Integrating Blockchain With Artificial Intelligence for Privacy-Preserving Recommender Systems

Rabeya Bosri, *Member, IEEE*, Mohammad Shahriar Rahman, Md Zakirul Alam Bhuiyan, *Senior Member, IEEE*, and Abdullah Al Omar

***Abstract*—Data privacy is one of the intriguing problems in e-commerce site. For personal or business purposes, users have to disclose their private data to these e-commerce sites. Often such businesses use these highly sensitive data for computing artificial intelligence-driven analyses like recommendation generation without user consent. In the case of recommendation generation, data need to be analyzed at the business platforms. An automated personalization, based on artificial intelligence, on a list of products with respect to user interest is generated by a recommender system. However, the secure utilization of user data is absent in such systems. This paper proposes *Private-Rec*, a privacy-preserving platform for a recommendation system through the integration of artificial intelligence and blockchain. In *Private-Rec*, blockchain gives the user a secure environment through the distributed attribute in which data can be used with the required permission. Under this platform, users receive incentives (i.e., point, discount) from the recommended company for sharing their data to be used for computing recommendations. The *Private-Rec* platform has been studied empirically.**

***Index Terms*—AI-based data analysis, distributed ledger technology, e-commerce, user-centric system.**

I. Introduction

RECOMMENDER system [1], an Artificial Intelligence (AI)-based subclass of information filtering system, makes a prediction on a list of product. These systems are the common applications of big data.

E-commerce companies (i.e. Amazon.com, Facebook, YouTube, Alibaba Group, eBay, Jingdong) are widely using recommender engines to generate an optimal recommendation based on customers' interests. Increasing product sales is one of the goals of a recommender system. Nowadays, recommender systems are not only being used by e-commerce companies, such systems are also being used by other companies like: Netflix [2], LinkedIn, Facebook [3], YouTube [4], Amazon [5]. Recommendations are generated in two ways: content-based filtering and collaborative filtering. In collaborative filtering, a prediction list is generated by determining the interrelation between users' history and other users' interest [6]. On the other hand, description of items and user profiles are explored in Content-based filtering. Here, user profiles are constructed from the user's history and the user's rating [7]. Companies collect and store a huge amount of customer data. Such data are used to generate the recommendations. Fig. 1 shows a general view of a recommender system.

A. Motivation

Nowadays, people are concerned about the privacy of their personal information which is being stored in various platforms (i.e. company, IoT device, healthcare services) for different purposes. To make an optimal recommendation using collaborative filtering [7], [8] companies store personal data of its customers. So, here comes the issue of data privacy in such AI-based platforms. There are several incidents of users private data disclosure [9]. These days, through social media, millions of users share their personal information. Recently, nearly 87 million user data were hijacked/leaked from Facebook [10]. One of the main reasons for this data leakage is weak privacy settings. As such, while the companies are collecting and storing user information users have no control over their data.

Several platforms [11]–[19] have been proposed to deal with privacy issues. However, there are still chances of exposing user data, since these sites may exploit or exchange user data with an unauthorized party without the user's consent. In this paper, we have designed a framework, named *Private-Rec*, which is a user-centric recommender system leveraging collaborative filtering. The whole process of data collection and storage is done by our platform without having to share the data with companies. No entities are able to access user data in our proposed *Private-Rec*, and the recommendation computation is performed in a secure manner. Thus, the companies will not have any chance to get access to user data. Whenever our platform uses user data, the transaction record of data sharing is stored on a blockchain. Users of our platform will receive incentives from the platform when their data are

Manuscript received March 31, 2020; revised September 7, 2020; accepted September 13, 2020. Date of publication October 14, 2020; date of current version July 7, 2021. A preliminary version of this paper appears in The 5th International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications (DependSys 2019). This is the full version. This work was supported by the Fordham University Faculty Research and Education Grant. Recommended for acceptance by Dr. Zhihong Tian. *(Rabeya Bosri and Abdullah Al Omar contributed equally to this manuscript.) (Corresponding authors: Abdullah Al Omar and Md Zakirul Alam Bhuiyan.)*

Rabeya Bosri is with the Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1205, Bangladesh (e-mail: rabeyabosri.cse@gmail.com).

Mohammad Shahriar Rahman is with the Department of Computer Science and Engineering, University of Liberal Arts, Dhaka 1205, Bangladesh (e-mail: shahriar.rahman@ulab.edu.bd).

Md Zakirul Alam Bhuiyan is with the Department of Computer and Information Sciences, Fordham University, Bronx 10458, NY, USA (e-mail: zakirulalam@gmail.com).

Abdullah Al Omar is with the Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1209, Bangladesh (e-mail: omar.cs.bd@gmail.com).

Digital Object Identifier 10.1109/TNSE.2020.3031179

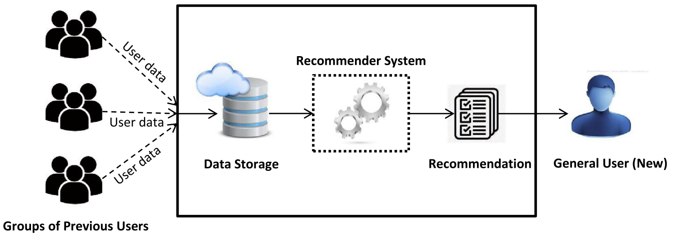

Fig. 1. Users and Recommender System's general functions.

being used. Blockchain ensures that no user data is used for computing recommendation without incentives. Blockchain [20] is a secure, ordered, and immutable data structure that stores transactions, and it pledges identification, authentication, and authorization. The benefit of using blockchain is that it tracks and handles transactions without a third party. We use duly authorized blockchain (permissioned) in our platform which enables access control. Permissioned blockchains allow a node to participate only when its identity and role are defined. This nature of permissioned blockchain prevents creating fake participants in a distributed network that thwarts different security threats (i.e., Sybil attack). The current environment of a public blockchain is suffering from scalability problem. Therefore, we are using permissioned blockchain which will give a better performance than public blockchain. We use collaborative filtering to find an optimal recommendation with respect to user interest. Our platform ensures data protection through blockchain as users have access to blockchain to track the transaction. Furthermore, user data is stored on our database, and companies are no longer able to secretly gather data from the user. Our platform provides protection of user data which property is missing from existing recommender systems.

