An Accuracy-Assured Privacy-Preserving Recommender System for Internet Commerce

Zhigang Lu, Hong Shen
The University of Adelaide
\{zhigang.lu, hong.shen\}@adelaide.edu.au

Abstract

Recommender systems, tool for predicting users’ potential preferences by computing history data and users’ interests, show an increasing importance in various Internet applications such as online shopping. As a well-known recommendation method, neighbourhood-based collaborative filtering has attracted considerable attention recently. The risk of revealing users’ private information during the process of filtering has attracted noticeable research interests. Among the current solutions, the probabilistic techniques have shown a powerful privacy preserving effect. The existing methods deploying probabilistic methods are in three categories, one \[19\] adds differential privacy noises in the covariance matrix; one \[1\] introduces the randomisation in the neighbour selection process; the other \[29\] applies differential privacy in both the neighbour selection process and covariance matrix. When facing \(k\) Nearest Neighbour (\(k\)NN) attack, all the existing methods provide no data utility guarantee, for the introduction of global randomness. In this paper, to overcome the problem of recommendation accuracy loss, we propose a novel approach, Partitioned Probabilistic Neighbour Selection, to ensure a required prediction accuracy while maintaining high security against \(k\)NN attack. We define the sum of \(k\) neighbours’ similarity as the accuracy metric \(\alpha\), the number of user partitions, across which we select the \(k\) neighbours, as the security metric \(\beta\). We generalise the \(k\) Nearest Neighbour attack to \(\beta k\) Nearest Neighbours attack. Differing from the existing approach that selects neighbours across the entire candidate list randomly, our method selects neighbours from each exclusive partition of size \(k\) with a decreasing probability. Theoretical and experimental analysis show that to provide an accuracy-assured recommendation, our Partitioned Probabilistic Neighbour Selection method yields a better trade-off between the recommendation accuracy and system security.

Keywords: Privacy preserving, Differential privacy, Neighbourhood-based collaborative filtering recommender systems, Internet Commerce
1 Introduction

Recommender systems predict users’ potential preferences by aggregating history data and users’ interests. Recently, an increasing importance of recommender systems has been shown in various Internet applications. For example, Amazon has been receiving benefits for a decade from the recommender systems by providing personal recommendation to their customers, and Netflix posted a one million U.S. dollars award for improving their recommender system to make their business more profitable [10,15,25]. Currently, in recommender systems, Collaborative Filtering (CF) is a famous technology with three main popular techniques [17], i.e., neighbourhood-based methods [14], association rules based prediction [24], and matrix factorisation [16]. Among these techniques, neighbourhood-based methods are the most widely used in the industry because of its easy implementation and high prediction accuracy.

One of the most popular neighbourhood-based method is $k$ Nearest Neighbour ($k$NN) which provides recommendations by aggregating the opinions of a user’s $k$ nearest neighbours [2]. Although $k$NN recommender systems present very good performance of recommendation accuracy efficiently, the risk of revealing users’ private information during the process of filtering is still a growing concern, e.g., the $k$NN attack presented by Calandrino et al. [5] exploits the property that the users are more similar when sharing same rating on corresponding items to reveal user’s private data. Thus presenting an efficient privacy preserving neighbourhood-based CF algorithm against $k$NN attack, which achieves a trade-off between the system security and recommendation accuracy, has been a natural research interest.

The literature in CF recommender systems has developed several approaches to preserve users’ privacy. Generally, cryptographic, obfuscation, perturbation, probabilistic methods and differential privacy are applied [29]. Among them, cryptographic methods [11,21] provide the most reliable security but the unnecessary computational cost cannot be ignored. Obfuscation methods [22,27] and Perturbation methods [3,4] introduce designed random noise into the original matrix to preserve customers’ sensitive information; however the magnitude of noise is hard to calibrate in these two types of methods [9,29]. The probabilistic methods [1] provided a similarity based weighted neighbour selection of the $k$ nearest neighbours. Similar to perturbation, McSherry et al. [19] presented a naive differential privacy method which adds calibrated noise into the covariance (similarity between users/items) matrix. Similar to the probabilistic neighbour selection [1], Zhu et al. [29] proposed a Private Neighbour Selection to preserve privacy against $k$NN attack by introducing differential privacy in selecting the $k$ nearest neighbours randomly (also adding noise into covariance matrix with differential privacy). Although the methods in [1,19,29] successfully preserve users’ privacy against $k$NN attack, the low prediction accuracy due to
the global randomness should be noted. Even worse, \cite{29} failed to maintain
differential privacy in the process of neighbour selection. Therefore, none
of the existing privacy preserving CF recommender systems can provide
enough utility while preserving users’ private information.

**Motivation.** The current privacy preserving neighbourhood-based CF
methods did not guarantee the data utility against $k$NN attack. Therefore,
in this paper, we aim to present a privacy preserving neighbourhood-based
CF recommendation scheme which satisfies the following properties:

1. Easy implementation.
2. Absolutely keep differential privacy.
3. Significantly decrease the magnitude of noise in differential privacy.
4. Quantify the level of recommendation accuracy and system security.

Actually, it is clear that the probabilistic methods (including naive probabilistic methods and differential privacy methods) are efficient methods
against $k$NN attack; however, because of the global noises, the neighbour
good quality, namely the prediction accuracy, is impacted significantly. Thus, to
decrease the magnitude of differential privacy noise, we may propose the
following approach: we can simply add Laplace noise to the final rating
prediction after the normal $k$NN CF recommendation. But Sarathy et al.
has shown in \cite{23} that the above method will release users’ privacy because
Laplace mechanism does not work well in numeric data. So, to control the
neighbour quality and to decrease the magnitude of noise, it is natural to
avoid the global randomness and repeatedly adding noise. Therefore, we
present a partitioned probabilistic neighbour selection method without any
perturbations in the process of rating prediction.

**Contributions.** In this paper, to overcome the problems of low recom-
mendation accuracy, we propose a novel method, Partitioned Probabilistic
Neighbour Selection. The main contributions of this paper are:

1. We expand the classic $k$NN attack to a more general case, $\beta$-$k$NN
attack, which flexibly adjusts the size of fake user’s set to improve the attack
effectiveness. $\beta$ is essentially regarded as a security measure denoting the
degree of difficulty for an attacker to break the neighbourhood-based CF
recommender systems. We are the first to consider the case when $\beta > 1$.

2. To protect users’ data privacy against $\beta$-$k$NN attack, we propose
a novel differential privacy preserving neighbourhood-based CF method,
which ensures a required prediction accuracy while achieving a better trade-
off between the system security and recommendation accuracy against $k$NN
attack.

3. To the best of our knowledge, we are the first to propose a theoretical
analysis of the recommendation accuracy and system security on the recom-
mendation results from any randomised neighbour selection methods in the
neighbourhood-based CF recommender systems. Previous related work only gave the experimental analysis on the same issues.

**Organisation.** The rest of this paper is organised as follows: In Section 2 we summarise both the advantages and disadvantages in the existing privacy preserving methods on CF recommender systems. In Section 3 we introduce the relevant background knowledge in this paper. In Section 4 we introduce an existing attack to neighbourhood-based CF recommender systems, then expand it to a general case, $\beta$-$k$NN attack. Next, We proposed a novel differential privacy recommendation approach, Partitioned Probabilistic Neighbour Selection, in Section 5. Afterwards, the theoretical analysis of our approach on the performance of both recommendation accuracy and system security are provided in Section 6. Then, in Section 7, we show the experimental evaluation results. Finally, in Section 8 we conclude this paper.

## 2 Related Work

A noticeable number of literature has been published to preserve customers’ private data in recommender systems. However, Calandrino et al. [5] proposed a neighbourhood-based CF attack, $k$NN attack, which is a serious privacy threat to the neighbourhood-based CF recommender systems in e-commerce, e.g., Amazon. In this section, we briefly discuss some of the research literature in privacy preserving neighbourhood-based CF recommender systems.

