

# Trust-Based Collaborative Filtering Recommendation Systems on the Blockchain

# Tzu-Yu Yeh, Rasha Kashef

Electrical, Computer, and Biomedical Engineering Department, Ryerson University, Toronto, Canada Email: tzuyu.yeh@ryerson.ca, rkashef@ryerson.ca

How to cite this paper: Yeh, T.-Y. and Kashef, R. (2020) Trust-Based Collaborative Filtering Recommendation Systems on the Blockchain. *Advances in Internet of Things*, **10**, 37-56. https://doi.org/10.4236/ait.2020.104004

Received: September 4, 2020 Accepted: October 10, 2020 Published: October 13, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). http://creativecommons.org/licenses/by/4.0/

**O** Open Access

(cc)

# Abstract

A blockchain is a digitized, decentralized, public ledger of all cryptocurrency transactions. The blockchain is transforming industries by enabling innovative business practices. Its revolutionary power has permeated areas such as banking, financing, trading, manufacturing, supply chain management, healthcare, and government. Blockchain and the Internet of Things (BIOT) apply the usage of blockchain in the inter-IOT communication system, therefore, security and privacy factors are achievable. The integration of blockchain technology and IoT creates modern decentralized systems. The BIOT models can be applied by various industries including e-commerce to promote decentralization, scalability, and security. This research calls for innovative and advanced research on Blockchain and recommendation systems. We aim at building a secure and trust-based system using the advantages of blockchain-supported secure multiparty computation by adding smart contracts with the main blockchain protocol. Combining the recommendation systems and blockchain technology allows online activities to be more secure and private. A system is constructed for enterprises to collaboratively create a secure database and host a steadily updated model using smart contract systems. Learning case studies include a model to recommend movies to users. The accuracy of models is evaluated by an incentive mechanism that offers a fully trust-based recommendation system with acceptable performance.

# **Keywords**

Blockchain, Recommendation Systems, Smart Contract, Predictions, Accuracy

# **1. Introduction**

Internet users have experienced a 40% increase in information over the past two decades, reaching the number count of 3.2 billion. Though many avenues have

been opened courtesy of the increased knowledge, it is hard to decide which information is correct, most accurate, or most recent to use. Recommender systems are introduced to assess users to obtain accurate decisions tailored to their preferences.

Recommendation systems play an integral part in many applications, primarily e-commerce, to encourage people to purchase more items based on historical remuneration, purchase behaviors, and ratings. Nowadays, it will be impossible to construct an advisory system without a massive amount of data. The problem usually occurs for small and medium-sized businesses since the required amount of data cannot be collected due to insufficient resources. The recommender systems' quality can be profoundly affected by the adopted model for prediction, input data, and context. Security and trustworthiness are not well-addressed for many online retailers to ensure customer privacy and data protection. The decentralization architecture in the blockchain technology is useful primarily due to its ability to enhance security issues, which has been a long-time problem with the online transactions [1] [2]. The concept was first introduced by Bitcoin, through a cryptocurrency, but it was recently adopted by other international organizations with operations majors in information sharing or financial transactions [3]. The issues of security and trustworthiness are both solved based on proof of work (POW) and the public ledger between two parties, as witnessed with Bitcoin or Ethereum. For the Proof-of-Work, the transactions are initiated using particular individual nodes known as "miners," promoting Personal Data Control (PDC). Alternatively, the sale may be executed using a private or public key distributed to various participants. The public ledger consists of an immutable chain of transactions, which ensures that in any case the records or recommendations by a participant are tempered with, the entire operation becomes invalidated.

Inspired by [4], a recommendation engine that incorporates smart contracts [5] [6] and recommender systems are introduced. However, they used simple mathematical models and functions in the smart contract, *i.e.*, the weighted average and simple average methods, to acquire the recommended score for an item, which resulted in inaccurate recommendations. One of the main shortcomings of their work is that the developed model may repeatedly recommend the identical item multiple times. The ratings may also be fake due to possible massive negative reviews and purpose sales from competitive companies. To overcome these drawbacks, in this paper, we have developed collaborative-filtering recommendation algorithms using blockchain technology. We use a similarity matrix to calculate the recommended score for items. A similarity represents the relation between each user such that a higher similarity will result in a more significant impact on recommendation scores. The proposed rating system provides the most proper recommendation to users based on the user's characteristics regarding items. Our proposed method considers each user's preference and offers the best recommendation. The proposed system computes new blockchain addresses, sends notifications to the company, automatically uses collaborative filtering to process the data to provide recommendation, and finally improves visualization of the recommendations by both user and organization. The proposed system allows the whole aspect of consumer profiles to be more secure and private. Also, it provides consumers with an opportunity to operate anonymously, this confidentiality is guaranteed with blockchain engineering.

This paper is organized as follows: In Section 2, a review of the literature in the area of the recommendation system is summarized, as well as a discussion on the different types of collaborative filtering algorithms. In Section 3, a back-ground on blockchain technology and its impact is introduced. In Section 4, the proposed trust-based recommendation system is introduced. In Section 5, experimental analysis and results are discussed. Section 6 provides conclusions and future directions.

# 2. Recommendation Systems

In the era of e-commerce, recommender systems [7] [8] [9] have become vital tools as people seek recommendations for the best items to purchase. A recommender system filters information to predict a user's preference and tailor suggestions to the needs or desires of a specific user. However, the need for recommender systems has been amplified by the overload of information online. Recommender systems are divided into different categories [10], including collaborative filtering, model-based filtering, content-based, context-based, and hybrid-based recommendation systems [11].