Fig. 2 shows the general working module of our platform. If any user claims that his data was shared then through blockchain it will be very easy to find out when his data was shared with which company. Our platform resolves all the aforementioned problems in the e-commerce environment. Our platform ensures that a user has control over his data unlike the current systems.

B. Our Contribution :

In this article, our contributions are as follows.

- We are proposing *Private-Rec*, an AI-based privacy-preserving recommender system ensuring user data privacy.
- We utilize blockchain to store data transactions to make the company accountable.
- *Private-Rec* guarantees accountability, integrity, pseudonymity, and privacy. Data privacy problems have been addressed by storing all data in the accountable data cloud, and our platform uses cryptographic features to ensure privacy.
- A cluster-based incentive mechanism has been introduced. The users who will share data to generate recommendations will receive some point as an incentive. These points can be used in the platform later.

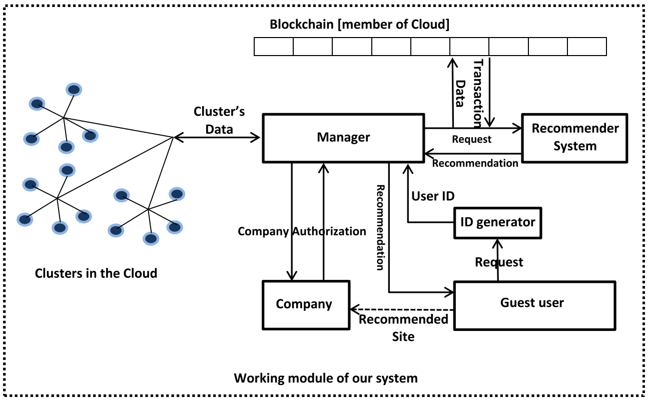

Fig. 2. An application of Private-Rec.

- Three algorithms have been introduced to handle: the request sending mechanism of Guest user, Recommendation generation mechanism, Joining request management, and incentive mechanism. The platform has been evaluated in different recommender systems set up.

Paper Organization: The rest of the paper is structured as follows: Section II outlines the related work. In section III we provide the preliminaries defining our platform's notations, collaborative filtering, and properties. Section IV discusses the description of our platform's protocol and the operating scenario. The protocols are built-in Section V. Section VI introduced security analysis. Computational evaluation appears in Section VII. Section VIII holds some concluding remarks.

II. Related Work

In this section, we discuss some of the related works done in this area.

In [9], Lam *et al.* addressed critical research questions relating to the privacy of a recommender program. They focused on collaborative (automated) filtering based on AI. A detailed discussion of user trust infringement and the possibility of leakage of personal information emerge out of their work. Researchers have taken various actions to address data privacy. Various techniques have been proposed recently. Some recently developed data management platforms utilize blockchain to protect personal data [21].

A stable recommender framework was proposed in [11]. In addition to secure multiparty calculations, the authors suggested a stable recommender network using blockchain. Companies can store consumer data in the system and use blockchain to store data (e.g. preference list, preferences, history of shopping, confidential data such as credit card data). Due to cryptographic functionalities, all data are encrypted and not accessible without customer permission. Companies provide incentives if the customers give permission to access their data for computing recommendation. In their proposal, collaborative filtering operates on the mutual data to figure out an appropriate recommendation, so that the organizations are unable to see the recommendation. Customers are able to receive the recommendation through a notification that gives secure access to the resulting recommendation. While the authors have introduced a secure framework, it does not address the following critical points. First, although the companies are not permitted to access the customer data stored in their proposed system, they are permitted to arbitrarily collect and store customer data without their permission. Second, they assert that the whole computation is carried out in blockchain. But computation in the blockchain is practically infeasible. Third, their fully anonymous system allows an unauthenticated entity to access and create dummy profiles. These dummy profiles can be used to manipulate the rating of their own products. Hence, the idea of fully anonymous customers leaves open the possibilities for fraudulent activities.

A. *Recommender System*

Users of different e-commerce websites become confused to find out the best product due to numerous options, hence the recommender system comes with the solution. Not only e-commerce websites but mobile recommender systems [22] are also becoming more popular for an AI-based personalized recommendation. Increasing revenue is one of the recommender system's primary goals by boosting the product sell. The google news personalization recommender system [23] is another most common personalized recommender system. This system sends recommendation from the users' click history. The basic principle of recommendation is finding the correlation between the user and product-centric activity [24]. Here, product-centered operation means the user's ratings, and a data-driven approach can be used to find correlation from the ranking matrix. Different types of rating scales are described in [24]. Suppose, a 5-star rating scale defined as {-2, -1, 0, 1, 2} determines how much a user likes or dislikes a particular product. It is called an interval-based rating system, where a series of ordered numbers are used to evaluate likes or dislikes. In binary rating systems, 0 or 1 represents likes or dislikes. Another form is a categorized rating scale where the rating scale is set by ordered categorized values such as: Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree.

B. *Blockchain*

We use blockchain at our platform to store the data sharing transaction. Blockchain is a data structure that has the following characteristics: immutability, append-only, organized, open and transparent, safe (identification, authentication, authorization). Blockchain is becoming more common in cryptocurrencies like Bitcoin [25]. It is a decentralized ledger for storing and managing the transaction and historical states. In 2008 Satoshi Nakamoto launched the first blockchain [26]. The explanation for its success is its decentralized role in storing transactions. These transactions are processed and registered without third parties [27]. All the data stored are encrypted and an encrypted result can be found using a Fully Homomorphic Encryption Algorithm on encrypted data. The layout of Blockchain was conceived by linearly sequenced blocks. To ensure consistency and immutability of the chain, each block contains the cryptographic hashes corresponding to the preceding and current blocks. The chaining function ensures that this stable data structure is completely integral. Blockchain can be opened to the public in such a way that anybody can join or it can be fully private where only licensed parties are allowed. These are also classified as Public and Private Blockchain. The two popular methods of validating the transaction on a blockchain are Proof-of-Work (PoW) [26] and Proof-of-Stake (PoS) [28].

TABLE I
TERMINOLOGY TABLE

Notation	Description
GU_i	Guest User
BC	Blockchain
c_i	Clusters in the cloud
RS	Recommender system
ID_g	ID generator
TDS	Temporary data storage
M	Manager
ID_i	ID
$\mathcal{C}$	Cloud of our system
p_i	user preference for a particular item
$\mathcal{P}$	Set of user preferences
req	Request to the recommender system
$\mathcal{A}_{req}(\delta)$	Request to the Cloud
δ^{c_i}	Data of particular c_i
$\mathcal{R}_{\texttt{recommendation}}$	Recommendation generated by RS
J_{req}	Joining request to GU_i

III. PRELIMINARIES

In this section, we first address collaborative filtering, and our platform's properties. We also discussed how to integrate collaborative filtering and privacy. We describe our platform's cryptographic building blocks and cryptographic tools. Then finally, we explain the assumptions that we made to construct our protocols.