### 2.1 Traditional Privacy Preserving Recommender Systems

Amount of traditional privacy preserving methods have been developed in CF recommender systems [29], including cryptographic [11, 21], obfuscation [22, 27], perturbation [31] and probabilistic methods [1]. Erkin et al. [11] applied homomorphic encryption and secure multi-party computation in privacy preserving recommender systems, which allows users to jointly compute their data to receive recommendation without sharing the true data with other parties. Nikolaenko et al. [21] combined a famous recommendation technique, matrix factorization, and a cryptographic method, garbled circuits, to provide recommendations without learning the real user ratings in database. The Cryptographic methods provide the highest guarantee for both prediction security ans system security by introducing encryption rather than adding noise to the original record. Unfortunately, unnecessary computational cost impacts its application in industry [29]. Obfuscation and perturbation are two similar data processing methods. In particular, obfuscation methods aggregate a number of random noises with real users rating to preserve user’s sensitive information. Parameswaran et al. [22] proposed an obfuscation framework which exchanges the sets of similar items before
submitting the user data to CF server. Weinsberg et al. [27] introduced extra reasonable ratings into user’s profile against inferring user’s sensitive information. Perturbation methods modify the user’s original ratings by a selected probability distribution before using these ratings. Particularly, Bilge et al. [4] added uniform distribution noise to the real ratings before the utilisation of user’s rating in prediction process. While, Basu et al. [3] regarded the deviation between two items as the adding noise. Both perturbation and obfuscation obtain good trade-off between prediction accuracy and system security due to the tiny data perturbation, but the magnitude of noise or the percentage of replaced ratings are not easy to be calibrated [9,29]. The probabilistic method [1] applied weighted sampling in neighbour selection which preserves users’ privacy against kNN successfully; however, it cannot provide enough accuracy due to its global randomness. Because the performance of the neighbourhood-based CF methods largely depends on the quality of neighbours. We suppose the top $k$ neighbour as the highest quality neighbour set, the randomised weighted selection process will return neighbours with lower similarity with a high probability. Then the prediction accuracy will be impacted significantly [29]. Therefore, achieving a trade-off between privacy and utility, while calibrating the adding noise are difficult tasks for these techniques.

2.2 Differential Privacy Recommender Systems

As a well-known privacy definition, the differential privacy technology [7] has been applied in the research of privacy preserving recommender systems. For example, McSherry et al. [19] provided the first differential privacy neighbourhood-based CF recommendation algorithm. In fact, their naive differential privacy protects the neighbourhood-based CF recommender systems against kNN attack successfully, as they added Laplace noise into the covariance (similarity between users/items) matrix globally, so that the output $k$ nearest neighbours set is no longer the original top $k$ neighbours. However, the global noise decreases the accuracy of their recommendation algorithms significantly.

Another differential privacy neighbourhood-based CF recommender systems algorithm is proposed by Zhu et al. [29] which inspired this study. It aims to provide better prediction accuracy than McSherry et al. [19] while aiming to keep differential privacy at both neighbour selection stage and rating prediction stage. They proposed a Private Neighbour Collaborative Filtering (PNCF) by introducing exponential differential privacy [20] to the process of neighbour selection to guarantee the system security against kNN attack. After selecting the $k$ neighbours, same with McSherry et al. [19], they also added Laplace noise into the similarity matrix to make the final prediction.

Unlike the $k$ nearest neighbour method which selects the $k$ most similar
candidates, the PNCF method \cite{29} randomly selects the $k$ neighbours with each candidate $u_i$’s weight $\omega_i$. According to exponential mechanism of differential privacy, the selection weight is measured by a score function and its corresponding sensitivity as follow,

$$\omega_i = \exp\left(\frac{\epsilon}{4k \times RS} q_a(U(u_a), u_i)\right),$$  

(1)

where $q$ is the score function, $RS$ is the Recommendation-Aware Sensitivity of score function $q$ for any user pairs $u_i$ and $u_j$, $\epsilon$ is differential privacy parameter, and $U(u_a)$ is the set of user $u_a$’s candidate list. For a user $u_a$, the score function $q$ and its Recommendation-Aware Sensitivity are defined as follows:

$$q_a(U(u_a), u_i) = \text{sim}_{ai},$$  

(2)

$$RS = \max \left\{ \max_{s \in S_{ij}} \left( \frac{r_{is} \cdot r_{js}}{\|r_i\| \cdot \|r_j\|} \right), \max_{s \in S_{ij}} \left( \frac{r_{is} \cdot r_{js} (\|r_i\| \cdot \|r_j\| - \|r'_i\| \cdot \|r'_j\|)}{\|r_i\| \cdot \|r_j\| \cdot \|r'_i\| \cdot \|r'_j\|} \right) \right\},$$  

(3)

where $r_{is}$ is user $u_i$’s rating on item $t_s$, $\text{sim}_{ai}$ is the similarity between user $u_a$ and $u_i$, $r_i$ is user $u_i$’s average rating on every item, $S_{ij}$ is the set of all items co-rated by both users $i$ and $j$, i.e., $S_{ij} = \{ s \in S | r_{is} \neq \emptyset \& r_{js} \neq \emptyset \}$.

However, the above naive differential privacy neighbour selection is nearly the same to the probabilistic neighbour selection \cite{1}. To address the above problem of low prediction accuracy in \cite{1}, a truncated parameter $\lambda$ was introduced in \cite{29}. Simply speaking, the candidates whose similarity is greater than $(\text{sim}(a,k) + \lambda)$ are selected to the neighbour set, while, whose similarity is less than $(\text{sim}(a,k) - \lambda)$ will not be selected, where $\text{sim}(a,k)$ denotes the similarity of user $u_a$’s $k$th neighbour. Theorem 3.1 in \cite{29} provided an equation to calculate the value of $\lambda$, i.e. $\lambda = \min(\text{sim}(a,k), \frac{4k \cdot RS}{\epsilon} \ln \frac{k(n-k)}{\rho})$, where $\rho$ is a constant, $0 < \rho < 1$.

However, we observe that the above idea in \cite{29} has three weaknesses. Firstly, it adds random noise in the process of neighbour selection twice; however, it is not necessary. Because we can preserve privacy against $k\text{NN}$ successfully only by introducing randomness once, the extra randomness will decrease the prediction accuracy significantly. Secondly, the value of $\lambda$ may not be achievable. This is because when computing the value of $\lambda$ by $\rho$, it results in a good theoretical recommendation accuracy, but does not yield a good experimental recommendation accuracy on the given test datasets in \cite{29}. So the PNCF method \cite{29} will actually be a method of Global Probabilistic Neighbour Selection \cite{1} and cannot guarantee any recommendation accuracy. Thirdly, the PNCF scheme breaks differential privacy in the process of neighbour selection. Suppose there is a tiny change in the dataset, then the value of similarity between target user $u_a$ and other users $u_i$ in the candidate list will change. There may exist a user $u_c$ whose probability of being selected may change from 0 to $x > 0$, then the ratio between the two probabilities will be 0 or infinite, none of which satisfy Definition 1 in Section 3.2.
3 Preliminaries

In this section, we introduce the foundational concepts and mathematical model related with this paper in collaborative filtering, differential privacy, and Wallenius’ non-central hyper-geometric distribution.

3.1 k Nearest Neighbour Collaborative Filtering

A collaborative filtering based recommender system predicts users’ potential preferences by aggregating the relevant historical data. Collaborative filtering, a popular technique in recommender systems, is in three categories: neighbourhood-based methods, association rules based methods, and matrix factorisation methods [17]. The neighbourhood-based methods generally provides recommendations by combining the opinions of a user’s k nearest neighbours [2].