# 2.1. Collaborative Filtering

Collaborative filtering systems are based on the user's and item's historical data. They can be classified into item-based or user-based [12]. A user-based system allows searching for like-minded users assuming users have similarities, for instance, similar questions. The item-based filtering techniques search for similarly rated items by other users [13]. The item-based recommender solves the most significant challenge of inefficient scalability with user-based algorithms. The main limitation of collaborative filtering systems is that it suffers from limited data information expandability and the problem to a cold start [12]. Mansur *et al.* [14] have observed common challenges across most collaborative filtering algorithms, including difficulty in providing recommendations to new users, limited trust in the conclusions made from small data, user data privacy, and sparsity of data [15] [16].

#### 2.2. Model-Based

In these systems, patterns that depend entirely on training data through the designing and developing models using machine learning and data mining algorithms [17] are discovered. In [18], learning models are first developed, and data is then provided to a collaborative filtering model by cognitive filtering of the output from those learned models. These systems offer integrated reasoning for the recommendation; however, model building is relatively expensive. This is mainly because the model takes a significantly more extended period to formulate and requires fresh computation in case data matrix changes. This occurs every time a new user inputs a rating. Typically, tiny changes remain unprocessed. However, when they become quite large, the entire model needs to be reformulated [13]. Several model-based filtering includes classification, latent model, clustering [19], and Markov decision process.

#### 2.3. Content-Based

In content-based filtering, a system is based solely on content, and therefore, its focus is on the product features. It creates the user's profile by use of previous reviews. These methods show the actual interest a user has and help predict what they might want to do at any given time. It accurately indicates the users' interest [20]. This system conditionally adapts to the likes and dislikes of the users through its adaptive filtering. It provides a clear comparison between the content provided by the users' interests and the item description's materials.

#### 2.4. Context-Based

This system is essential in delivering more information about users with different preferences. Various techniques are used by a context-based recommendation system to acquire situational information about users. The users' data can be obtained from the devices they are using, which shows for example the time of the day, which helps provide recommendations. The information derived from the user's device location can provide much information essential for tourists, like providing weather conditions and routes [21]. Context-based recommender systems have opinion-based recommender systems, *i.e.*, users are given a chance to provide a review and feedback on the system's interaction for future improvements and advancements. These methods are most applicable to instances with known data among individuals, such as name, location, and descriptions.

# 2.5. Hybrid-Based

Hybrid-based recommendation systems combine different recommender systems to acquire better outcomes [17] and overcome problems in current recommendation systems such as data limitation, cold start, and expandability. The various recommender techniques, combined with collaborative filtering to achieve the best outcomes, are mostly the content-based approaches. Users are part of a group after those with the same interests are put together [12]. The problem of scalability would be solved mainly by the reduction of data size during recommendation. The clustering of users with similar rating patterns also helps decrease neighborhood scope during recommendations, thus improving online performances [22].

# 3. The Blockchain

Blockchain is a decentralized ledger (either public or private), containing various kinds of transactions in a Peer-to-Peer (P2P) network arrangement system [23] [24] [25]. The P2P network usually accesses several computers at once, but with no alteration of the participant's data without obtaining consensus for the connected network system. A typical blockchain has a P2P network with a decentralized database. Each transaction activities have a public key used to identify each of the players uniquely. Each block in the blockchain has its own time and hash. It is possible to validate malicious attacks directed to the Blockchain. A blockchain is always an immutable open-only transaction application with a replication of network nodes. Every single block of transactions refers back to its processor. There must be a providence of proof-of-work by every new block to be identifiable and acknowledgeable by other network users. Bitcoin, the first-ever peer-to-peer electronic cash system, is built on blockchain technology. Every open blockchain is attached to a specific price to offer financial incentives to help adhere to the laid down protocols and expound the blockchain [26].

## 3.1. The Blockchain Models

The blockchain engineering is characterized by various consensus models that help arrange transactions chronologically to ensure that the public ledger functions work efficiently without a broad geographical and decentralized scope. These models include proof-of-work, proof-of-stake (POS), proof-of-elapsed-time (PoET), and byzantine fault tolerance (BFT). The proof of stake model allows the consumer to obtain products in abundance [27]. Instead of using the puzzle and mathematical questions to generate a winning value, the POS recommends the use of cryptocurrency, which can, in turn, be used to purchase bulk blocks to assist in creating the blockchain [28]. The POS is also used to improve the proof of work (POW) model. On the other hand, the PoET model is majorly used by the cryptocurrency, bitcoin, to generate self-generated winners and form the next block [1]. However, since this is performed at the organizational level, the model incorporates the use of the Trusted Executive Environment (TEE) panel to promote fairness at all stages of the transactions. The TEE takes the role of validating the leading nodes amongst various participants. Still, it must generate a proof and send it to other nodes, in a way that seems realistic, convincing, and welcoming. A fault is usually occurred when participants experience the same problem simultaneously but in various ways. Therefore, it is difficult for a single participant on a node to identify whether the fault was due to overall system failure since no proper information is available. In such cases, users on various nodes are required to utilize the byzantine fault tolerant (BFT) model to discuss and identify before setting up the new block so that they declare if the node has failed or should be eliminated from the decentralized network system.