The notations used in this paper are listed in Table I.

A. *Collaborative Filtering*

Collaborative filtering [29] is a recommender system technique which gives users a preference based on other users rating. That means if a user wants a recommendation on a particular item or product, she will be given a recommendation based on other users' responses. Here comes the scenario of voting to generate a recommendation. Based on other users' preferences, the model should contain the preference of users. These preferences are taken as a vote from users on different items. Two voting techniques [30] have been introduced in collaborative filtering:

- Explicit Voting: This technique refers to the fact when the system takes the preference p_i of the user on a questionnaire basis. System asks the user to vote for their p_i on a particular item and also on a particular rating set. Upon their feedback or vote, the system generates a recommendation for other users. The system uses a set of preferences to generate recommendations.

$$\mathcal{P} \supset \{p_1, p_2, p_3, p_4, \ldots, p_i\} \tag{1}$$

- Implicit Voting: The system sometimes does not ask any question related to the item or does not provide any rating set. In that case, the system makes a record of the user's action in the system. A log of their action (e.g., browsing data, purchase history) is recorded for generating future recommendations.

Both techniques are being used in collaborative filtering but there are some privacy issues with the implicit voting technique. Usually, users are not notified about the log of their actions being stored by the system. This action tends to create issues like privacy infringement. Also, with this private data, a lot of companies are doing business without taking proper consents from the users.

Apart from the voting techniques, collaborative filtering has two algorithmic approaches to follow:

1) Memory-based Algorithm [31]: Recommendation or prediction for a user's next action is determined by the vote, $vi.j$ which she has cast earlier to the system. If a user i gives a vote on item j, then we can define the mean of votes from the recorded votes,

$$\overline{v}_i = \frac{1}{|I_i|}\sum_{j\in I_i} vi.j \tag{2}$$

Prediction of the next vote of the user depends on some other factors like weights, $w(a, i)$ and pra, i. Pearson correlation coefficient is used to determine the weight, $w(a, i)$.

$$w(a,i) = \frac{\sum_j (va,j - \overline{v}_a)(vi,j - \overline{v}_i)}{\sqrt{\sum_j (va,j - \overline{v}_a)^2 \sum_j (vi,j - \overline{v}_i)^2}} \tag{3}$$

This correlation coefficient is used to predict the user preference for the next vote or for the active user. Prediction for an active user a for item j, pra, j is calculated by the equation below,

$$pra,j = \overline{v}_a + \mathcal{K}\sum_{i=1}^{n} w(a,i)(vi,j - \overline{v}_i) \tag{4}$$

We are using this approach in our platform.

2) Model-based Algorithm [32]: The recommendation of a particular item depends on a model that has been created to generate a classifier so that all the users having similar interests stay in a single model. Previous votes of the user are counted in this algorithm. The model-based algorithm makes recommendations by computing a probability. There are some machine learning approaches to build the model in this technique. Bayesian network, clustering and rule-based are some important approaches. Equations are used to generate the probability of the next vote in this algorithm,

$$pa,j = E(va,j) = \sum_{i=0}^{m} Prob(va,j = i | va,k,k \in I_a)i \tag{5}$$

B. Properties

1) Privacy: Accountability, integrity, pseudonymity, and privacy are the key focus points of this system. The system's design is based on ensuring proper privacy to the users, their interactions, and their data. Some key points of privacy are briefly described below:

1) Pseudonymity: The users cannot recognize each other in the system. Even the system does not refer to users with their identity.
2) Privacy: Activities of a user are not visible to other users.
3) Integrity: Users can own private data. Private data have been restricted through authentication-based access.
4) Accountability: All instances of data access are recorded against the entity accessing the data.

C. Protocol Building Tools

Here we introduce the collaborative filtering technique that we have used to build our protocol. Also, we discuss the cryptographic tool namely, Schnorr signature scheme.

Definition 1. (Cluster-based Collaborative algorithm):[1]Let, Pbk be the likelihood that a (random) consumer is in class k, Pbl The chance of a (random) object being in l class and $Pb_{kl}\longleftarrow$ class k linked to l (Probability of a person in class k being associated with an object in class l)

Here, Y_{ij} be the observed data:

$$Y_{ij} = 1, \text{if user i likes item j and,}$$
$$Y_{ij} = 0, \text{Otherwise.}$$

Let C_i be the class that user i is in and let C_j be the class that item j is in. Model parameters are the base rates for the user and item, Pb_k and Pb_l and the probabilities of a person in class k liking a item in class l, Pb_kl. Then the probability that user i is in class k (i.e., that $C_i = k$) given the model parameters and all the other class assignments is proportional to,

$$Pb_k \prod_l Pb_{kl}^{\sum_{j:C_j=l} Y_{ij}} (1 - Pb_{kl})^{\sum_{j:C_j=l} 1-Y_{ij}}$$

Let, X_{kz} = number of items in class l voted by users in class k.

and, $N_{kl} - X_{kl}$ = number of items in class l not voted by users in class k

$$Pb_{kl} = \beta(X_{kl} + 0.5\delta_{kl}, N_{kl} - X_{kl} + 0.5\delta_{kl})$$

$$[\text{Jeffries prior and beta distribution}, \beta(\text{a,b,c,...})]$$

let, $count_k$ be the number of users in class k. Then user class k membership probability,

$$Pb_k = \frac{\gamma(count_k + 0.5)}{\sum_k \gamma(count_k + 0.5)} \quad [\text{gamma distribution}]$$

and let, $count_l$ be the number of items in class l. for class item k probability,

[1] This definition follows Gibbs sampling.