Neighbour Selection and Rating Prediction are two main stages in neighbourhood-based CF [29]. At the Neighbour Selection stage, a target user $u_a$’s neighbours are selected according to their similarity value in the target user $u_a$’s similarity array $S_a$, where similarities between any two users/items are calculated by a measurement metric. Two of the most popular similarity measurement metrics are the Pearson correlation coefficient and Cosine-based Similarity [2]. In kNN method, we select the k most similar neighbours of a target user/item.

(1) Pearson Correlation Coefficient (user-based):

$$sim_{ij} = \frac{\sum_{s \in S_{ij}} (r_{is} - \bar{r}_i)(r_{js} - \bar{r}_j)}{\sqrt{\sum_{s \in S_{ij}} (r_{is} - \bar{r}_i)^2 \sum_{s \in S_{ij}} (r_{js} - \bar{r}_j)^2}}, \quad \text{(4)}$$

(2) Cosin-based Similarity (user-based):

$$sim_{ij} = \cos(r_i, r_j) = \frac{r_i \cdot r_j}{\|r_i\|\|r_j\|} = \frac{\sum_{s \in S_{ij}} r_{is}r_{js}}{\sqrt{\sum_{s \in S_{ij}} r_{is}^2}\sqrt{\sum_{s \in S_{ij}} r_{js}^2}}, \quad \text{(5)}$$

where $r_{is}$ is user $u_i$’s rating on item $t_s$, $r_{is} \in \mathcal{R}$, $\mathcal{R}$ is the user-item rating dataset, $sim_{ij}$ is the similarity between user $u_i$ and user $u_j$, $\bar{r}_i$ is user $u_i$’s average rating on every item, $S_{ij}$ is the set of all items co-rated by both users $i$ and $j$, i.e., $S_{ij} = \{s \in S | r_{is} \neq \emptyset \& r_{js} \neq \emptyset \}$.

At the stage of Rating Prediction in user-based CF methods, the predicted rating $\hat{r}_{ax}$ of user $u_a$ on item $t_x$ is calculated as an aggregation of other users’ rating on item $t_x$ [2-29]. The prediction of $\hat{r}_{ax}$ is computed as follow:

$$\hat{r}_{ax} = \frac{\sum_{u_i \in N_k(u_a)} sim(a,i)r_{ix}}{\sum_{u_i \in N_k(u_a)} |sim(a,i)|}, \quad \text{(6)}$$
where, \( N_k(u_a) \) is a sorted set which contains user \( u_a \)'s \( k \) nearest neighbours, \( N_k(u_a) \) is sorted by similarity in a descending order, \( \text{sim}(a, i) \) is the \( i \)th neighbour of \( u_a \) in \( N_k(u_a) \).

### 3.2 Differential Privacy

Informally, differential privacy \[7,8\] is a scheme that minimises the sensitivity of output for a given statistical operation on two different (differentiated in one record to protect) datasets. Specifically, differential privacy guarantees no matter whether one specific record appears in a database, the privacy mechanism will shield the specific record to the adversary. The strategy of differential privacy is adding a random noise to the result of a query function on the database.

To understand the spirit of differential privacy clearly, several items will be introduced in advance. Firstly, \( X(x_1, x_2, \cdots, x_n) \) and \( X'(x'_1, x'_2, \cdots, x'_n) \) are two databases with \( n \) entries which differ in only one entry, where \( x_i \) and \( x'_i \) are the \( i \)th entry of \( X \) and \( X' \). We call \( X \) and \( X' \) are neighbouring dataset. Secondly, \( f(X) \) is the query function on database \( X \), the respond is the combination of the real answer \( a = f(X) \) and a chosen random noise. Thirdly, the privacy mechanism \( T \), namely, the respond, is computed by \( T(X) = f(X) + \text{Noise} \). A formal definition of Differential Privacy is shown as follow:

**Definition 1 (\( \epsilon \)-Differential Privacy \[7\]).** A randomised mechanism \( T \) is \( \epsilon \)-differential privacy if for all neighbouring datasets \( X \) and \( X' \), and for all outcome sets \( S \subseteq \text{Range}(T) \), \( T \) satisfies: \( \Pr[T(X) \in S] \leq \exp(\epsilon) \cdot \Pr[T(X') \in S] \), where \( \epsilon \) is a privacy budget.

The privacy budget \( \epsilon \) is set by the database owner. Usually, a smaller \( \epsilon \) denotes a higher privacy guarantee because the privacy budget \( \epsilon \) reflects the magnitude of difference between two neighbouring datasets.

There are two main applications of the randomised mechanism \( T \): the Laplace mechanism \[7\] and the Exponential mechanism \[20\]. The definitions are shown as below:

**Definition 2 (Laplace Mechanism \[7\]).** Let a query function \( f: \mathbb{R} \rightarrow \mathbb{R}^d \), the \( \epsilon \)-differential privacy mechanism \( T \) obeys that \( T(X) = f(X) + \text{Lap}^{-1}\left(\frac{\Delta f}{\epsilon}, \Pr\right)^d \), where the sensitivity of function \( f \), \( \Delta f = \max |f(X) - f(X')| \), for all neighbouring datasets \( X, X' \in \mathcal{D}^n \), and \( d \) represents the dimension.

**Definition 3 (Exponential Mechanism \[20\]).** Given a score function of a database \( X, q(X, x) \), which reflects the score of query respond \( x \). The exponential mechanism \( T \) provides \( \epsilon \)-differential privacy, if \( T(X) = \{ \text{the probability of a query respond } x \propto \exp(\frac{-q(X, x)}{2\Delta q}) \} \), where \( \Delta q = \max |q(X, x) - q(X', x)| \), denotes the sensitivity of \( q \).
3.3 Wallenius’ Non-central Hyper-geometric Distribution

Wallenius’ non-central hyper-geometric distribution is a distribution of weighted sampling without replacement. Formally, it is defined as follow [13]: We assume there are \( c \) distinct categories in the population, each category contains \( m_i \) individuals, \( i \in [1, c] \). All the individuals from category \( i \) have the same weight \( \omega_i \), \( i \in [1, c] \). The probability of an individual is sampled at a given draw is proportional to its weight \( \omega_i \). Let \( x_v = (x_{1v}, x_{2v}, \ldots, x_{cv}) \) denote the total number of the individuals in each colour sampled after the first \( v \) draws. The probability that the next draw gives a individual of colour \( i \) is:

\[
P_i(v+1)(x_v) = \frac{(m_i - x_{iv})\omega_i}{\sum_{j=1}^{c}(m_j - x_{jv})\omega_j}.
\]  

(7)

The weighted sampling process without replacement is repeatedly until \( k \) individuals have been retained, namely, \( k = \sum_{i=1}^{c} x_i \), where \( x_i \) denotes the number of individuals sampled from category \( i \) by Wallenius’ non-central hypergeometric distribution.

Wallenius [26] proposed the probability mass function for this distribution in the univariate case (\( c = 2 \)). Chesson [6] expanded Wallenius’s solution to the multivariate case (\( c > 2 \)). In this paper, we focus on the multivariate Wallenius’ non-central hyper-geometric distribution’s probability mass function because we regard one user/item in a recommender system as one individual in Wallenius’ non-central hyper-geometric distribution. The multivariate probability mass function (PMF) is shown as blow:

\[
m_{\text{wchnhypg}} = \Lambda(x)I(x),
\]  

(8)

where \( \Lambda(x) = \prod_{i=1}^{c} \left( \frac{m_i}{x_i} \right) \), \( I(x) = \int_{0}^{1} \prod_{i=1}^{c} (1 - t^{\omega_i/d})x_i \, dt \), \( d = \omega \cdot (m - x) = \sum_{i=1}^{c} \omega_i (m_i - x_i) \), \( x = (x_1, x_2, \ldots, x_c) \), \( m = (m_1, m_2, \ldots, m_c) \), \( \omega = (\omega_1, \omega_2, \ldots, \omega_c) \).

