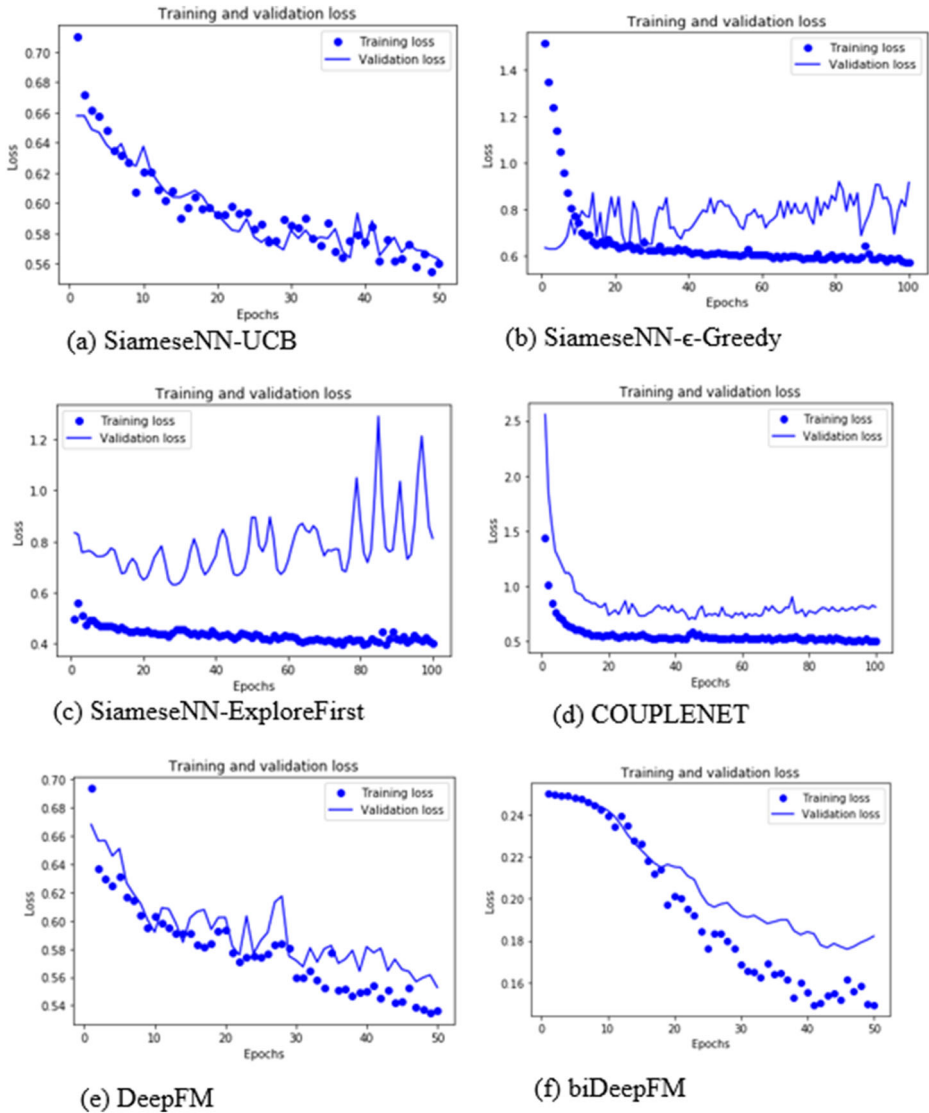
All empirical evaluation was conducted on speed-dating experiment data. We compared the proposed approach SiameseNN-UCB with following state-of-the-art algorithms:

- RECON (Pizzato et al., 2010): It is a non-neural technique but one of the best-known contents based RRS for online dating and most of the RRS models proposed later are based on it. Unlike RECON, speed-dating experiment data does not have data such as messages exchanged. Instead, we used a 1 (true) or 0 (false) value to indicate whether a user has dated with another one.