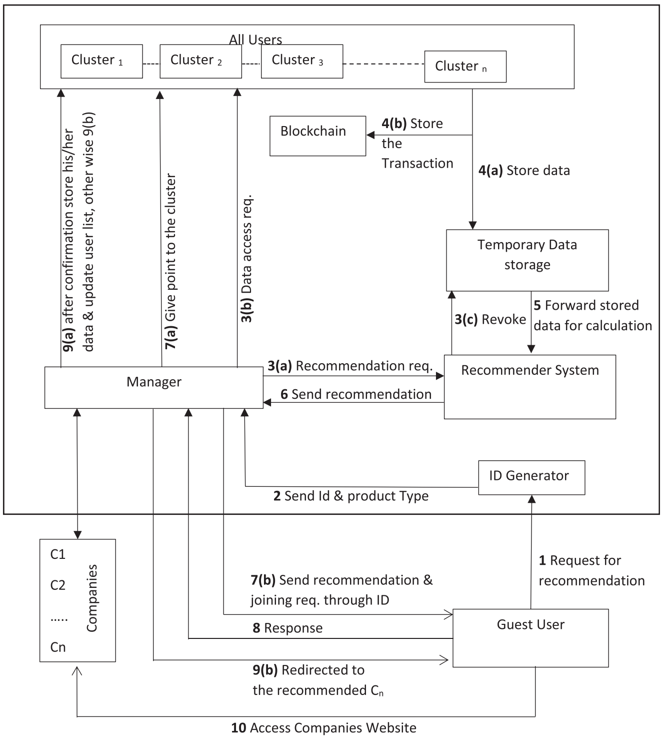
All Users
Cluster 1
Cluster 2
Cluster 3
Cluster n
Blockchain
4(b) Store the Transaction
4(a) Store data
Temporary Data storage
3(c) Revoke
5 Forward stored data for calculation
9(a) after confirmation store his/her data & update user list, other wise 9(b)
7(a) Give point to the cluster
3(b) Data access req.
Manager
3(a) Recommendation req.
6 Send recommendation
Recommender System
ID Generator
2 Send Id & product Type
C1
C2
.....
Cn
Companies
1 Request for recommendation
7(b) Send recommendation & joining req. through ID
8 Response
Guest User
9(b) Redirected to the recommended Cn
10 Access Companies Website

Fig. 3. *Private-Rec*: User-Centric Recommender System Platform.

$$Pb_l = \frac{\gamma(count_l + 0.5)}{\sum_l \gamma(count_l + 0.5)} \quad \text{[gamma distribution]}$$

clustering of objects with multiple attributes could be easily handled with this model. Converging to the true distribution is guaranteed by Gibbs sampling.

Definition 2. (Schnorr Signature Scheme): Let $\mathbb{G}$ be a cyclic group of prime order q and g be a generator of $\mathbb{G}$ in which the DLP is hard.

Let $h : \{0,1\}* \times \mathbb{G} \longrightarrow \mathbb{Z}_q$ be a hash function. The Schnorr signature scheme is defined as follows:

`Key generation:`Let $x \longrightarrow R\mathbb{Z}_q$, and $y = gx$. The private key is x and the public key is y.

`Signature:`To sign a message $m \in \{0,1\}*$, draw $a \longrightarrow R\mathbb{Z}_q$, compute $r = ga$, $c = h(m,r)$, and $s = a + cx$ mod q. The signature is (s,c).

`Verification:`Given a message $m \in \{0,1\}*$, and a claimed signature (s,c), compute $r = gsy{-}c$ and check that $c = h(m,r)$.

In the random oracle model, Schnorr Signature Scheme is secured under DLP assumption.

D. Assumptions

It is assumed that ID_g will generate an ID_i for each GU_i randomly.[2] ID_i will be shared through a secure channel. Manager is the honest entity who will do the rest of the work for GU_i after the request for a recommendation. Upon receiving a request from Manager cloud will send the data to the temporary data storage through a secure channel. Data response from the cloud is recorded in blockchain so that any malicious cluster could be found. GU_i also can claim the wrong recommendation. In that case, we assume that cluster is responsible for this wrong recommendation and we will audit the blockchain against that claim.[3] The recommendation is issued in a digitally signed format for GU_i.[4] Then GU_i will have to verify and get his recommendation. The point will be distributed equally in the cluster to the registered active users and it is assumed that all the registered users are active.

IV. Platform Description: Private-Rec

The architecture as well as the design view of our platform are provided in this section.

A. Overview of Our Platform

Fig. 3 displays our entities, and their functions are briefly defined here:

Cloud ($\mathcal{C}$): $\mathcal{C}$ is used as a data storage in our platform. User details that would be involved in accessing our platform are stored in $\mathcal{C}$. Only M is capable of creating a connection with $\mathcal{C}$. Therefore, all user data is stored in $\mathcal{C}$. Thus, no third party (i.e., companies, unauthorized authorities, etc.) can have access to user data. There are two entities in $\mathcal{C}$-

1) User List (UL): UL stores all users who are keen to join our system. No companies have access to UL. The only entity that can connect to UL is M, and UL is usually modified by M.
2) Cluster: Cluster stores a details unique to the product. There are two parts in a cluster-
    1) Product Type (P_i): In this section the name of the product is stored. Using this P_i, $\mathcal{C}$ will define a specific c_i and they can be given points by M.
    2) Point: The same point would apply to all users in a c_i. Whenever a recommendation, R_i is calculated using the information of a particular c_i then all users of that c_i get incentives by receiving points from M. Users may get coupons or discounts from companies that use these points.

Recommender System (RS): RS computes R_i using the corresponding data and sends it to M.

Blockchain (BC): It is one of the main components of this system. We are using ethereum network. We are keeping the data transaction records on blockchain. A smart contract ensures that whenever a data transaction will happen this will be recorded on the blockchain. BC retains a $\mathcal{C}$ to RS transaction. The intent of this data transaction log is to assert what data was exchanged with which GU_i at what time if any c_i data is tampered with.

[2] Any random function could be used.

[3] We do not define any action against reported cluster user, Clusters might get a deduction by a point.

[4] Schnorr Signature Scheme has been used in this protocol, any light signature scheme could be used.

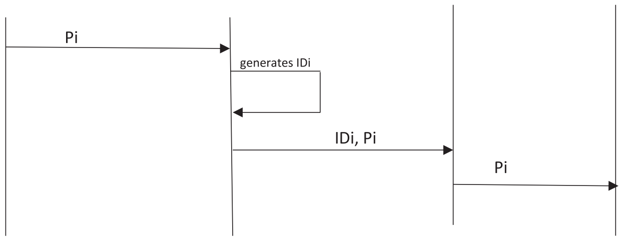

Fig. 4. Low-level view of sending request.