While in this paper, we mainly use the following properties to evaluate different probabilistic relevant approaches. Manly [18] gave the approximated solution \( \mu^* = (\mu_1^*, \mu_2^*, \ldots, \mu_c^*) \) to the mean \( \mu = (\mu_1, \mu_2, \ldots, \mu_c) \) of \( x \) after the final draw:

\[
\left(1 - \frac{\mu_1^*}{m_1}\right)^{1/\omega_1} = \left(1 - \frac{\mu_2^*}{m_2}\right)^{1/\omega_2} = \ldots = \left(1 - \frac{\mu_c^*}{m_c}\right)^{1/\omega_c},
\]  

(9)

where \( \sum_{i=1}^{c} \mu_i^* = k, \forall i \in C : 0 \leq \mu_i^* \leq m_i \).

Fog [12] stated the following properties of Equation (9): firstly, the solution \( \mu^* \) is valid under the conditions that \( \forall i \in C : m_i > 0 \) and \( \omega_i > 0 \). Secondly, the mean given by Equation (9) is a good approximation in most cases. Thirdly, Equation (9) is exact when all \( \omega_i \) are equal.
4 A Generalised Privacy Attack for Recommender Systems

In this section, we firstly introduce a popular attack, \( k \) nearest neighbour attack, then we expand the concept to a general attack, \( \beta-k \) nearest neighbour attack.

4.1 \( k \) Nearest Neighbour Attack

Calandrino et al. [5] stated a user-based attack called \( k \) Nearest Neighbour (kNN) attack. Simply, the kNN attack exploits the property that the users are more similar when sharing same rating on corresponding items to reveal user’s private data.

We assume that the recommendation algorithm (kNN CF recommendation) and its parameter \( k \) are known to the attacker. Furthermore, the attacker’s auxiliary information consists of a target user \( u_a \)’s partial history rating values, i.e., he already knows the ratings of \( m \) items that \( u_a \) has rated. Usually, \( m \approx 8 \). He aims to catch \( u_a \)’s transactions that he does not yet know about.

To achieve this goal, the attacker firstly creates \( k \) fake users who have the same ratings with \( u_a \) only on the \( m \) items. With a high probability, each fake user’s \( k \) nearest neighbours set \( N_k \) (fake user) will include the other \( k-1 \) fake users and the target user \( u_a \). Because the target user \( u_a \) is the only neighbour who has ratings on the items which are not rated by the fake users, to provide recommendations on these items to the fake users, the recommender system has to give \( u_a \)’s rating to the fake users directly. Obviously, the fake users learn the target user \( u_a \)’s whole rating list successfully with kNN attack.

4.2 \( \beta-k \) Nearest Neighbours Attack

According to the existing privacy preserving neighbourhood-based CF recommendation methods, we expand the kNN attack to a more general case, named \( \beta-k \) Nearest Neighbour (\( \beta-k\)NN) attack.

As we know, to preserve the target user \( u_a \)’s private information against kNN attack, we should avoid selecting the true \( k \) nearest neighbours, so the existing methods applied the randomness techniques. However, suppose the final \( k \) neighbours are selected from the top \( \beta k \) users of \( u_a \)’s candidate list, also the parameters \( \beta \) and \( k \) are known to the attacker, the attacker would catch \( u_a \)’s private data with a high probability by creating \( \beta k \) fake users. When \( \beta \) is not great enough, it is still not difficult to break the privacy preserving neighbourhood-based CF recommender systems. Therefore, the \( \beta-k\)NN attack can flexibly adjust the size of fake user’s set to improve the attack effectiveness. Actually, kNN attack can be regarded as 1-kNN attack in the expanded case of \( \beta-k\)NN attack.
In $\beta$-kNN attack, $\beta$ can be treated because a security measure as a greater value of $\beta$ represents a higher fraud cost. We will show the relationship between the prediction utility and $\beta$ in Section 6.

5 Privacy Preservation by Partitioned Probabilistic Neighbour Selection

In this section, we firstly provide two performance metrics on the privacy preserving neighbourhood-based CF recommender systems against $\beta$-kNN attack. Then we propose our Partitioned Probabilistic Neighbour Selection algorithm based on the previous analysis.

5.1 Performance Metrics

5.1.1 Accuracy Metric

For any privacy preserving neighbourhood-based CF recommender systems, if the sum of similarity of the selected $k$ neighbours is greater, the predicted rating value will be better. The reason is simple: the neighbour is closer to the target user $u_a$ means the predicted result is more reliable, namely, we prefer the method which selects the greater similarity sum. Therefore, we define the accuracy metric $\alpha$ as the sum of $k$ selected neighbours’ similarity.

Because we propose a random neighbour selection method, the accuracy metric $\alpha$ should be regarded as the expected sum of $k$ selected neighbours’ similarity. However, it is not obvious to directly compute the expectation of the $k$ neighbours similarity sum: $\mathbb{E}(\sum_{i \in N_k(u_a)} sim(a, i))$, as we need to find all the user combinations and corresponding probabilities. So we give another way to compute this expectation,

$$\mathbb{E}(\sum_{i \in N_k(u_a)} sim(a, i)) = \sum_{i=1}^{n} (sim(a, i)\mathbb{E}(x_i)) = \sum_{i=1}^{n} sim(a, i)\mu_i, \quad (10)$$

see Section 3.3 for the definition of $x_i$ and $\mu_i$. So we compute the accuracy by the following equation in this paper:

$$\alpha = \sum_{i=1}^{n} sim(a, i)\mu_i. \quad (11)$$

5.1.2 Security Metric

According to the $\beta$-kNN attack, suppose the final $k$ neighbours are selected from the top $\beta k$ users of $u_a$’s candidate list. We assume that the parameters $\beta$ and $k$ are known to the attacker, so the attacker would catch $u_a$’s privacy with a high probability through the same process of kNN attack by creating
fake users. When β’s value is not great, it is still not difficult to break the privacy preserving recommender systems. Therefore, we define β as the security metric, the greater value of β denotes the higher fraud cost for the attacker, namely, we want to achieve a trade-off between the security metric β and a fixed prediction accuracy metric α.

5.2 Partitioned Probabilistic Neighbour Selection Algorithm

According to the motivation and previous analysis, we provide an original version of our Partitioned Probabilistic Neighbour Selection algorithm. We firstly partition the a target user’s candidate list (descending order of similarity value) by the given k, then apply a geometric distribution on the candidate list to select \( [p(1 - p)^{i-1}k] \) neighbours (apply exponential differential privacy in every partition) from partition \( i \) until we have a total of \( k \) neighbours, where integer \( i \in [1, +\infty) \), \( p \) is a geometric distribution parameter. It is clear that our original partitioned probabilistic neighbour scheme satisfies property (1) (easy implementation) in Section 1 for it does not introduce any extra computational cost. In fact, it is natural to regard the low neighbour quality as the noise in the process of neighbour selection, since the low neighbour quality has the same impact on the prediction accuracy as the noise. So our method satisfies property (3) (decreasing the magnitude of noise) in Section 1 in two ways: 1. it only adds noise in the process of neighbour selection. 2. it controls the neighbour quality by tuning the geometric distribution parameter \( p \) in the process of neighbour selection. However, the original version does not satisfy the property (2) (keeping differential privacy) and (4) (quantifying the accuracy and security), we now show the reasons and modify it to satisfy the property (2) and (4).