- Zheng et al. (2018): It is among one of the recent content based RRS models for online dating on the same dataset that we used in this paper. It is a non-neural technique based on multi-criteria utility theory and utility is computed as the similarity between user expectation and dating experience.
- COUPLENET (Tay et al., 2018) is a deep learning-based network for estimating the compatibility of two users. Network proposed in the paper follows a ‘Siamese’ architecture, with shared parameters for each side of the network. Gated Recurrent Units (GRU) encoders with attentional pooling were used to learn feature vectors.
- DeepFM (Guo et al., 2017) integrates factorization machines for recommendation and deep learning for feature learning in a neural network architecture. Deep neural network is used to model the high-order feature interactions and low-order feature interactions are modelled with factorization machine. We have implemented DeepFM on speed experiment data. Our training samples consist of  $n$  instances  $(\chi, y)$  where  $\chi$  is an  $m$ -fields data record involving a pair of users  $u$  and  $v$ , and  $y = \{0, 1\}$  is the associated label indicating whether user  $u$  is a reciprocal match of user  $v$  or not.
- biDeepFM (Yildirim et al., 2021) is a multi-objective learning approach for online recruiting that fits into reciprocal recommendation problems. In this paper, a prediction model  $(y'_u, y'_v) = f(x^{(i)})$  to jointly estimate the probability of a candidate applying for a specific job and the probability of a recruiter calling that specific candidate was learned. However, the data used in this work is not publicly available. Therefore, we have implemented biDeepFM on speed experiment data to jointly estimate the probability of a user  $u$  being interested in user  $v$  and user  $v$  being interested in user  $u$ .  $x^{(i)}$  is the concatenation of features of both the users ( $u$  and  $v$ ). To learn  $y'_u$  and  $y'_v$  we used individual preferences (i.e., the Yes/No decisions for each partner which was observed from “dec” attribute in the dataset). We combined the unilateral preferences  $y'_u$  and  $y'_v$  obtained from biDeepFM using harmonic mean. We chose harmonic mean to combine unilateral preference scores to get a single reciprocal score because aggregated results obtained by harmonic mean is closer to the minimum of its inputs, which is a fundamental requirement in RRS that both users should be interested in one another, in order to produce successful reciprocal recommendations (Kumari et al., 2021).
- SiameseNN- $\epsilon$ -Greedy is a variant of proposed approach that follows Siamese architecture to transform raw features and selects a random arm (explore) with probability  $\epsilon$ , or the action with highest estimated reward with probability  $1 - \epsilon$ .
- SiameseNN-ExploreFirst is another variant of proposed approach that follows Siamese architecture to transform raw features and selects random arms (explore) for the first  $N$  predictions and after that only the best arm is selected.

SiameseNN-UCB (proposed approach), SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst (variants of proposed approach), COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017) and biDeepFM (Yildirim et al., 2021) are deep neural network-based approaches. All these approaches were implemented in Tensorflow with Python. We trained SiameseNN-UCB, SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst, COUPLENET (Tay et al., 2018) and DeepFM (Guo et al., 2017) using Adam (Kingma & Ba, 2014) optimizer with a learning rate of  $10^{-2}$  since this learning rate consistently produced the best results across these approaches. The batch size was tuned amongst 16, 32, 64. All these models were trained for 20, 50, 100, 250 epochs and all the learnable parameters were initialized with normal distributions with a mean of 0 and standard deviation of 0.1. With Early stopping call-back function, we stopped training when there was no improvement in the validation loss values for fifteen consecutive epochs. We used 15 as the patience value.

Batch normalization was used to normalize the activations and a dropout of 0.3 was used for SiameseNN-UCB, SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst and COUPLENET. For DeepFM, we kept the Factorization Machine part unchanged. We used a multi-layer perceptron block of depth 3 with relu activation function, 0.8 dropout and “constant” network shape (128-128-128) because “constant” network shape was empirically better than the other options (increasing, decreasing or diamond) (Guo et al., 2017). We adapted same architecture for biDeepFM except the sigmoid extension as output for each objective.