# 3.2. Blockchain and Recommendation Systems

Companies gather users' data and analyze it to develop better customer needs,

thus ensure customer value in an organization. Sharing customer's data with organizations has benefits for both the customers and the organization [29]. However, there is still an issue with the privacy of information collected by organizations. Companies are working towards addressing this issue with the use of many resources and advancements. For instance, organizations provide a privacy policy to its users, which helps in improving transparency. Additionally, there have been many efforts to reduce privacy tension [30], including storing and mining data, which ensure users' privacy.

The blockchain technology is decentralized systems that ensure peer to peer encryption of transactions [31] [32]. The technology is immutable because information cannot be tampered with once it is confirmed in the order [33]. Since the development of the technology back in 2008, multiple companies worldwide are increasingly adopting blockchain technology in storing and transferring data [30].

Blockchain technology is secure because of its features, including encryption, immutability, and data mining. The blockchain technology has improved the security and transparency of systems in the modern world. Lisi *et al.* [4] suggest a smart contract framework that is compatible with recommender systems. In this framework, users have a simple access mechanism whenever they want to rate a product. They used the Ethereum and Ropsten platforms to conduct their tests to provide recommendations. Their system helps prevent an attack on massive negative reviews that are usually helpful in broadening user information before making a purchase decision. Such integration offers a secure and private platform, where beacon and blockchain technologies are applied simultaneously.

# 4. Trust-Based Recommendation Systems on the Blockchain

In this paper, we have developed a trust-based collaborative filtering recommendation system using the blockchain technology. In this system, a blockchain holds records of transactions, under no specific jurisdiction or authority. As global preference and relevance are the main pointers when making suggestions to clients, the provided ratings influence the purchaser's decision. Thus, this data can create a rating system on the blockchain in the form of a smart contract that users can use when selecting new products to purchase. The goal of the recommender system is to provide specific recommendations of items to users who are interested in those items while preserving user's privacy. The application of the multiple collaborative-filtering recommenders in this paper ensures in promoting the recommendation of accurate information to the end-users. The main components of the proposed system include a smart contract, and the Meta Mask.

#### 4.1. The Smart Contract System

We used the Ethereum (ETH) platform as the most common open-source system that features the smart contract element. The Ethereum platform is a practical application for blockchain, creating new kinds of money and digital assets. The platform constructs decentralized property and virtual environments that are managed collectively. Web applications are supported by Ethereum smart contracts; therefore, instead of using a centralized system such as a bank, these applications count on the blockchain for storage and logic of the program, which allows any user to connect to the public Ethereum network. This technology is called Dapp (Decentralized Applications), which is encapsulated in a smart contract and executed against the blockchain. The smart contract function in blockchain technology uses the rating system to place some products and brands higher in the ranking than others. The natural selection path is that consumers will select the product or the brand with the best rating. The most common algorithm in this approach is the Proof of Work (PoW) algorithm. In this case, the smart contract prevents modification of the main blocks, and the peer-to-peer review gets replicated across the network.

## 4.2. The Metamask

The core technology associated with Ethereum blockchain is the smart contract system, which is considered as a digital protocol that is automatically activated if the protocol meets the required conditions. It is witnessed as using "If, else..." command when coding. By associating the Ethereum blockchain application, smart contracts can avoid uncertainty and complexity from third-party organizations such as banks. It can bring transparency and efficiency to both sides. The most well-known knowledge associated with smart contract systems is Metamask, which is used in this paper. Metamask considers the wallet and Ethereum browser, allowing users to directly interact with smart contracts or dapps without installing extra software and blockchain. **Figure 1** shows how the Metamask relates to the blockchain. Each code execution must be afforded by the user, with a dedicated currency that has gas. As each transaction is linked with an Ethereum account, the required gas is calculated by converting the ETH cryptocurrency.



Figure 1. The role of metamask in the EVM, Client's browser and records of transaction.

# 4.3. The Proposed Recommendation System

In this paper, by using advanced resources in recommendation systems, blockchain platforms, and prediction markets, we have designed framework to gather numerous amounts of data, allow participants to potentially benefit, and host a mutual recommender learning model as a public resource. Innumerable participants can collaboratively train the model as the model is open and free for others to use for additional purposes while maintaining privacy. This is achieved mainly by three configurable components, as illustrated in **Figure 2**, including the incentive mechanism, the data handler, and the machine learning model [27]. The incentive mechanism authorizes the payment; the Data Handler stores data into the blockchain, which assures all future uses are available and not restricted to the smart contract. Finally, the recommender learning model is updated by training collaborative algorithms. Initially, the smart contract is generated and accepts "AppendData" behaviors from users. Each of these components is discussed next.

#### 4.3.1. The Incentive Mechanism

Blockchains enable users to share model parameters, information such as new movie titles, new words, and new images are used to update existing models hosted by a group of users or a specific person. In this paper, the prediction incentive mechanism is introduced to activate transactions or allow other actions. In the process of appending data, validation in the incentive mechanism is required. In our proposed method, users can cultivate the model's accuracy by submitting the data. The baseline is proposed with no financial incentives in any form to deduct the barrier to entry. The points can be recorded on-chain in a smart contract by taking the user's wallet address as identification, such as submitting data with non-repetitive labels or submitting various data samples. These metrics can be productively expanded off-chain or computed onchain.



Figure 2. The three main components in the proposed system [27].