In the original version, we select \( [p(1 - p)^{i-1}k] \) neighbours with exponential differential privacy from partition \( i \) until we have \( k \) neighbours. Actually, it breaks differential privacy with the same reason (see details in Section 2.2) of the PNCF method [29]. Simply speaking, there may exist some users whose probability of selection will be changed from zero to a positive number because of a tiny change in rating set. To guarantee the prediction accuracy, we only modify the original scheme by changing the way we select the last neighbour (see details in next paragraph). The modified scheme keeps absolute differential privacy because no matter how we change the dataset, every candidate’s probability of selection cannot be zero. To quantify the level of recommendation accuracy and system security, we use the performance metrics \( \alpha \) and \( \beta \). We compute the parameter \( p \) and the security metric \( \beta \) by a given \( \alpha \) by Equation (20).

Algorithm 1 shows the Partitioned Probabilistic Neighbour Selection (PPNS) algorithm. In lines 1 to 5, we compute the necessary parameters by Equation (5), (3), (2), (1) and (20). In lines 6 to 18, we select the \( k \) neighbours by Partitioned Probabilistic Neighbour Selection, then re-
turn the target user’s $k$ neighbours and the security metric value $\beta$. We firstly mark all of the partitions as unvisited. Next, we select $\lceil p(1 - p)^{i-1}k \rceil$ neighbours with exponential differential privacy from partition $i$ (mark this partition as visited) until we have a total of $k - 1$ neighbours. Finally, we select the last neighbour from all the unvisited partitions.

Algorithm 1: Partitioned Probabilistic Neighbour Selection.

Input:
- Original user-item rating set, $\mathcal{R}$;
- Target user, $u_a$ and prediction item, $t_x$;
- Number of neighbours, $k$;
- Differential privacy parameter, $\epsilon$;
- Accuracy metric, $\alpha$.

Output:
- Target user $u_a$’s $k$-neighbour set, $N_k(u_a)$;
- Security metric, $\beta$.

1: Compute the similarity list for target user $u_a$, $S_a$;
2: Sort $S_a$ in descending order, $S_a$;
3: Compute exponential differential privacy sensitivity, $RS$;
4: Compute user $u_i$’s selection weight, $\omega_i$;
5: Compute the geometric distribution parameter, $p$;
6: Partition the sorted $S_a$ by $k$;
7: for $i = 1$ to $n$ do
8:   if Neighbour Number $\neq k - 1$ then
9:     Select $\lceil p(1 - p)^{i-1}k \rceil$ neighbours from partition $i$ to $N_k(u_a)$;
10:    Mark partition $i$ as visited;
11:   else
12:     Neighbour Number += $\lceil p(1 - p)^{i-1}k \rceil$;
13:   end if
14: end for
15: Select one neighbour from unvisited partitions;
16: $\beta =$ last neighbour’s partition index number;
17: return $N_k(u_a)$, $\beta$;

6 Theoretical Analysis

In this section, we use multivariate Wallenius’ non-central hyper-geometric distribution to analyse any randomised neighbour selection methods on both performance of accuracy and security against kNN attack theoretically. The reason is both multivariate Wallenius’ non-central hyper-geometric distribution and randomised neighbour selection methods are weighted sampling
without replacement, the samples are selected one by one from universe, and the sampling weight is only depends on each sample’s attribute, i.e., the ball’s colour or user’s similarity.

6.1 Accuracy Analysis

In this part, to analyse the accuracy performance, we will firstly modify the Equation (9) to match with a general randomised neighbour selection method. As the selection weight in a general probabilistic neighbour selection method only relies on the user’s similarity, we regard user \( u_i \)’s similarity \( \text{sim}(a, i) \) as the sample’s colour in multivariate Wallenius’ non-central hyper-geometric distribution. Thus in randomised neighbour selection methods, \( m_i = 1 \), \( c = n \), \( N = \sum_{i=1}^{c} m_i = \sum_{i=1}^{n} m_i = n \). Therefore, we rewrite the Equation (9) as:

\[
A = (1 - \mu_1)^{1/\omega_1} = (1 - \mu_2)^{1/\omega_2} = \ldots = (1 - \mu_n)^{1/\omega_n},
\]

where \( A \) is a constant.

Now we start evaluating the Partitioned Probabilistic Neighbour Selection by Equation (12). To make it easy, we also partition the candidate list in PNCF method \[29\] and Probabilistic Neighbour Selection \[1\] by the given \( k \).

**Lemma 1.** \( C \) is an \( n \) sized set. We independently sample several samples with multivariate Wallenius’ non-central hyper-geometric distribution from \( C \) twice, suppose \( \mu_i \) and \( \hat{\mu}_i \) are the expected number of sample \( i \) from the two samplings. Then \( \forall i \in [1, n], \mu_i > \hat{\mu}_i \Leftrightarrow \sum_{i=1}^{n} \mu_i > \sum_{i=1}^{n} \hat{\mu}_i \).

**Proof.** Let \( \sum_{i=1}^{n} \mu_i = \hat{X}, \sum_{i=1}^{n} \hat{\mu}_i = \bar{X}, A = (1 - \mu_1)^{1/\omega_1}, \hat{A} = (1 - \hat{\mu}_1)^{1/\omega_1} \).

(1) Proof of sufficient condition, \( \mu_i > \hat{\mu}_i \Rightarrow \sum_{i=1}^{n} \mu_i > \sum_{i=1}^{n} \hat{\mu}_i \):

\[ A = (1 - \mu_1)^{1/\omega_1} \Rightarrow \hat{\mu}_1 = 1 - \hat{A}^{1/\omega_1} \]
\[ \Rightarrow \sum_{i=1}^{n} \mu_i = k - \sum_{i=1}^{n} \hat{A}^{1/\omega_i} = \hat{X}. \]

Similarly, \( \hat{X} = \sum_{i=1}^{n} \hat{\mu}_i > \sum_{i=1}^{n} \hat{\mu}_i = \bar{X}, \) and \( \mu_i \) and \( \hat{\mu}_i \) share the same \( \omega_i \),

\[ \sum_{i=1}^{n} \hat{A}^{1/\omega_i} < \sum_{i=1}^{n} \hat{A}^{1/\omega_i} \]
\[ \Rightarrow \hat{A} < \bar{A} \]
\[ \Rightarrow (1 - \mu_1)^{1/\omega_1} < (1 - \hat{\mu}_1)^{1/\omega_1} \]
\[ \Rightarrow \mu_i > \hat{\mu}_i. \]

Therefore, we have \( \forall i \in [1, n], \mu_i > \hat{\mu}_i \Leftrightarrow \sum_{i=1}^{n} \mu_i > \sum_{i=1}^{n} \hat{\mu}_i. \) \( \square \)
Lemma 1 shows the fact that when selecting neighbours with multivariate Wallenius’ non-central hyper-geometric distribution by several randomised neighbour selection methods from a same sized partition, if one method selects more neighbours, then the expected number of each neighbour in that method is greater too, and vice versa.

**Lemma 2.** The method, which selects more users from the first partition (contains user \( u_1 \) to \( u_k \)) of a descending order similarity list, yields a better rating prediction, i.e., \( \sum_{i=1}^{k} \mu_i > \sum_{i=1}^{k} \hat{\mu}_i \Rightarrow \alpha \geq \hat{\alpha} \), where \( \alpha \) denotes the accuracy metric value.

**Proof.** Let \( X_j = \sum_{i \in \text{group}_j} \mu_i \), \( \hat{X}_j = \sum_{i \in \text{group}_j} \hat{\mu}_i \), e.g., \( X_1 = \sum_{i \in \text{group}_1} \mu_i = \sum_{i=1}^{k} \mu_i \). Assume an extreme case:

\[
X_1 > \hat{X}_1 \\
X_2 < \hat{X}_2 \\
X_3 < \hat{X}_3 \\
\vdots < \vdots
\]

\[
\therefore k = \sum_j X_j = \sum_j \hat{X}_j,
\]

\[
\therefore X_1 - \hat{X}_1 = (\hat{X}_2 - X_2) + (\hat{X}_3 - X_3) + \cdots.
\]

It is obvious that, in both sides of the above equation, every item \( > 0 \).