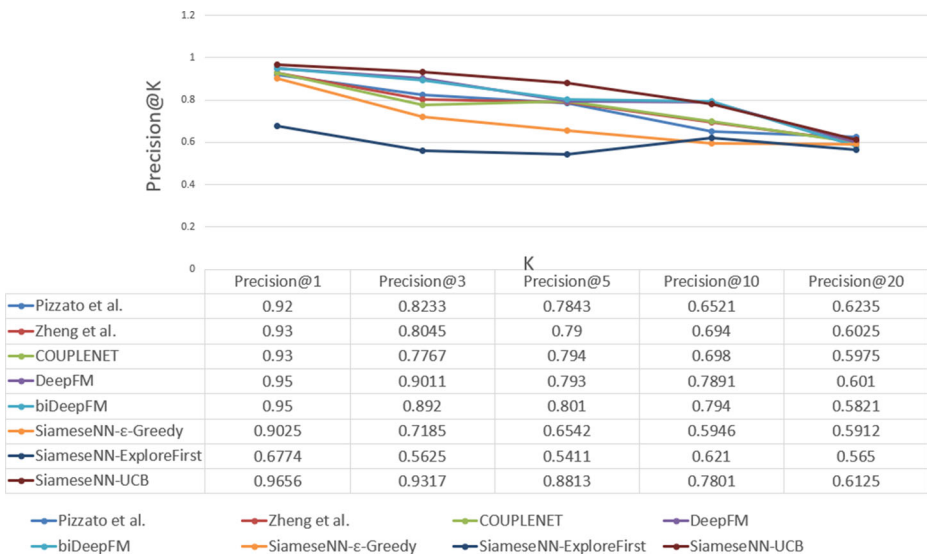
**Fig. 12** Training and validation loss of neural-network based approaches (a) SiameseNN-UCB, (b) SiameseNN- $\epsilon$ -Greedy, (c) SiameseNN-Explore First (d) COUPLENET (Tay et al., 2018), (e) DeepFM (Guo et al., 2017) and (f) biDeepFM (Yıldırım et al., 2021)

Weighted LogLoss with gradient descent was used during the training of biDeepFM as in Yıldırım et al. (2021).

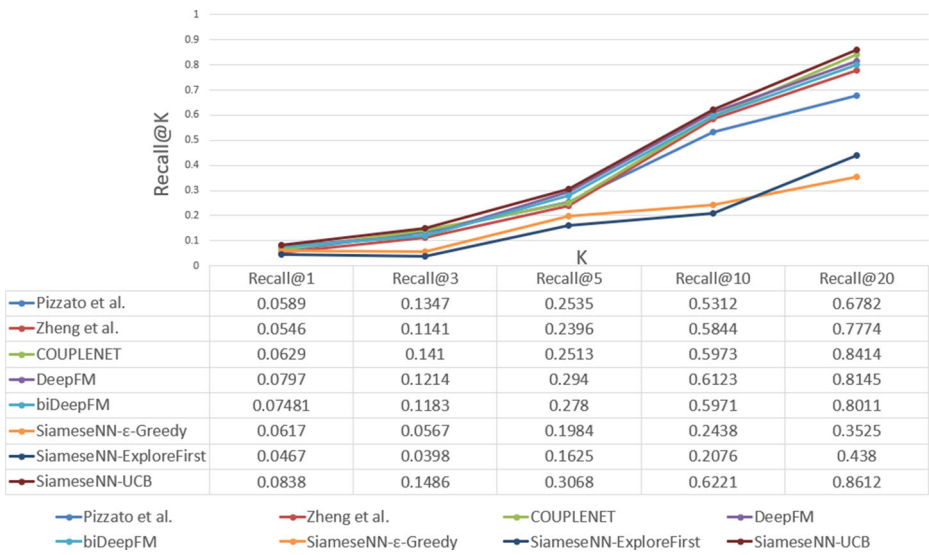
Figure 12 shows the training and validation loss of neural network-based approaches namely SiameseNN-UCB, SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst, COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017) and biDeepFM (Yıldırım et al., 2021) when these models were trained with a batch size of 64 for {50,100} epochs. We can observe from the figure that although for SiameseNN-ExploreFirst and COUPLENET learning curve for training loss and validation loss show improvement but a large gap remains between these curves indicating validation dataset does not provide sufficient information to evaluate the ability of the model to generalize. The learning curve for training loss and validation loss of SiameseNN- $\epsilon$ -Greedy shows some noisy movements around the training loss. The learning curve of DeepFM suggests that the model generalizes well with some noisy movements around the training loss. Learning curve for biDeepFM shows that the plot of validation loss decreases to a point and begins increasing again. However, in SiameseNN-UCB learning curve of validation loss decreases to a point of stability and has a small gap with the training loss.