#### 4.3.2. The Data Handler

Blockchain technology encounters two distinct features in the data handling field. Initially, a plentiful source of information (interaction patterns between accounts) is offered by the blockchain network or stored in the blockchain. Secondly, blockchains can agree with trustworthy data analytic environments for diverse data sharing by appending the correct element into data and coming up with analytical models. The data handler will store the data and then run the update function on the blockchain. The primary purpose of the data handler is to ensure the accessibility for any user, and that data is not restricted to the smart contract.

#### 4.3.3. The Prediction Model

Prediction is made off-chain by running the predict function in the advantage of smart contract code. The recommender model is used in smart contracts. The default gas limit for these models is approximately 9 M to deploy smart contracts. Additionally, for appending data, any user can examine the model for predictions. **Figure 3** demonstrates how our movie recommender models are deployed in the smart contract and further compiled into the Ethereum local node. Because it will be unable to process a massive amount of data in the Ethereum blockchain due to the limited gas available, all necessary features are extracted in Python. For example, a similar matrix is calculated for User-Based and item-Based collaborative filtering models. Additionally, the most popular model requires the number of watched times of the movies from users. Moreover, the random model excludes viewed movies from users. After all, models are trained and saved, and model files are exported as JSON files to be read from the smart contract.





At the client's side, the solidity compiler will compile the smart contract with the recommender model, and then the address of deployment will be recorded with the required gas cost. Finally, the prediction model is loaded into the Ethereum local network.

Some restrictions exist in the Ethereum blockchain, such as the cost of memory. This paper focuses on applications related to handling input that can be compressed easily, such as numbers or words. It will be enormously consumable for some deep neural networks which have complicated structures or models. Initialization of model deployment and a huge smart contract can result in massive gas costs, which lead to a possible denial of transaction in the nodes. According to the latest Ethereum, the limit of gas has approximately 8 million, which is not sufficient in our experiment. Therefore, we will change the limit of gas so that our models can be appended to smart contracts in many transactions once the contracts are deployed. In this paper, we have focused on using collaborative filtering techniques such as user-based, item-based, most popular, and random as commonly used recommender systems that suffer from attacks and privacy issues. Each of them will result in a different prediction of recommended movies for users. In Figure 4 shown below, users in our recommender model can watch and rate movies. The action of rating movies is recorded every time, so the system learns how to recommend correct movies to users by calculating the similarity matrix between users. Moreover, the watched movie cannot be rated by the same user repeatedly. Initially, users rate different movies (romantic, action, dramas, comedies, etc.). Then, the predictions will be automatically provided by the system regarding the user's rating for a specific movie that has not been rated yet.

**Figure 5** demonstrates the calculation of the recommendation score, where S is the symmetric similarity matrix between users based on the User-Based collaborative model, U is the rating of movies from users, and R is the recommender score. In our case, the values of the similarity matrix are decimal numbers. Normally, the value of U ranges between 1 - 5 and is represented in column vectors, except those users who never watch movies will be given as 0. R can be acquired if the multiplication is performed between S and U. For example, user #2 acquires the recommending score by selecting the row of user #2 in the similarity matrix and multiply each column of the rating in row 2 and add up the values. Before getting the real recommending score for user #2, originally watched movies by user #2 should be filtered since rated movies will not be viewed again. Therefore, the highest recommending score for user #2 is 7.13, which means that movie #10 is the best option for user #2, as in **Figure 6**.

# **5. Experimental Analysis**

In this section, we evaluated the performance of the proposed recommendation systems using two real datasets. The adopted recommendation models are user-based, item-based, popular-based, and random-based collaborative filtering.



Figure 4. An example of user's rating matrix.







Figure 5. Calculation of the recommendation score.



Figure 6. Examples of movie recommendations.

We have chosen different collaborative filtering models to observe which algorithm can acquire the best performance.

# **5.1. Experimental Datasets**

*Movielens dataset*: "Movielens 100K dataset" includes 100K movie ratings with 100,000 ratings ranging from 1 to 5 (integral) stars from 943 users on 1682 movies, and it was released in April 1998. Additionally, each user has rated at least 20 movies. The data source is available by the following link: <a href="https://grouplens.org/datasets/movielens/100k/">https://grouplens.org/datasets/movielens/100k/</a>.

**Netflix dataset:** The movie rating files contain over 100 million ratings from 480 thousand randomly chosen, anonymous Netflix customers over 17 thousand movie titles. The data was collected between October 1998 and December 2005 and reflect the distribution of all ratings received during this period. The ratings are on a scale from 1 to 5 (integral) stars. To protect customer privacy, each customer id has been replaced with a randomly assigned id. The date of each rating and the title and year of release for each movie id is also provided. The data source is available by the following link:

https://www.kaggle.com/netflix-inc/netflix-prize-data/data.

# 5.2. Gas Cost in Blockchain

The Gas cost is the pricing value or fee needed to proceed with a transaction further to execute the smart contract on the Ethereum blockchain. The sub-units of the cryptocurrency ETH are denoted as gwei, which can be converted to  $10^{-9}$ ether. The calculation of gas cost depends on the use of memory and the complexity of the algorithm. The blockchain's gas cost is a significant issue, due to the restriction of the gas limit in smart contracts on Ethereum blockchain, the dataset is filtered to 10 users and 200 movies only, resulting in 540 users for Movielens and 750 for the Netflix dataset. The figures below show the cost of gas when deploying the recommender model in smart contracts into the blockchain using different collaborative filtering algorithms. If using the original excessive amount of data, the smart contract system will result in crashes in Ethereum virtual machines. Therefore, for laboratory simulation, we have increased the gas limit from  $8.9 \times 10^6$  to  $8 \times 10^7$  due to the complex operation in smart contracts to be deployed properly into the Ethereum blockchain. According to recent votes from Ethereum miners, the total gas limit on the Ethereum blockchain has been agreed to increase by 25% (from  $1 \times 10^7$  to  $1.25 \times 10^7$ ) so that all transactions can be safely deployed.