According to Lemma 1: \( X > \hat{X} \Leftrightarrow \mu_i > \hat{\mu}_i \); we have,

\[
\sum_{\text{group}_1} (\mu_i - \hat{\mu}_i) = \sum_{\text{group}_2} (\hat{\mu}_i - \mu_i) + \sum_{\text{group}_3} (\hat{\mu}_i - \mu_i) + \cdots
\]

and every \((\cdot) > 0\).

\[
\therefore 1 \geq \sum_{\text{group}_1} \text{sim}(a, i) \geq \sum_{\text{group}_2} \text{sim}(a, j) \geq 0, (i < j),
\]

\[
\therefore \sum_{\text{group}_1} \text{sim}(a, i)(\mu_i - \hat{\mu}_i) \geq \sum_{\text{group}_2} \text{sim}(a, i)(\hat{\mu}_i - \mu_i) + \sum_{\text{group}_3} \text{sim}(a, i)(\hat{\mu}_i - \mu_i) + \cdots
\]

\[
\therefore \sum_{i=1}^{n} \text{sim}(a, i)\mu_i \geq \sum_{i=1}^{n} \text{sim}(a, i)\hat{\mu}_i.
\]

According to Equation (11), we have \( \alpha \geq \hat{\alpha} \).

Therefore, the method, which selects more users from the first group, is more reliable on the predicted rating value.

**Theorem 1.** If \( p > 1 - \left( \frac{n-k}{n} \right)^{\omega_1} \), the recommendation accuracy performance of Partitioned Probabilistic Neighbour Selection is better than PNCF method [29] and Probabilistic Neighbour Selection [1].

**Proof.** We firstly demonstrate the best case for the PNCF method [29] and Probabilistic Neighbour Selection [1]: \( \text{sim}(a, 1) = \cdots = \text{sim}(a, k) = 1 > 0 = \text{sim}(a, k+1) = \cdots = \text{sim}(a, n) \).

\[
\therefore k = k\mu_1 + (n-k)\mu_n.
\]

According to Equation (12), \( A = (1 - \mu_1)^{1/\omega_1} = (1 - \mu_n)^{1/\omega_n} \), \( \mu_n = 1 - (1 - \mu_1)^{1/\omega_1} \). Let \( \Delta = \mu_1 - \mu_n = (1 - \mu_1)^{1/\omega_1} - (1 - \mu_1) \), then \( \mu_1 = \frac{k}{n} + \frac{(n-k)\Delta}{k} < \frac{k}{n} + \Delta \), namely, \( \mu_1 < \frac{k}{n} + (1 - \mu_1)^{1/\omega_1} - (1 - \mu_1) \), then \( \mu_1 < 1 - \left( \frac{n-k}{n} \right)^{\omega_1} \).
In PNCF method [29] and Probabilistic Neighbour Selection [1], \( \sum_{i \in \text{group}_1} \mu_i \leq k \mu_1 \), while in Partitioned Probabilistic Neighbour Selection, \( \sum_{i \in \text{group}_1} \mu_i = pk \). Therefore, according to Lemma [2], when \( p > 1 - \left( \frac{n-k}{n} \right)^{\omega_1} \), \( \alpha \geq \hat{\alpha} \), namely, the recommendation accuracy of Partitioned Probabilistic Neighbour Selection is better than PNCF method [29] and Probabilistic Neighbour Selection [1].

Since we have qualitatively analysed the recommendation accuracy performance between Partitioned Probabilistic Neighbour Selection and PNCF method [29] and Probabilistic Neighbour Selection [1], now we provide the quantitative analysis of our Partitioned Probabilistic Neighbour Selection. Let \( \alpha_0 \) be the initial accuracy metric. 

\[
\sum_{i=1}^{n} \frac{\text{sim}(a,i) \mu_i}{\sum_{i=1}^{n} \text{sim}(a,i)} - \frac{\sum_{i=1}^{k} \mu_i}{\sum_{i=1}^{n} 1} = \sum_{i=1}^{k-1} \frac{\sum_{j=i+1}^{k} (\text{sim}(a,i) - \text{sim}(a,j)) (\mu_i \mu_j)}{k \sum_{i=1}^{n} \text{sim}(a,i)} \geq 0,
\]

then we have

\[
\sum_{i=1}^{k} \frac{\mu_i}{\sum_{i=1}^{n} \text{sim}(a,i)} \leq \sum_{i=1}^{n} \frac{\text{sim}(a,i) \mu_i}{\sum_{i=1}^{n} \text{sim}(a,i)}.
\]

Namely,

\[
\frac{p}{k} = \frac{\sum_{i=1}^{k} \mu_i}{\sum_{i=1}^{n} 1} \leq \frac{\sum_{i=1}^{k} \text{sim}(a,i) \mu_i}{\sum_{i=1}^{n} \text{sim}(a,i)} \leq \frac{\sum_{i=1}^{k} \text{sim}(a,i) \mu_i + \sum_{i=k+1}^{n} \text{sim}(a,i) \mu_i + \cdots}{\sum_{i=1}^{n} \text{sim}(a,i)} = \frac{\alpha_0}{\sum_{i=1}^{n} \text{sim}(a,i)}.
\]

Thus, \( p \leq \frac{\alpha_0}{\sum_{i=1}^{n} \text{sim}(a,i)} \). Namely, when \( p \geq \frac{\alpha_0}{\sum_{i=1}^{n} \text{sim}(a,i)} \) the actual accuracy \( \alpha \) must be greater than \( \alpha_0 \). Therefore, we give the range of \( p \)'s value to guarantee the accuracy metric \( \alpha \geq \alpha_0 \), \( p \in \left[ \frac{\alpha_0}{\sum_{i=1}^{n} \text{sim}(a,i)}, 1 \right] \).

### 6.2 Security Analysis

In this section, we firstly provide the range of \( p \), so that our approach guarantees the system security against kNN attack. Next, we present the quantitative analysis by providing a relationship between the the probabilistic parameter \( p \) and the security metric \( \beta \).

In PNCF method [29], according to Equation (8), the probability mass function is:

\[
PMF = I(x) = \int_{0}^{1} \prod_{i=1}^{n} (1 - t^{\omega_i/d}) x_i dt,
\]

\[
d = \omega \cdot (m - x) = \sum_{i=1}^{n} \omega_i (1 - x_i),
\]

where, \( x = (x_1, x_2, \ldots, x_n) \), \( \omega = (\omega_1, \omega_2, \ldots, \omega_n) \).

For the case of selecting the top-\( k \) users, we have:

\[
x_i = \begin{cases} 1 & \text{if } i \in [1,k] \\ 0 & \text{if } i \in [k+1,n] \end{cases}.
\]
Thus, the probability of selecting top-$k$ users in PNCF method \cite{29} and Probabilistic Neighbour Selection \cite{1} is:

$$Pr = I(x) = \int_0^1 \prod_{i=1}^k (1 - t^{\omega_i/d}) dt > 0,$$

$$d = \omega \cdot (m - x) = \sum_{i=k+1}^n \omega_i.$$ \hfill (16)

In Partitioned Probabilistic Neighbour Selection, because we actually select $\lceil pk \rceil$ users from the top-$k$ users, when $p \leq \frac{k-1}{k}$, the probability of selecting top-$k$ users as the final $k$ neighbours is 0, namely, we provide the absolute security against the $k$NN attack when setting $p \leq \frac{k-1}{k}$.