Precision@K, recall@K and F1-score@K were computed with threshold value of 0.7 (top-K most similar matches having similarity score  $\geq 0.7$ ) for K= 1, 3, 5 and 10. We lowered the threshold value to 0.6 for obtaining top-K (K=20) recommendation results.

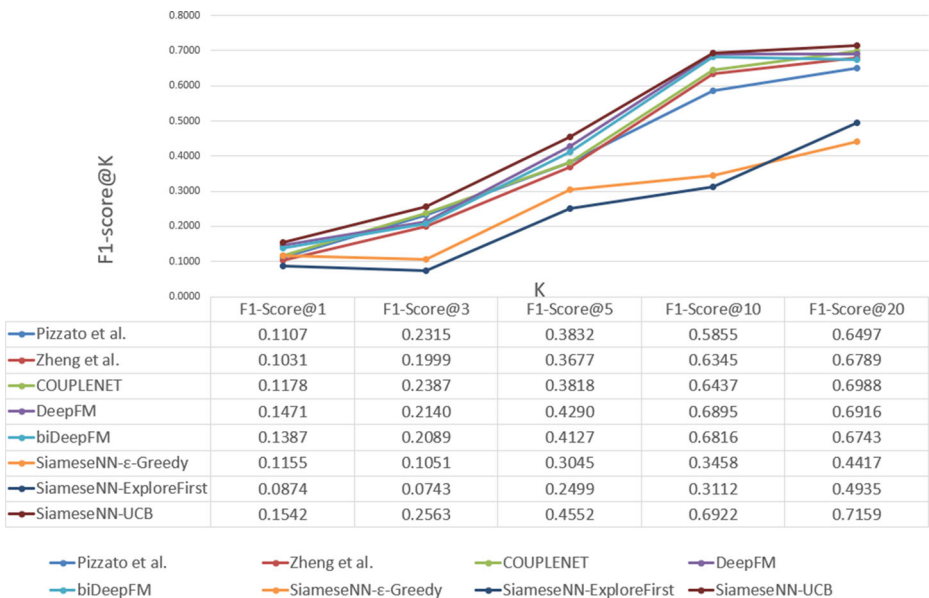
We observed a trade-off between precision and recall, recall was increasing but precision dropped when we retrieved more recommendation results. From Figs. 13 and 14, we observe that (a) Precision decreases when K increases: This is expected because as the value of K increases, number of top-K recommendation results increases but new results may include



**Fig. 13** Precision@K obtained with Pizzato et al. (2010) and Zheng et al. (2018), COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017), biDeepFM (Yıldırım et al., 2021), SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst and SiameseNN-UCB for varying K (K=1, 3, 5, 10 and 20)



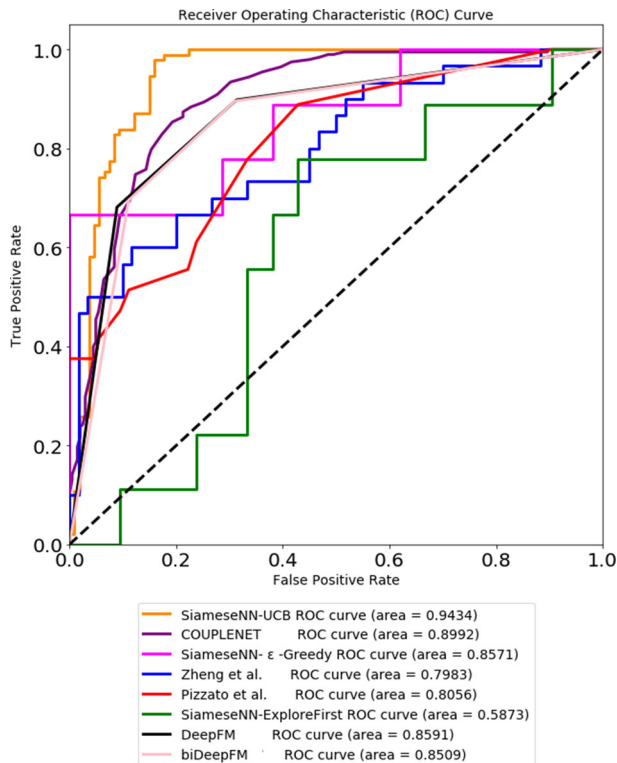
**Fig. 14** Recall@K obtained with Pizzato et al. (2010) and Zheng et al. (2018), COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017), biDeepFM (Yıldırım et al., 2021), SiameseNN-ε-Greedy, SiameseNN-ExploreFirst and SiameseNN-UCB for varying K (K=1, 3, 5, 10 and 20)