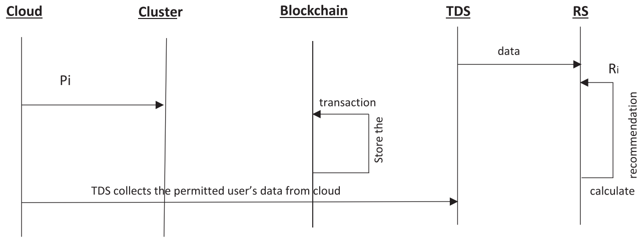

Fig. 5. Low-level view of recommendation computation.

ID Generator (ID_g): Users can link to our platform via a trusted party which is described as ID_g.[5] ID_g receives the GU_i request and produces a special ID_i corresponding to the GU_i request. Upon generation of the ID_i, ID_g sends it with P_i to M. ID_g does not store any ID_i each time a specific one is created and shared with M.

Guest User (GU_i): A Guest user who is new to this system is defined as GU_i. GU_i is capable of connecting with ID_g and M. GU_i requests ID_g for R_i and receives the R_i from M.

Temporary data storage (TDS): TDS is a trusted party of our platform who[6] can temporarily store the data. If M sends a request for data access to $\mathcal{C}$, then $\mathcal{C}$ sends the data to TDS for the specific c_i. TDS waits for a response of $\mathcal{C}$. TDS will stop gathering data after a certain amount of time and will forward the shared data to RS. TDS never permanently stores the data and does not exchange any data with other parties on our platform.

Manager (M): The entire requesting and sending process for R_i happens via the M. It can respond to all other entities. The cycle begins with M obtaining the ID_i and P_i from ID_g and finishes with GU_i being redirected to the company's website or to the website of a specific product. Between these two steps, M will do some other works, such as storing the ID_i, sending a request to RS for R_i, sending a request for data access to $\mathcal{C}$ for sharing with TDS the particular c_i. M receives R_i from RS and then gives a point to the c_i that shared the data with TDS to compute R_i. M connects to GU_i by sending GU_i the R_i and the connection request to that device. M has a UL update capability if GU_i approves the order. Companies connect to this platform through M.

Companies: Any e-commerce site can join with this platform through registration. After the registration, companies will be an entity of this platform and will be able to connect with the platform through M. Users of the companies will be added to this platform because users will get the recommendation through this platform.

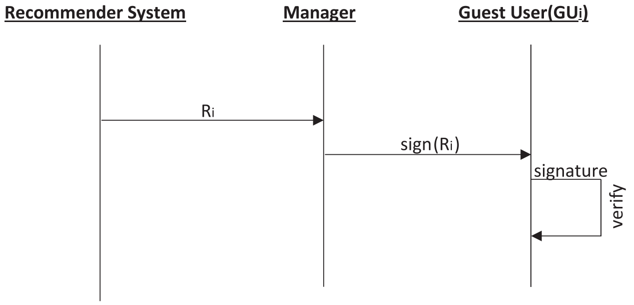

Fig. 6. Low-level view of recommendation sending.

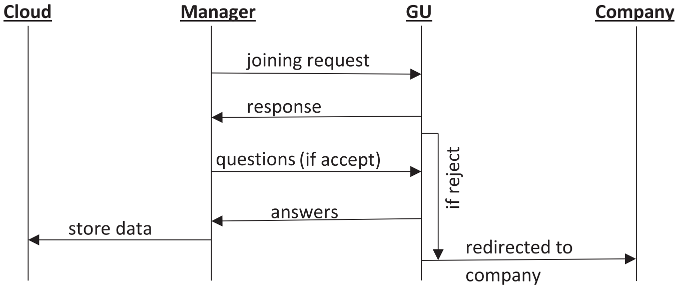

Fig. 7. Low-level view of adding a new member.

B. Formal Description

1) Request Sending to Our Platform: Fig. 4 displays the low level view of request sending. Using P_i, GU_i will submit the request to our platform. ID_g receives the order, producing a special ID_i corresponding to P_i. ID_g sends out ID_i and P_i to M. M will store the ID_i and forwards P_i to $\mathcal{C}$.

When computing ID_i, ID_g will use a random function.

Random(P_i)=ID_i

2) Recommendation Computation: After receiving P_i from M, $\mathcal{C}$ will look for the c_i that has the same product as P_i. All users of the corresponding c_i must then offer permission for data access. At the time of entering our platform, this consent will be taken from the users. The data fetching will then continue from $\mathcal{C}$ to TDS. After a certain amount of time TDS would cease data storage and transfer the shared data to RS. Fig. 5 displays a low-level view of the R_i computation procedures.

Our platform will use a collaborative filtering method to compute R_i. Here collaborative filtering will be carried out in two steps, firstly gathering data from the users who have the same P_i as GU_i. This data collection process will be done by TDS, with cloud discovering that the specific c_i has the same P_i. Secondly, RS will compute the R_i corresponding to GU_i.

3) Sending the Recommendation: Fig. 6 demonstrates the low-level view of our platform sending the R_i to GU_i. RS will forward R_i to M, then M sends R_i to GU_i. M must recognize GU_i from the stored ID_i, referring to R_i, and

[5] In this platform, ID_g is a trusted party.

[6] In this platform, TDS is a trusted party.

Algorithm 1: Recommendation Request by the Guest User, GU_i.

```
Input: $ID_i$
Output: $\delta^{c_i}$
    $\tau \longleftarrow$ time for waiting state for $TDS$
    $\mathcal{T} \longrightarrow$ timestamp of requesting, $TDS$   {By Recommender
    system, $RS$}
 1: $ID_g \xrightarrow{ID_i, prod_{type}} Manager$
    Request for recommendation by $GU_i$, $prod_{type}$,
    Product type requested by $GU_i$
 2: $\mathcal{C} \xleftarrow{req} M$
 3: $RS \xleftarrow{\mathcal{A}req(\delta)} Manager$
    {Steps: 2 and 3 are concurrent requests.}
 4: Request, $RS$
    {$\tau$ will be started by the $TDS$}
 5: while $\tau \neq 0$ do
 6:   if ($\mathcal{A}req(\delta) \in c_i$) then
 7:     {$\mathcal{C}$, searches for $\mathcal{A}req(\delta)$ corresponding cluster, $c_i$)}
 8:     $\delta^{c_i} \longrightarrow TDS$
 9:     $\mathcal{C} \xrightarrow{\texttt{Transaction}} BC$
        {Steps: 8 and 9 are concurrent.}
10:   else
11:     $\tau = 0$
12:   end if
13: end while
14: if ($\tau = 0$) then
15:   return $\emptyset$
16: end if
```

GU_i will get a signed R_i. We use Schnorr Signature Scheme here. GU_i must validate the signature after it has obtained R_i. If the authentication succeeds the GU_i will be routed by M to the platform recommended the company's page or website.