To compute the value of $\beta$ according to our selection process, we select $p(1-p)^{i-1}k$ users from group $i$, so the first time we select one user from a group, the number $j$ of this group obeys the following inequation:

$$p(1-p)^{j-1}k < \frac{3}{2} \Rightarrow j > 1 + \frac{(\ln 3 - \ln 2) - \ln pk}{\ln(1-p)}.$$ \hfill (18)

Before the group $j + 1$, we have selected $pk + p(1-p)k + \cdots + p(1-p)^{j-1}k$ users, there are $(1-p)^{j-1}k$ users can be selected. Since each of the $(1-p)^{j-1}k$ comes from one group, the total number of the groups where the $k$ neighbours come from is:

$$\beta = (j - 1) + \frac{(1-p)^{j-1}k}{1} = (j - 1) + (1-p)^{j-1}k.$$ \hfill (19)

### 6.3 Analysis Results

According to the previous analysis, when setting the probabilistic parameter $p$ as $1 - \left(\frac{n+k}{n}\right)^{\omega_i} < p \leq \frac{k-1}{k}$, our Partitioned Probabilistic Neighbour Selection achieve better performance of recommendation accuracy than Private Neighbour Selection \cite{29} and Probabilistic Neighbour Selection \cite{1}. Then we give the the relationship between the accuracy metric $\alpha$ and security metric $\beta$ of our Partitioned Probabilistic Neighbour Selection by the following equation:

$$\begin{cases}
 p \in \left[\frac{\alpha_0}{\sum_{i=1}^n \text{sim}(a,i)}, 1\right] \\
 j = \left[1 + \frac{(\ln 3 - \ln 2) - \ln pk}{\ln(1-p)}\right] \\
 \beta = (j - 1) + (1-p)^{j-1}k
\end{cases}$$ \hfill (20)

We guarantee to achieve $\alpha_0$ accuracy against $\beta$-$k$NN attack.
6.4 A representative Example

In this section, we show a simple but representative example of the range of the probabilistic parameter \( p \). Suppose \( k = \theta n, \theta \in (0, 1) \), we know the lower bound of \( p \),

\[
1 - \left( \frac{n-k}{n} \right)^{\omega_1} = 1 - (1 - \theta)^{\omega_1},
\]

is a monotone-increasing function of \( \theta \). Because the value of \( \theta \) is always small \((k \in [30, 50] \text{ and } n \text{ is always greater than } 1000)\), the value of the lower bound of \( p \) will be very small. In the mean time, consider the upper bound of \( p \), it would be a number close to 1. Therefore, the range of value \( p \) is very large in the set of \((0, 1)\).

Now we will show an example in a real scenario. Let \( k = 50, n = 500, \epsilon = 1, RS = 1 \), so the lower bound of \( p \) would be

\[
1 - \left( \frac{n-k}{n} \right)^{\exp\left(\frac{4\epsilon k^2 RS}{4k^2 RS}\right)} = 1 - \left( \frac{500-50}{500} \right)^{\exp\left(\frac{1400}{400}\right)} \approx 0.1,
\]

and the upper bound of \( p \) would be \( \frac{k-1}{k} = \frac{50-1}{50} = 0.98 \). Thus, in the above real scenario, when we set \( p \) in the range of \((0.1, 0.98) \subset [0, 1)\), the Partitioned Probabilistic Neighbour Selection would yield better performance of recommendation accuracy against \( k \)NN attack.

7 Performance Evaluation

In Section 6, we theoretically analyse the performance on both recommendation accuracy and system security, and prove that to successfully preserve customer’s privacy against \( k \)NN attack, our method ensures a better performance of recommendation accuracy than the PNCF method \[29\] and Probabilistic Neighbour Selection \[1\]. In this section, we compare the recommendation accuracy between Partitioned Probabilistic Neighbour Selection and global Neighbour Selection \[29\] and Probabilistic Neighbour Selection \[1\] by the experiments on real world dataset.

The dataset in the experiments is the MovieLens dataset\[1\]. The MovieLens dataset consists of 100,000 ratings (1-5 integral stars) from 943 users on 1682 films, where each user has voted more than 20 films, and each film received 20–250 users’ rating. Specifically, we randomly select one rating of a random user, and then predict this user’s potential value by \( k \) Nearest Neighbour (\( k \)NN), Partitioned Probabilistic Neighbour Selection (PPNS), Probabilistic Neighbour Selection (nPNS) \[1\], Private Neighbour Selection Collaborative Filtering (PNCF) \[29\].

In this paper, we use a famous measurement metric, Mean Absolute Error (MAE) \[28,29\], to measure the recommendation accuracy:

\[
MAE = \frac{1}{T} \sum_{i \in T} |r_{ai} - \hat{r}_{ai}|,
\]

\[1\]http://grouplens.org/datasets/movielens/
where \( r_{ai} \) is the real rating of user \( u_a \) on item \( t_i \), and \( \hat{r}_{ai} \) is the predicting rating. \( T \) is the test times. To guarantee a reasonable experimental result, in our experiments, \( r_{ai} \neq 0 \). Clearly, a lower MAE value denotes a better prediction accuracy. Note that in each experiment, we consider the \( k \)NN CF recommendation method as a baseline (the best method on accuracy performance).

In our experiments, we compute the parameter \( RS \) by the previous theory [29]. We set \( T = 10,000 \), namely, we do the experiments 10,000 times to compute the MAE. Specifically, we randomly select one target user and item at each time. Our experiments are run on user-based CF (because both \( k \)NN attack and \( \beta \)-\( k \)NN attack are user-based attack), and we use the cosine-based metric to compute the similarity between users. Table 1 and Figure 1 show the relationship between accuracy performance of Partitioned Probabilistic Neighbour Selection and parameter \( p \), where we set \( \epsilon = 1, k = 50, \rho = 0.5 \). Table 2 and Figure 2 show the relationship between security performance of Partitioned Probabilistic Neighbour Selection (value of \( \beta \)) and parameter \( k \), where the total partition number is 19. Table 3 and Figure 3 show the relationship between accuracy performance of all the four methods and parameter \( k \), where we set \( \epsilon = 1, p = 0.5, \rho = 0.5 \). Table 4 and Figure 4 show the relationship between accuracy performance of PNCF [29] and parameter \( \rho \), where we set \( \epsilon = 1, p = 0.5, k = 50 \).

| \( p \) | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|--------|-----|-----|-----|-----|-----|
| \( k \)NN | 0.6956 | 0.7027 | 0.6908 | 0.6835 | 0.7074 |
| PPNS   | 0.8333 | 0.7813 | 0.7289 | 0.7134 | 0.7085 |
| nPNS   | 0.8762 | 0.8918 | 0.8884 | 0.8797 | 0.8878 |
| PNCF   | 0.8798 | 0.8928 | 0.8885 | 0.8738 | 0.8753 |

| \( p \) | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|--------|-----|-----|-----|-----|-----|
| \( k \)NN | 0.6849 | 0.6899 | 0.6847 | 0.6746 | 0.6897 |
| PPNS   | 0.6872 | 0.6914 | 0.6854 | 0.6792 | 0.6897 |
| nPNS   | 0.8863 | 0.8889 | 0.8873 | 0.8781 | 0.8893 |
| PNCF   | 0.8790 | 0.8845 | 0.8869 | 0.8783 | 0.8940 |

Figure 1: Impacts of \( p \) on accuracy \( (\epsilon = 1, k = 50, \rho = 0.5) \)
Figure 2: Impacts of $p$ on security (total partition number = 19)

Table 2: Impacts of $p$ on security (total partition number = 19)

| $p$  | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|------|-----|-----|-----|-----|-----|
| $k$NN | 1   | 1   | 1   | 1   | 1   |
| PPNS  | 15  | 13  | 10  | 8   | 7   |
| nPNS  | 17  | 17  | 17  | 17  | 17  |
| PNCF  | 17  | 17  | 17  | 17  | 17  |