# **5.3. Experimental Results**

In this section, we have conducted various experiments to evaluate the performance of the proposed system, including calculating the gas cost of each recommender algorithm and assessing the accuracy of each recommender model using multiple evaluation metrics.

#### 5.3.1. Gas Cost Calculations

As depicted in **Figures 7-9** below, results are shown with the filtered data; it can be observed that the gas needed to deploy the smart contracts is still extremely large. Also, by the observation, the User-Based CF algorithm has the highest gas cost due to the calculation of similarity matrices between users. As there is no calculation of the similarity matrix for the "most popular" and "random" recommendation algorithms, the gas required for the smart contract compilation is less than that of the User-Based CF algorithm. The random CF algorithm has the lowest gas cost due to no permutation needed to recommend the ideal movies to specific users.

Deploying 'Recommender	-
> transaction hash: 78c43b1eeeda	0x4bb0aad09b202c37ec4ec4fa9bf5e4bd4e1d3f82efef1d0591c6
> Blocks: 1	Seconds: 22
> contract address:	0x349CB2Fd1f5EB66e8B2b15E80C8944Dcc939F84d
<pre>&gt; block number:</pre>	5
> block timestamp:	1595258996
<pre>&gt; account:</pre>	0xa178e026c9fC37728584eE0Ef9DC5A2534097E5a
<pre>&gt; balance:</pre>	98.64167118
> gas used:	62776862
> gas price:	20 gwei
<pre>&gt; value sent:</pre>	0 ETH
<pre>&gt; total cost:</pre>	1.25553724 ETH

Figure 7. Gas cost (User-based CF algorithm).

> transaction hash:	0xfacc97683ac508c38d880c72e8beb459ad086ab6f63c8ee454bc
le2c88b63300	
> Blocks: 0	Seconds: 0
> contract address:	0x349CB2Fd1f5EB66e8B2b15E80C8944Dcc939F84d
<pre>&gt; block number:</pre>	5
> block timestamp:	1597327938
> account:	0xa178e026c9fC37728584eE0Ef9DC5A2534097E5a
<pre>&gt; balance:</pre>	99.64260976
> gas used:	12853250
> gas price:	20 gwei
<pre>&gt; value sent:</pre>	0 ETH
> total cost:	0.257065 ETH

Figure 8. Gas cost (Most popular CF algorithm).

Deploying 'Recommender	- 1 
> transaction hash: 94f132c99db6	0x1dd9c6404adf8a4bbebfe7c100d71c1e0d6c818cd99eefc6f01b
> Blocks: 0	Seconds: 0
> contract address:	0x349CB2Fd1f5EB66e8B2b15E80C8944Dcc939F84d
<pre>&gt; block number:</pre>	5
> block timestamp:	1597334243
> account:	0xa178e026c9fC37728584eE0Ef9DC5A2534097E5a
<pre>&gt; balance:</pre>	99.67200694
> gas used:	11383391
> gas price:	20 gwei
> value sent:	0 ETH
> total cost:	0.22766782 ETH
Deployed classifier to	0x349CB2Fd1f5EB66e8B2b15E80C8944Dcc939F84d.

**Figure 9.** Gas cost (Random CF algorithm).

# 5.3.2. Movie Recommendations

**Table 1** shows the results collected from the blockchain, which offers all users with their recommended movies, which have the top priority. Restrictions for the number of user id must be ranged from 1 - 10, and for the number of movie id must be ranged from 1 - 200. Also, the rating metric must be ranged from 1 - 5. It can be shown that user #1 has watched all the movies; thus, no recommendation is provided for user #1.

User id	Recommended Movie id	Recommended Movie Name
1	Х	None to recommend because all watched
2	174	(Raiders of the Lost Ark (1981) 01-Jan-1981)
3	174	(Raiders of the Lost Ark (1981) 01-Jan-1981)
4	127	(Godfather, The (1972) 01-Jan-1972)
5	127	(Godfather, The (1972) 01-Jan-1972)
6	144	(Die Hard (1988) 01-Jan-1988)
7	1	(Toy Story (1995) 01-Jan-1995)
8	100	(Fargo (1996) 14-Feb-1997)
9	127	(Godfather, The (1972) 01-Jan-1972)
10	89	(Blade Runner (1982) 01-Jan-1982)

**Table 1.** Recommendation for movie watchers.

**Figure 10** shows that training new data is available in blockchain by consuming gas as a transaction through Metamask. Gas cost is required every time if training new data. Moreover, the gas used in Metamask is bitcoin. Watched movies cannot be rated by the same user again. For example, **Table 2** shows the action of rating the movie " $174 \rightarrow$  (Raiders of the Lost Ark (1981) 01-Jan-1981)" with rating 4 to user 3. After rating movie #174, the system will recalculate the recommending score and provide the proper predication for user #3. In this case, the movie " $50 \rightarrow$  (Star wars (1997) 01-Jan-1977)" will be recommended to user #3, as shown in **Table 2**.