**Fig. 15** F1-score@K obtained with Pizzato et al. (2010) and Zheng et al. (2018), COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017), biDeepFM (Yıldırım et al., 2021), SiameseNN-ε-Greedy, SiameseNN-ExploreFirst and SiameseNN-UCB for varying K (K=1, 3, 5, 10 and 20)

more false positives. (b) Recall increases when  $K$  increases: This is also expected because Recall@  $K$  measures how many relevant recommendations are returned. With increasing  $K$ , number of true positives tend to increase. From Figs. 13–15, it is evident that proposed approach SiameseNN-UCB outperforms in terms of precision, recall and F1-score.

Figure 16 shows ROC curve (Receiver Operating Characteristic), defined with respect to minority class (“Match”) for Pizzato et al. (2010) and Zheng et al. (2018), COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017), biDeepFM (Yıldırım et al., 2021), SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst and SiameseNN-UCB. DeepFM and biDeepFM have almost identical performance in terms of evaluation criteria used. This is because we used DeepFM to learn reciprocal matches and biDeepFM to learn two unilateral preference scores which were eventually combined using harmonic mean. However, as clearly evident from the Fig. 16, the proposed SiameseNN-UCB achieves the highest Area Under the ROC curve value which highlights its effectiveness in correctly identifying the observed class (“Match”). Our findings suggest that learning mutual preferences with contextual information and exploratory aspects improves the generated reciprocal recommendations.



**Fig. 16** ROC curve for Pizzato et al. (2010) and Zheng et al. (2018), COUPLENET (Tay et al., 2018), DeepFM (Guo et al., 2017), biDeepFM (Yıldırım et al., 2021), SiameseNN- $\epsilon$ -Greedy, SiameseNN-ExploreFirst and SiameseNN-UCB



## 6 Conclusion and future work

RRS face exploitation-exploration trade-off which present dilemma on choices, i.e., whether to select best decisions based on known information or gather more information about the environment. In this paper, we formulated reciprocal recommendation generation task in online dating as a contextual bandit problem to solve this inherent exploitation-exploration trade-off in RRS. We proposed a deep neural network-based algorithm, SiameseNN-UCB, that follows Siamese architecture to transform raw features and performed upper confidence bound type exploration in the last layer of the neural network. With the proposed approach we incorporated multiple aspects such as demographical information, hobbies, exploratory behaviour, primary goal and frequency of dating in addition to user's given preferences for partner while generating reciprocal recommendations for any user. Experimental study conducted with speed dating experiment data set and comparison with existing state-of-the-art approaches demonstrate the effectiveness of the proposed approach. It was observed that SiameseNN-UCB achieved higher recall, precision and F1 scores as compared to the existing algorithms emphasizing the importance of considering exploratory aspects along with their contextual information for generating successful reciprocal recommendations in online dating. In the future, the proposed approach will be validated on datasets from other reciprocal domains to help better understand the generalization of the method.

**Author Contributions** Prof. Punam Bedi: Supervision, Reviewing, Editing, Resources

Dr. Ravish Sharma: Supervision, Writing, Review, Editing, Project Administration, Resources, Validation

Ms Tulika Kumari: Conceptualization, Methodology, Software, Resources, Validation, Investigation, Data curation, Formal analysis. Writing-Original draft preparation, Visualization, Writing- Reviewing and Editing.