4) Joining Request to GU_i and Adding GU_i to This Platform: Fig. 7 demonstrates the joining request procedures to GU_i and the introduction of GU_i to this platform. When GU_i receives the R_i concurrently, GU_i receives another request from M, which is the invitation to join this platform. If GU_i approves the request then GU_i will address some queries, such as his favorite items, etc., then all these details and the shopping backgrounds of GU_i will be stored by M in $\mathcal{C}$. By acknowledging the attached request users also offer permission to access the data.[8] Finally, by updating the user list by M and redirecting it to the company's page or website after updating the UL, GU_i becomes a member of our platform. If GU_i does not approve the submission, then M redirects GU_i to the site or website of the company.

V. Protocol Construction

A. Algorithm 1 for Recommendation Request

Algorithm 2: Recommendation Generation.

```
Input: $\delta^{c_i}$
Output: $R_i$
 1: $TDS \xrightarrow{\delta^{c_i}} RS$
 2: $RS \xrightarrow{generate} R_i$
    {$RS$ will generate recommendation, $R_i$}
 3: $R_i \longrightarrow Manager$
 4: if ($R_i \in \delta^{c_i}$) then
 5:   $sign(R_i) \longrightarrow (R_i, signature) - pair$
 6:   $send, R_i, signature) - pair \longrightarrow GU_i$
 7:   $send, J_{req} \longrightarrow GU_i$
      {Steps: 6 and 7 are concurrent.}
 8:   $point, p \longrightarrow \mathcal{C}$
    {Cloud, $\mathcal{C}$ will get the incentive as $point$, $p$ for corresponding $c_i$}
 9: else
10:   return $\emptyset$
11: end if
```

Anonymous users will be able to request for recommendations in our system. Upon that request req, our system will generate an ID_i for him. To keep track of the user and also for further request our system needs this ID_i.

$$ID_g \xrightarrow{\{\texttt{unique}, \texttt{random}\}} ID_i$$

Then M forwards req to RS and submits a data access request $\mathcal{A}_{req}(\delta)$ to $\mathcal{C}$. Then $\mathcal{C}$ searches for the c_i which has the corresponding data. $\mathcal{C}$ is made up of clusters.

Upon getting the corresponding c_i, $\mathcal{C}$ will forward δ to TDS and store the `Transaction` on BC.

$$\mathcal{C} \xrightarrow{\texttt{Transaction}} BC$$

After getting the req, RS will request TDS for a data response and on getting the particular c_i's data δ^{c_i}, $\mathcal{C}$ will forward δ^{c_i} to TDS. From requesting to response with the δ^{c_i}; TDS has a fixed time τ to wait for $\mathcal{C}$'s response. If $\mathcal{C}$ forwards δ^{c_i} in τ then TDS will forward δ^{c_i} to RS for recommendation computation. Algorithm 1 ends at the point where, TDS have the δ^{c_i}.

B. Algorithm 2 for Recommendation Generation

In Algorithm 2, RS will get the δ^{c_i} to generate recommendation. After generating R_i, RS sends it to the M. Then M digitally sign the R_i and send to GU_i along with the joining request, J_{req}.

$$sign(R_i) \longrightarrow (R_i, signature) - pair$$

C. Algorithm 3 for Joining Request and Incentive Distribution

$\mathcal{C}$ will get the incentive as p which will be forwarded to the corresponding c_i. It will be able to use p while purchasing any item.[9] Signed version of data need to be verified by the GU_i.

After that GU_i will get the J_{req} to join our system. If it joins our system it will be added to the c_i from which it was recommended with R_i.

[8] Platform must inquire separately about the permission.

[9] All the active users of the corresponding cluster will be able to use the p on the time of purchasing any item through our system.

Algorithm 3: Joining Request and Incentive Distribution.

```
Input: $p$, $R_i, signature) - pair$
Output: New $GU_i$
  {If guest user accepts the joining request.} $GU_i \longleftarrow J_{req}$
for (j=0 to i) do
  if ($c_i$ = corresponding cluster) then
    $c_i \longleftarrow p$
  end if
end for
if $(verify_{sign}(R_i, signature) - pair)$= verified, $R_i$) then
  $GU_i \xrightarrow{R_i} Recommended\ company\ site$
else
  return $\emptyset$
end if
if $(GU_i \xrightarrow{accepts} J_{req})$ then
  go to, Company, $cmp$'s site
else
  return $\emptyset$
end if
```

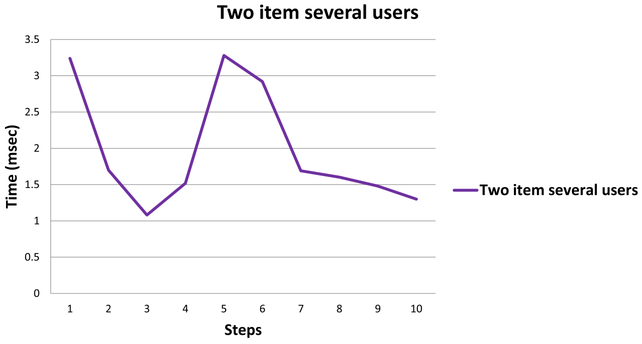

Fig. 8. Item to item based with 2 item and several users.

VI. Security Analysis

- Pseudonymity: GU_i is identified by M while sending R_i. Any other entities who are connected with our platform (i.e., company) are not capable to identify GU_i during interaction with our platform or blockchain transaction. This provides pseudonymity.
- Privacy: The privacy of users in UL is preserved in the system. The information of GU_i is not preserved by our platform during its computations. M acts as an intermediary between GU_i and the company. GU_i will receive R_i but will not be able to trace back the source, user(s) of the data.
- Integrity: The user data is private for all users. M makes an access request along with the R_i to GU_i. GU_i accepts the access request and permits M to store and access his data. If not, the data will not be stored by our platform and the R_i will not be available to any other entity in the system. Furthermore, all data access for computing R_i are stored as transactions on BC. Because of this access procedure and immutable recording of data access as transactions in the BC, the integrity of each user's data is preserved.