The same operation is performed if we are willing to train other new data. For example, **Table 3** shows appending a watched movie "1 (Toy Story (1995) 01-Jan-1995)" with rating 3 to user 7. After rated movie #1, the system will recalculate the recommending score and provide the proper predication user 7. In this case, movie "124  $\rightarrow$  (Lone Star (1996) 21-Jun-1996)" will be recommended to user 7. Additionally, it can be observed that the recommended movie for user 3 remained "50  $\rightarrow$  (Star Wars (1997) 01-Jan-1977)" as shown in **Table 3**, because the early trained data will not be changed unless we clean the database by entering "yarn clean" command in our blockchain and re-compile the smart contract.

#### 5.4. Complexity Analysis

For smart contracts, the complexity is due to the cost of gas. Three functions must be implemented: "constructor", "predict", and "update" to structure the smart contract and deploy it correctly. Each function in the smart contract will be analyzed next. The preliminary condition is set that the number of users is n and the number of movies to be m where m > n. The complexity for the constructor function is  $O(n^2m)$  as it is required to reconstruct the similarity matrix, which is used for the updating functions. The complexity of the prediction function is  $O(m^2n)$ , as it calculates the similarities between movies for each user. The complexity of the update function is O(m) as it scans the set of users only for updates, where *n* is the number of data records (*i.e.*, users), and *m* is the number of features (*i.e.*, movies).



Figure 10. Processing a new transaction through metamask with gas cost in the block-chain.

Table	2.	Recommen	dations	for	user	#3.
-------	----	----------	---------	-----	------	-----

User id	Recommended Movie id	Recommended Movie Name
1	Х	None to recommend as movies all already watched
2	174	(Raiders of the Lost Ark (1981) 01-Jan-1981)
3	50	(Star Wars (1977) 01-Jan-1977)
4	127	(Godfather, The (1972) 01-Jan-1972)
5	127	(Godfather, The (1972) 01-Jan-1972)
6	144	(Die Hard (1988) 01-Jan-1988)
7	1	(Toy Story (1995) 01-Jan-1995)
8	100	(Fargo (1996) 14-Feb-1997)
9	127	(Godfather, The (1972) 01-Jan-1972)
10	89	(Blade Runner (1982) 01-Jan-1982)

#### Table 3. Recommendations for user #7.

User id	Recommended Movie id	Recommended Movie Name
1	Х	None to recommend because all watched
2	174	(Raiders of the Lost Ark (1981) 01-Jan-1981)
3	50	(Star Wars (1977) 01-Jan-1977)
4	127	(Godfather, The (1972) 01-Jan-1972)
5	127	(Godfather, The (1972) 01-Jan-1972)
6	144	(Die Hard (1988) 01-Jan-1988)
7	124	(Lone Star (1996) 21-Jun-1996)
8	100	(Fargo (1996) 14-Feb-1997)
9	127	(Godfather, The (1972) 01-Jan-1972)
10	89	(Blade Runner (1982) 01-Jan-1982)

#### 5.5. Performance Analysis

Figures 11-15 demonstrate the results of testing the performance of the adopted algorithms. The x-axis reveals the percentage of original data as testing, and the y-axis is one of the five metrics used for performance evaluation. We used bar charts to represent precision, recall, F-measure, TPR, and FPR for the *Movie-lens* and *Netflix data*. Figure 11 shows that the User-Based CF, item-Based CF and Most popular CF outperform the Random CF measures by the high precision values. We can also observe that both datasets have the same trend, in which the precision values increase as the training sets increase, but then it decreases when it gets closer to 80% - 90% of the original data. The reason is that our training models are getting under fitted so that the recommendation will not be accurate as expected. In Figure 12, we can observe that the recall values for all the models have the similar trend of deceasing due to the reduction in the training data except for random model. The recall is defined as the ratio of hit count

the number of test set

The reason of the declining trend in both graphs is because the test set is getting larger while the hit count remains unchanged compared to precision method. Using the values from precision and recall will not be sufficient. Although we expect both values of precision and recall can be as high as possible, sometimes they are repulsive which indicates one has a higher value and one has a lower value. Therefore, F-measure =  $\frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$  can observe this re-

pulsive scenario.

**Figure 13** shows that all models reached the best performance when the test set is around 0.2 and 0.3 of the original data for the F-measure. **Figure 14** illustrates that the TPR has a similar trend as the recall. For both graphs, TPRs are the highest when the test set is around 0.1 and 0.2 of the original data. On the contrary, we expect FPR to be as low as possible. **Figure 15** shows that the random model has the highest FPR, no matter how the test set is changing. For other models, all the FPR are small, less than 0.6% for Movielens, and 1.5% for Netflix, respectively.



Figure 11. Precision values for the four recommendation algorithms.



Figure 12. Recall values for the four recommendation algorithms.

















Figure 15. FPR values for the four recommendation algorithms.

# 6. Conclusion and Future Directions

The blockchain platform provides a traceable and transparent platform and thus enables trust-free transactions. We have successfully implemented and combined a movie recommendation system based on our laboratory Ethereum blockchain in this paper. Additionally, the smart contracts are successfully deployed onto the blockchain so that the correct movies are recommended to users properly through the blockchain. A big challenge is that it cannot be performed on the main Ethereum network with the limitation of the gas limit. Our movie recommender model operates properly and provides the correct prediction for the movie's watcher in the blockchain. Also, by the adoption of the Metamask, training data is enabled through the digital transaction. According to the models, the blockchain calculates and predicts results by adopting different models into smart contracts. In the future, the proposed framework can be loaded by different types of models; the total amount of data can be reduced by simplifying the data shapes. A reduction algorithm is required to be considered to reduce the similarity matrix. Also, we are interested in investigating the applications of other smart contract platforms and various recommendation algorithms.