**Funding** \*No funding was received to assist with the preparation of this manuscript.

**Availability of data and materials** No datasets were generated. Dataset used in the current study is publicly available.

**Code Availability** 'Yes'

### Declarations

**Ethics approval** 'Yes'

**Consent to participate** 'Yes'

**Consent for Publication** 'Yes'

**Conflict of Interests** The authors declare that they have no conflict of interest.

## References

- Abbasi-Yadkori, Y., Pál, D., & Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24, 2312–2320.
- Akehurst, J., Koprinska, I., Yacef, K., Pizzato, L., Kay, J., & Rej, T. (2011). Ccr—a content-collaborative reciprocal recommender for online dating. In *Twenty-second international joint conference on artificial intelligence*.

- Alanazi, A., & Bain, M. (2016). A scalable people-to-people hybrid reciprocal recommender using hidden markov models. 2nd Int. Work. Mach. Learn. Methods Recomm Syst.
- Altulyan, M. S., Huang, C., Yao, L., Wang, X., & Kanhere, S.S. (2021). Contextual bandit learning for activity-aware things-of-interest recommendation in an assisted living environment. In *ADC* (pp. 37–49).
- Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2), 235–256.
- Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R.E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1), 48–77.
- Balakrishnan, A., Bouneffouf, D., Mattei, N., & Rossi, F. (2018). Using contextual bandits with behavioral constraints for constrained online movie recommendation. In *IJCAI* (pp. 5802–5804).
- Barto, A. G., & Anandan, P. (1985). Pattern-recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics* (3), 360–375.
- Bouneffouf, D., & Rish, I. (2019). A survey on practical applications of multi-armed and contextual bandits. arXiv preprint arXiv:1904.10040.
- Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., & Shah, R. (1993). Signature verification using a “siamese” time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04), 669–688.
- Fisman, R., Iyengar, S. S., Kamenica, E., & Simonson, I. (2006). Gender differences in mate selection: Evidence from a speed dating experiment. *The Quarterly Journal of Economics*, 121(2), 673–697.
- Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: A factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247.
- He, X., An, B., Li, Y., Chen, H., Guo, Q., Li, X., & Wang, Z. (2020). Contextual user browsing bandits for large-scale online mobile recommendation. In *Fourteenth ACM conference on recommender systems* (pp. 63–72).
- Intayoad, W., Kamyod, C., & Temdee, P. (2020). Reinforcement learning based on contextual bandits for personalized online learning recommendation systems. *Wireless Personal Communications*, 1–16.
- Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 1–54.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Kumari, T., Sharma, R., & Bedi, P. (2021). Multifaceted reciprocal recommendations for online dating. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1–6). IEEE.
- Langford, J., & Zhang, T. (2007). The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems*, 20(1), 96–1.
- Lattimore, T., & Szepesvari, C. (2020). *Bandit algorithms*. Cambridge University Press.
- Li, C.-T. (2019). Mentor-spotting: Recommending expert mentors to mentees for live trouble-shooting in codemontor. *Knowledge and Information Systems*, 61(2), 799–820.
- Li, L., Chu, W., Langford, J., & Schapire, R.E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (pp. 661–670).
- Li, L., & Li, T. (2012). Meet: A generalized framework for reciprocal recommender systems. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (pp. 35–44).
- Liu, W., Li, S., & Zhang, S. (2018). Contextual dependent click bandit algorithm for web recommendation. In *International Computing and Combinatorics Conference* (pp. 39–50). Springer.
- Liu, B., Wei, Y., Zhang, Y., Yan, Z., & Yang, Q. (2018). Transferable contextual bandit for cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- Malinowski, J., Keim, T., Wendt, O., & Weitzel, T. (2006). Matching people and jobs: A bilateral recommendation approach. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, (Vol. 6 pp. 137–137). IEEE.
- Neve, J., & Palomares, I. (2019). Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems* (pp. 219–227).
- Neve, J., & Palomares, I. (2020). Hybrid reciprocal recommender systems: Integrating item-to-user principles in reciprocal recommendation. In *Companion Proceedings of the Web Conference 2020* (pp. 848–853).
- Pilani, A., Mathur, K., Agrawal, H., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2021). Contextual bandit approach-based recommendation system for personalized web-based services. *Applied Artificial Intelligence*, 35(7), 489–504.
- Pizzato, L., Rej, T., Chung, T., Koprinska, I., & Kay, J. (2010). Recon: A reciprocal recommender for online dating. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (pp. 207–214).