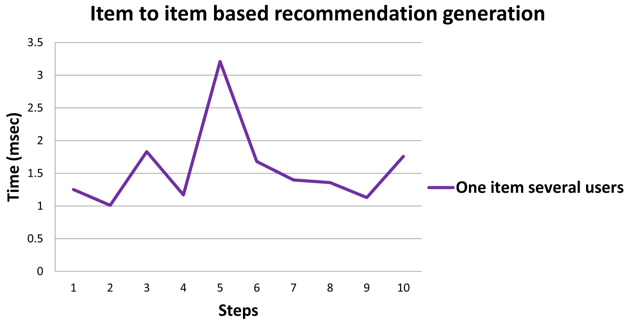

Fig. 9. Item to item based with 1 item and several users.

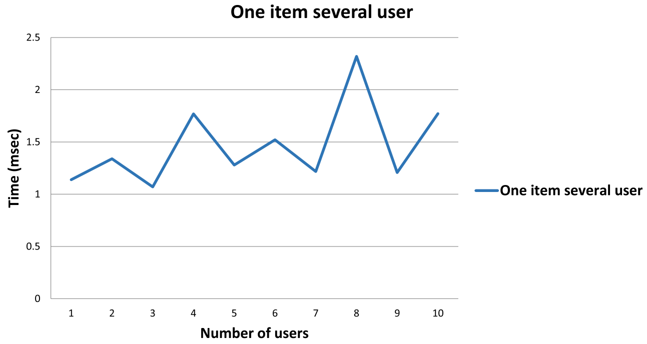

Fig. 10. Our system's recommendation generation with the increasing number of users.

- Accountability: M will be held responsible for any user data access. The data transaction is processed on the BC after M's request for access is accepted by GU_i. So, any transaction of the data can be tracked by the user.

VII. Empirical Study

A. Computation and Evaluation

We set up a protocol evaluation environment using an Intel Core i3, CPU-3.5 GHz computer, 4 GB RAM, Win10, 64-bit OS. We analyze a rating based item to item recommender system with a rating scale of 1,2,3,4,5.We measure the time to generate the recommendation for every guest user. Fig. 8 shows the time for generating a recommendation for 2 items with several users. The vertical axis shows the serial number of the recommendation request and the horizontal axis shows the time for generating the recommendation. The lowest time is nearly 1 ms for the third request and the highest time is nearly 3.3 ms for the fifth request.

Fig. 9 shows the time of recommendation generation in millisecond(ms) versus the serial number of the recommendation request from 1 to 10. From the resultant graph, we can say that it is a nonlinear graph. The lowest time is taken by the second request which is nearly 1 ms and the highest time is taken by the fifth request is nearly 3.20 ms.

We have built our system and measured the time to produce a recommendation, where the system recommends items that others like and are close to the interest of the user. For

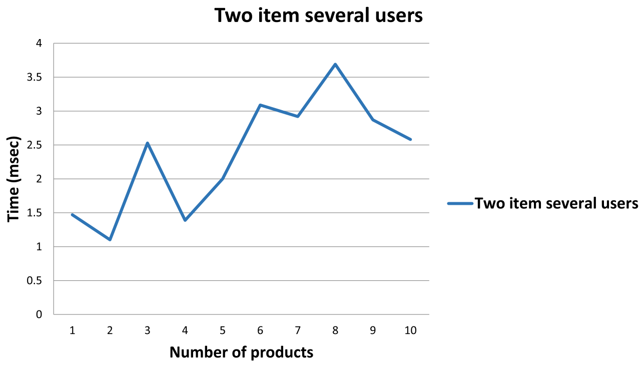

Fig. 11. Our system's recommendation generation with increasing number of products.

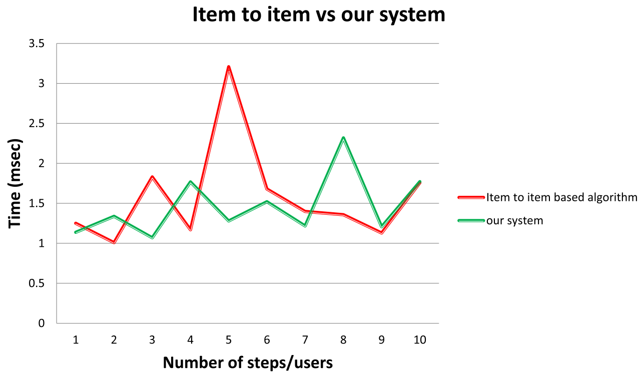

Fig. 12. Recommendation time comparison between our system and item to item based recommendation algorithm.

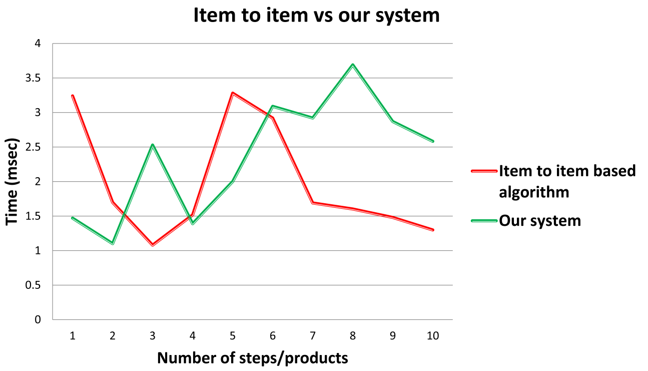

Fig. 13. Recommendation period distinction between a fixed product number and a variable product number.

calculating the time for generating recommendation in our system, we set up our environment with Mozila/5.0(Windows NT 10.0; Win64; x64), AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 OPR/57.0.3098.106103. 106.2.2. Fig. 10 shows the recommendation time versus the number of customers who reviewed our products. Here, the number of the product is the same in every step but the number of reviewers is increasing. We are trying to find out the effect of those products on the recommendation process. From the resultant graph, we can say that it is a non-linear graph. Increasing reviewer numbers has no direct impact on recommendation computation. Fig. 11 displays the Millisecond (ms) suggestion period for every 10 steps. We have the constant number of users in this setting and the number of the product is increasing at every step. Here, we are trying to figure out the effect of the the products on the time for a generation of recommendations. From the resultant graph, we find that it is a nonlinear graph There is no direct impact on recommendation generation for increasing products.