Figure 3: Impacts of $k$ on accuracy ($\epsilon = 1$, $p = 0.5$, $\rho = 0.5$)

Table 3: Impacts of $k$ on accuracy ($\epsilon = 1$, $p = 0.5$, $\rho = 0.5$)

| $k$  | 10  | 20  | 30  | 40  | 50  |
|------|-----|-----|-----|-----|-----|
| $k$NN | 0.8065 | 0.7149 | 0.7288 | 0.6942 | 0.6957 |
| PPNS  | 0.8962 | 0.8017 | 0.7716 | 0.7395 | 0.7430 |
| nPNS  | 0.9687 | 0.9131 | 0.9034 | 0.8867 | 0.8904 |
| PNCF  | 0.9856 | 0.9245 | 0.9094 | 0.8862 | 0.8941 |

Figure 4: Impacts of $\rho$ on accuracy ($\epsilon = 1$, $p = 0.5$, $k = 50$)

Table 4: Impacts of $\rho$ on accuracy ($\epsilon = 1$, $p = 0.5$, $k = 50$)

| $\rho$  | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---------|-----|-----|-----|-----|-----|
| $k$NN  | 0.6815 | 0.6962 | 0.6914 | 0.6740 | 0.6741 |
| PPNS  | 0.7310 | 0.7326 | 0.7175 | 0.7160 | 0.7374 |
| nPNS  | 0.8915 | 0.8817 | 0.8873 | 0.8770 | 0.8801 |
| PNCF  | 0.8911 | 0.8841 | 0.8932 | 0.8778 | 0.8862 |
According to the experiments results, we are clear that:

(1) From Fig. 1, when setting $p > 1 - \left(\frac{n-k}{n}\right)^{\omega_1}$, the accuracy performance of Partitioned Probabilistic Neighbour Selection is always better than the PNCF method [29] and Probabilistic Neighbour Selection [1]. When the value of $p$ is close to 1, the performance of Partitioned Probabilistic Neighbour Selection is close to $k$NN method. Particularly, when $p = 1$, the Partitioned Probabilistic Neighbour Selection method is the same as $k$NN method.

(2) From Fig. 1 and Fig. 2, the accuracy performance of the neighbourhood-based CF methods largely depends on the quality of neighbours. Our Partitioned Probabilistic Neighbour Selection method yields a better trade-off between the recommendation accuracy and security, as when we offer a better accuracy performance, we do not lose the security much.

(3) From Fig. 3, the size of neighbour set impacts the accuracy performance of all of the neighbourhood-based CF recommendation approaches. A large value of neighbour set size $k$ yields a better accuracy performance.

(4) From Fig. 4, the value of $\lambda$ in [29] is not achievable because the value of $\rho$ does not impact the accuracy performance of PNCF [29].

8 Conclusion

Recommender systems play an important role in e-commerce. To protect users’ private information during the process of filtering, the existing privacy preserving neighbourhood-bases CF methods fail to protect users’ privacy in rating prediction. The global probabilistic neighbour selection methods, such as the PNCF method [29] and Probabilistic Neighbour Selection [1] though can protect users’ privacy against $k$NN attack successfully, but provide no data utility guarantee. To overcome the weaknesses of the current methods, we propose a novel privacy preserving neighbourhood-based CF method, Partitioned Probabilistic Neighbour Selection, to ensure a required recommendation accuracy while maintaining high system security against $\beta$-$k$NN attack (generalisation of $k$NN attack). Theoretical and experimental analysis show that to provide an accuracy-assured recommendation against the most popular attack, $k$NN attack, our Partitioned Probabilistic Neighbour Selection method yields a better trade-off between the recommendation accuracy and system security than the PNCF methods [29] and Probabilistic Neighbour Selection [1].
References

[1] Adamopoulos, P., Tuzhilin, A.: On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 153–160. RecSys ’14, ACM, New York, NY, USA (2014)

[2] Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17(6), 734–749 (Jun 2005)

[3] Basu, A., Vaidya, J., Kikuchi, H.: Perturbation based privacy preserving slope one predictors for collaborative filtering. In: Trust Management VI, pp. 17–35. Springer (2012)

[4] Bilge, A., Polat, H.: An improved privacy-preserving dwt-based collaborative filtering scheme. Expert Systems with Applications 39(3), 3841–3854 (2012)

[5] Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: “you might also like: ” privacy risks of collaborative filtering. In: Proceedings of the 2011 IEEE Symposium on Security and Privacy. pp. 231–246. SP ’11, IEEE Computer Society, Washington, DC, USA (2011)

[6] Chesson, J.: A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation. Journal of Applied Probability 13(4), pp. 795–797 (1976)

[7] Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 4052, pp. 1–12. Springer Berlin Heidelberg (2006)

[8] Dwork, C.: Differential privacy: A survey of results. In: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation. pp. 1–19. TAMC’08, Springer-Verlag, Berlin, Heidelberg (2008)

[9] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proceedings of the Third Conference on Theory of Cryptography. pp. 265–284. TCC’06, Springer-Verlag, Berlin, Heidelberg (2006)

[10] Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. Found. Trends Hum.-Comput. Interact. 4(2), 81–173 (Feb 2011)
[11] Erkin, Z., Beye, M., Veugen, T., Lagendijk, R.L.: Privacy enhanced recommender system (2010)

[12] Fog, A.: Calculation methods for wallenius’ noncentral hypergeometric distribution. Communications in StatisticsSimulation and Computation 37(2), 258–273 (2008)

[13] Fog, A.: Sampling methods for wallenius’ and fisher’s noncentral hypergeometric distributions. Communications in StatisticsSimulation and Computation 37(2), 241–257 (2008)

[14] Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information retrieval 5(4), 287–310 (2002)

[15] Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 659–667. ACM (2013)

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

[17] Liu, B., Mobasher, B., Nasraoui, O.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Data-Centric Systems and Applications, Springer (2011)

[18] Manly, B.F.J.: A model for certain types of selection experiments. Biometrics 30(2), 281–294 (June 1974)

[19] McSherry, F., Mironov, I.: Differentially private recommender systems: Building privacy into the net. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 627–636. KDD ’09, ACM, New York, NY, USA (2009)

[20] McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science. pp. 94–103. FOCS ’07, IEEE Computer Society, Washington, DC, USA (2007)

[21] Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joyce, M., Taft, N., Boneh, D.: Privacy-preserving matrix factorization. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. pp. 801–812. CCS ’13, ACM, New York, NY, USA (2013)

[22] Parameswaran, R., Blough, D.M.: Privacy preserving collaborative filtering using data obfuscation. In: Granular Computing, 2007. GRC 2007. IEEE International Conference on. pp. 380–380 (Nov 2007)
[23] Sarathy, R., Muralidhar, K.: Evaluating laplace noise addition to satisfy differential privacy for numeric data. Trans. Data Privacy 4(1), 1–17 (Apr 2011)

[24] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web. pp. 285–295. WWW ’01, ACM, New York, NY, USA (2001)

[25] Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM conference on Electronic commerce. pp. 158–166. ACM (1999)

[26] Wallenius, K.T.: Biased sampling: The non-central hypergeometric probability distribution. Tech. Rep. 70, Stanford University (1963)

[27] Weinsberg, U., Bhagat, S., Ioannidis, S., Taft, N.: Blurme: inferring and obfuscating user gender based on ratings. In: Proceedings of the sixth ACM conference on Recommender systems. pp. 195–202. ACM (2012)

[28] Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate research 30(1), 79 (2005)

[29] Zhu, T., Ren, Y., Zhou, W., Rong, J., Xiong, P.: An effective privacy preserving algorithm for neighborhood-based collaborative filtering. Future Generation Computer Systems 36, 142–155 (2014)