# **Conflicts of Interest**

The authors declare no conflicts of interest regarding the publication of this paper.

# References

- Syed, T.A., Alzahrani, A., Jan, S., Siddiqui, M.S., Nadeem, A. and Alghamdi, T. (2019) A Comparative Analysis of Blockchain Architecture and Its Applications: Problems and Recommendations. *IEEE Access*, 7, 176838-176869. <u>https://doi.org/10.1109/ACCESS.2019.2957660</u>
- [2] Frey, R.M., Vuckovac, D. and Ilic, A. (2016) A Secure Shopping Experience Based on Blockchain and Beacon Technology. In RecSys Posters.
- [3] Xue Tan and Kashef, R. (2019) Predicting the Closing Price of Cryptocurrencies: A Comparative Study. *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems (DATA* '19), New York, December 2019, 1-5. <u>https://doi.org/10.1145/3368691.3368728</u>
- [4] Lisi, A., De Salve, A., Mori, P. and Ricci, L. (2019) A Smart Contract Based Recommender System. In: Djemame, K., Altmann, J., Bañares, J., Agmon Ben-Yehuda, O. and Naldi, M., Eds., *Economics of Grids, Clouds, Systems, and Services, GECON* 2019, *Lecture Notes in Computer Science*, Springer, Cham, 29-42. https://doi.org/10.1007/978-3-030-36027-6\_3
- [5] Fallis, A. (2013) Rootstock Platform: Bitcoin Powered Smart Contracts—White Paper. *Journal of Chemical Information and Modeling*, 53, 1689-1699. <u>https://doi.org/10.1021/ci400128m</u>
- [6] Luu, L., Chu, D.-H., Olickel, H., Saxena, P. and Hobor, A. (2016) Making Smart Contracts Smarter. *Proceedings of the* 2016 ACM SIGSAC Conference on Computer and Communications Security, 24-28 October 2016, 254-269. https://doi.org/10.1145/2976749.2978309
- [7] Ricci, F., Rokach, L. and Shapira, B. (2011) Introduction to Recommender Systems Handbook. Springer, Berlin. <u>https://doi.org/10.1007/978-0-387-85820-3</u>
- [8] Sree Lakshmi, S. and Adi Lakshmi, T. (2014) Recommendation Systems: Issues and Challenges. *International Journal of Computer Science and Information Technologies*, 5, 5771-5772.

- [9] Moreno, M.N., Segrera, S., Lopez, V.F., Muñoz, M.D. (2015) Web Mining Based Framework for Solving Usual Problems in Recommender Systems. A Case Study for Movies' Recommendation. Elsevier, Amsterdam. <u>https://doi.org/10.1016/j.neucom.2014.10.097</u>
- [10] Lu, J., Wu, D.S., Mao, M.S., Wang, W. and Zhang, G.Q. (2015) Recommender System Application Developments: A Survey. Decision Support Systems, 74, 12-32. <u>https://www.sciencedirect.com/science/article/pii/S0167923615000627</u> <u>https://doi.org/10.1016/j.dss.2015.03.008</u>
- [11] Alhijawi, B., Kilani, Y. and Alsarhan, A. (2020) Improving Recommendation Quality and Performance of Genetic-Based Recommender System. *International Journal* of Advanced Intelligence Paradigms, 15, 77-88. https://doi.org/10.1504/IJAIP.2020.104108
- [12] Nilashi, M., bin Ibrahim, O., Ithnin, N. and Sarmin, N.H. (2015) A Multi-Criteria Collaborative Filtering Recommender System for the Tourism Domain Using Expectation Maximization (EM) and PCA-ANFIS. *Electronic Commerce Research and Applications*, 14, 542-562. <u>https://www.sciencedirect.com/science/article/pii/S1567422315000599</u> <u>https://doi.org/10.1016/j.elerap.2015.08.004</u>
- [13] Levinas, C.A. (2014) An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals. MS Thesis, Universitat Politècnica de Catalunya, Barcelona.
- [14] Mansur, F., Patel, V. and Patel, M. (2017) A Review on Recommender Systems. 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 17-18 March 2017, 1-6. https://doi.org/10.1109/ICIIECS.2017.8276182
- [15] Guimarães, R., Rodríguez, D.Z., Rosa, R.L. and Bressan, G. (2016) Recommendation System Using Sentiment Analysis Considering the Polarity of the Adverb. 2016 *IEEE International Symposium on Consumer Electronics* (ISCE), Sao Paulo, 28-30 September 2016, 71-72. <u>https://ieeexplore.ieee.org/abstract/document/7797377/</u> <u>https://doi.org/10.1109/ISCE.2016.7797377</u>
- [16] Chen, J.R., Zhao, C.X., Uliji and Chen, L.F. (2019) Collaborative Filtering Recommendation Algorithm Based on User Correlation and Evolutionary Clustering. *Complex & Intelligent Systems*, 6, 147-156. <u>https://doi.org/10.1007/s40747-019-00123-5</u>
- [17] Jain, A., Jain, V. and Kapoor, N. (2016) A Literature Survey on the Recommendation System Based on the Sentimental Analysis. Advanced Computational Intelligence, 3, 25-36. <u>https://www.academia.edu/download/42598847/3116acii03.pdf</u> https://doi.org/10.5121/acii.2016.3103
- [18] Aggarwal, C. (2016) Model-Based Collaborative Filtering. In: *Recommender Systems*, Springer, Cham, 71-138.
  <u>https://link.springer.com/chapter/10.1007/978-3-319-29659-3\_3</u>
  <u>https://doi.org/10.1007/978-3-319-29659-3\_3</u>
- [19] Kashef, R. and Warraich, M. (2020) Homogeneous vs. Heterogeneous Distributed Data Clustering: A Taxonomy. In: Alhajj, R., Moshirpour, M. and Far, B., Eds., *Data Management and Analysis. Studies in Big Data*, Springer, Cham, 51-66. <u>https://doi.org/10.1007/978-3-030-32587-9\_4</u>
- [20] Yang, S., Korayem, M., AlJadda, K., Grainger, T. and Natarajan, S. (2017) Combining Content-Based and Collaborative Filtering for Job Recommendation System: A Cost-Sensitive Statistical Relational Learning Approach. *Knowledge-Based Systems*, 136, 37-45. <u>https://www.sciencedirect.com/science/article/pii/S095070511730374X</u> <u>https://doi.org/10.1016/j.knosys.2017.08.017</u>