In Fig. 12 we show the analysis of the Fig. 9 and Fig. 10. Though our system guarantees the privacy in average it does not take more time than the item to item based algorithm. The highest time to generate recommendation in our system is nearly 2.2 where the item to item based take 3.2 which is too high than our system's recommendation generation.

In Fig. 13 we analyze Fig. 8 and Fig. 11 together. Here the scenario is different because the product number is increasing in our system. For items, to the item-based algorithm, the product number is constant that is why at some places our system takes more time to generate recommendation than the item to item based algorithm.

In this platform, if we have a small number of users at the initial stage then it will not affect running the blockchain. Because the platform operator will participate in the blockchain and will operate all the transactions. Only keeping the transaction records will help this platform to support all the operation in terms of time complexity.

VIII. Conclusion

Recommender system is one of the most popular examples of Artificial Intelligence (AI) based system. The recommender system has a great influence on the revenue of online businesses. Since choosing one item from a large amount of product list is challenging task for customers, users also get benefit from recommender systems. In this paper, we propose a secure platform for a recommender system that guarantees customers data privacy using a blockchain system. Collecting customer's data without ensuring privacy is one of the major problems in recommender systems. Here, we described a solution to handle this situation. In our system, companies are not allowed to store or access customers' data. The whole process of storing data and sending recommendation is done by our platform. Our platform cryptographically guarantees privacy by using blockchain to store all the data transactions. The digital signature confirms the authenticity of customers. Our platform and customers are not fully anonymized. Thus, it is not possible here to build dummy profiles to boost the ranking of their own company. To the best of our knowledge, a user-centric platform guaranteeing data protection for the users through cryptographic techniques is first investigated in this work. Our future research will be to test the new framework for broad-scale consumers, companies, and big data.

References

[1] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997.

[2] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, 2016, Art. no. 13.

[3] P. Dutta and A. Kumaravel, “A novel approach to trust based identification of leaders in social networks,” *Indian J. Sci. Technol.*, vol. 9, no. 10, Mar. 2016.

[4] J. Davidson *et al.*, “The youtube video recommendation system,” in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 293–296.

[5] G. Linden, B. Smith, and J. York, “Amazon. com recommendations: Item-to-item collaborative filtering,” *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[6] J. Chen *et al.*, “Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering,” *Complex Intell. Syst.*, vol. 6, no. 1, pp. 147–156, 2020.

[7] M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” *Artif. Intell. Rev.*, vol. 13, no. 5–6, pp. 393–408, Dec. 1999.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.

[9] S. Lam, D. Frankowski, and J. Riedl, “Do you trust your recommendations? an exploration of security and privacy issues in recommender systems,” *Emerg. Trends Inf. Commun. Secur.*, pp. 14–29, 2006.

[10] T. Guardian, “Facebook to contact 87 million users affected by data breach | technology | the guardian,” 2018, Accessed Dec. 2, 2019. [Online] Available: https://www.theguardian.com/technology/2018/apr/08/facebook-to-contact-the-87-million-users-affected-by-data-breach, 08 2018.

[11] R. Frey, D. Wörner, and A. Ilic, “Collaborative filtering on the blockchain: A secure recommender system for e-commerce,’in *Proc. 22nd Amer. Conf. Inf. Syst.*, 2016.

[12] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, “Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability,” in *Proc. IEEE 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, 2017, pp. 468–477.

[13] A. Felt and D. Evans, “Privacy protection for social networking platforms,” presented at the Workshop on Web 2.0 Security and Privacy, Oakland, CA, USA, May 22, 2008.

[14] A. Al Omar, R. Bosri, M. S. Rahman, N. Begum, and M. Z. A. Bhuiyan, “Towards privacy-preserving recommender system with blockchains,” in *Proc. Int. Conf. Dependability Sensor, Cloud, Big Data Syst. Appl.*, 2019, pp. 106–118.

[15] J. Gong *et al.*, “Hybrid deep neural networks for friend recommendations in edge computing environment,” *IEEE Access*, vol. 8, pp. 10 693–10 706, 2020.

[16] X. Liu, G. Wang, M. Z. A. Bhuiyan, and M. Shan, “Towards recommendation in internet of things: An uncertainty perspective,” *IEEE Access*, vol. 8, pp. 12 057–12 068, 2020.

[17] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, “Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6266–6278, Jul. 2020.

[18] Z. Tian, X. Gao, S. Su, and J. Qiu, “Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3901–3909, May 2020.

[19] Z. Tian *et al.*, “Real-time lateral movement detection based on evidence reasoning network for edge computing environment,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 7, pp. 4285–4294, Jul. 2019.

[20] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, “Blockchain technology: Beyond bitcoin,” *Appl. Innov.*, vol. 2, pp. 6–10, 2016.

[21] R. M. Frey, D. Vuckovac, and A. Ilic, “A secure shopping experience based on blockchain and beacon technology,” in *RecSys Posters*, 2016.

[22] F. Ricci, “Mobile recommender systems,” *Inf. Technol. Tourism*, vol. 12, no. 3, pp. 205–231, Jan. 2011.

[23] A. S. Das, M. Datar, A. Garg, and S. Rajaram, “Google news personalization: Scalable online collaborative filtering,” in *Proc. 16th Int. Conf. World Wide Web.*, 2007, pp. 271–280.

[24] C. C. Aggarwal, “An introduction to recommender systems,” in *Recommender Systems.* Berlin, Germany: Springer, 2016, pp. 1–28.

[25] M. Iwamura, Y. Kitamura, T. Matsumoto, and K. Saito, “Can we stabilize the price of a cryptocurrency?: Understanding the design of bitcoin and its potential to compete with central bank money,” 2014.

[26] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2009.

[27] M. Swan, *Blockchain: Blueprint for a New Economy.* ” Sebastopol, CA, USA: O’Reilly Media”, 2015.

[28] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake,” 2012. [Online]. Available: https://www.chainwhy.com/upload/default/20180619/126a057fef926dc286accb372da46955.pdf

[29] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: An open architecture for collaborative filtering of netnews,” in *Proc. ACM Conf. Computer Supported Cooperative Work.*, 1994, pp. 175–186.

[30] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Adv. Artif. Intell.*, vol. 2009, 2009, Art. no. 4.

[31] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, “Probabilistic memory-based collaborative filtering,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 56–69, Jan. 2004.

[32] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” *ACM SIGIR Forum*, vol. 51, no. 2, 2017, pp. 227–234.