- Pizzato, L., Rej, T., Chung, T., Koprinska, I., Yacef, K., & Kay, J. (2010). Reciprocal recommender system for online dating. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (pp. 353–354).
- Potts, B. A., Khosravi, H., Reidsema, C., Bakharia, A., Belonogoff, M., & Fleming, M. (2018). Reciprocal peer recommendation for learning purposes. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge* (pp. 226–235).
- Rodríguez-García, M. Á., Valencia-García, R., Colomo-Palacios, R., & Gómez-Berbís, J.M. (2019). Blind-date recommender: A context-aware ontology-based dating recommendation platform. *Journal of Information Science*, 45(5), 573–591.
- Santana, M. R., Melo, L. C., Camargo, F. H., Brandão, B., Soares, A., Oliveira, R. M., & Caetano, S. (2020). Contextual meta-bandit for recommender systems selection. In *Fourteenth ACM conference on recommender systems* (pp. 444–449).
- Shen, Y., Deng, Y., Ray, A., & Jin, H. (2018). Interactive recommendation via deep neural memory augmented contextual bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 122–130).
- Slivkins, A. (2019). Introduction to multi-armed bandits. arXiv preprint arXiv:1904.07272.
- Tay, Y., Tuan, L. A., & Hui, S.C. (2018). Couplenet: Paying attention to couples with coupled attention for relationship recommendation. In *Twelfth International AAAI Conference on Web and Social Media*.
- Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4), 285–294.
- Wenxing, H., Yiwei, C., Jianwei, Q., & Yin, H. (2015). ihr+: A mobile reciprocal job recommender system. In *2015 10th International Conference on Computer Science & Education (ICCSE)* (pp. 492–495). IEEE.
- Xu, X., Dong, F., Li, Y., He, S., & Li, X. (2020). Contextual-bandit based personalized recommendation with time-varying user interests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, (Vol. 34 pp. 6518–6525).
- Yang, M., Li, Q., Qin, Z., & Ye, J. (2020). Hierarchical adaptive contextual bandits for resource constraint based recommendation. In *Proceedings of The Web Conference 2020* (pp. 292–302).
- Yang, S., Wang, H., Zhang, C., & Gao, Y. (2020). Contextual bandits with hidden features to online recommendation via sparse interactions. *IEEE Intelligent Systems*, 35(5), 62–72.
- Yu, M., Zhang, X., & Kreager, D. (2016). New to online dating? learning from experienced users for a successful match. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 467–470). IEEE.
- Yıldırım, E., Azad, P., & Ögüdücü, Ş.G. (2021). bideepfm: A multi-objective deep factorization machine for reciprocal recommendation. *Engineering Science and Technology, an International Journal*, 24(6), 1467–1477.
- Zang, X., Yamasaki, T., Aizawa, K., Nakamoto, T., Kuwabara, E., Egami, S., & Fuchida, Y. (2017). You will succeed or not? matching prediction in a marriage consulting service. In *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)* (pp. 109–116). IEEE.
- Zhang, C., Liu, W., Ma, H., & Fu, H. (2016). Siamese neural network based gait recognition for human identification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2832–2836). IEEE.
- Zhang, X., Xie, H., Li, H., & CS Lui, J. (2020). Conversational contextual bandit: Algorithm and application. In *Proceedings of The Web Conference 2020* (pp. 662–672).
- Zhang, X., Zhou, Q., He, T., & Liang, B. (2018). Con-cname: A contextual multi-armed bandit algorithm for personalized recommendations. In *International Conference on Artificial Neural Networks* (pp. 326–336). Springer.
- Zheng, Y., Dave, T., Mishra, N., & Kumar, H. (2018). Fairness in reciprocal recommendations: A speed-dating study. In *Adjunct publication of the 26th conference on user modeling, adaptation and personalization* (pp. 29–34).