- [21] Sulthana, A.R. and Ramasamy, S. (2019) Ontology and Context-Based Recommendation System Using Neuro-Fuzzy Classification. *Computers & Electrical Engineering*, 74, 498-510. <u>https://www.sciencedirect.com/science/article/pii/S0045790617337382</u> <u>https://doi.org/10.1016/j.compeleceng.2018.01.034</u>
- [22] Zhang, H.-R., Min, F., He, X. and Xu, Y.-Y. (2015) A Hybrid Recommender System Based on User-Recommender Interaction. *Mathematical Problems in Engineering*, 2015, Article ID: 145636. <u>https://doi.org/10.1155/2015/145636</u>
- [23] Maesa, D.D.F., Mori, P. and Ricci, L. (2019) A Blockchain Based Approach for the Definition of Auditable Access Control Systems. *Computers & Security*, 84, 93-119. <u>https://doi.org/10.1016/j.cose.2019.03.016</u>
- [24] Min, H. (2019) Blockchain Technology for Enhancing Supply Chain Resilience. Business Horizons, 62, 35-45.
   <u>https://www.sciencedirect.com/science/article/pii/S0007681318301472</u>
   <u>https://doi.org/10.1016/j.bushor.2018.08.012</u>
- [25] Kashef, R. and Niranjan, A. (2017) Handling Large-Scale Data Using Two-Tier Hierarchical Super-Peer P2P Network. *Proceedings of the International Conference* on *Big Data and Internet of Thing*, New York, 20-22 December 2017, 52-56. https://doi.org/10.1145/3175684.3175726
- [26] Fanning, K. and Centers, D. (2016) Blockchain and Its Coming Impact on Financial Services. *Journal of Corporate Accounting & Finance*, 27, 53-57. <u>https://doi.org/10.1002/jcaf.22179</u>
- [27] Harris, J.D. and Waggoner, B. (2019) Decentralized & Collaborative AI on Blockchain. 2019 *IEEE International Conference on Blockchain (Blockchain)*, Atlanta, 14-17 July 2019, 368-375. <u>https://doi.org/10.1109/Blockchain.2019.00057</u>
- [28] Ibrahim, A., Kashef, R., Li, M., Valencia, E. and Huang, E. (2020) Bitcoin Network Mechanics: Forecasting the BTC Closing Price Using Vector Auto-Regression Models Based on Endogenous and Exogenous Feature Variables. *Journal of Risk and Financial Management*, **13**, 189. https://doi.org/10.3390/jrfm13090189
- [29] Behnke, K. and Janssen, M.F.W.H.A. (2020) Boundary Conditions for Traceability in Food Supply Chains Using Blockchain Technology. *International Journal of Information Management*, **52**, Article ID: 101969. <u>https://www.sciencedirect.com/science/article/pii/S0268401219303536</u> <u>https://doi.org/10.1016/j.ijinfomgt.2019.05.025</u>
- [30] Corbet, S., Larkin, C., Lucey, B., Meegan, A. and Yarovaya, L. (2020) Cryptocurrency Reaction to FOMC Announcements: Evidence of Heterogeneity Based on Blockchain Stack Position. *Journal of Financial Stability*, **46**, Article ID: 100706. <u>https://doi.org/10.1016/j.jfs.2019.100706</u>
- [31] Kashef, R. and Kamel, M.S. (2008) Distributed Peer-to-Peer Cooperative Partitional-Divisive Clustering for gene expression datasets. 2008 *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Sun Valley, 15-17 September 2008, 143-150. <u>https://doi.org/10.1109/CIBCB.2008.4675771</u>
- [32] Crosby, M., Pattanayak, P., Verma, S. and Kalyanaraman, V. (2016) Blockchain Technology: Beyond Bit Coin. *Applied Innovation*, No. 2, 6-19. https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf
- [33] Shen, C. and Pena-Mora, F. (2018) Blockchain for Cities—A Systematic Literature Reviews. *IEEE Access*, 6, 76787-76819. <u>https://ieeexplore.ieee.org/abstract/document/8531608/</u> <u>https://doi.org/10.1109/ACCESS.2018.2880744</u>