Computer Communications 187 (2022) 1–19

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A survey on blockchain-based Recommender Systems: Integration architecture and taxonomy

Loubna Mekouar [a,*], Youssef Iraqi [a], Issam Damaj [b], Tarek Naous [c]

[a] School of Computer Science, Mohammed VI Polytechnic University, Ben Guerir, Morocco
[b] Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff, United Kingdom
[c] Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon

ARTICLE INFO

Keywords:
Recommender Systems
Blockchain
Smart contracts
Oracles
Distributed ledgers
Security

ABSTRACT

A Recommender System (RS) is an integral part of present-day leading web services, such as YouTube, Amazon, Netflix, and many others. Modern RSs are challenged to go beyond their traditional role of predicting user preferences to efficiently provide reliable, carefully personalized, and highly accurate recommendations. This paper thoroughly explores and analyzes state-of-the-art literature surveys on RS to extract important challenges and open issues. Our goal in this paper is to survey the literature to extract essential features of RSs and Blockchain (BC), focusing on their integration. Because of the lack of an existing foundation of BC-based RSs, the intrinsic BC aspects in RSs are identified and described. Integrating BC technology within RSs can achieve many benefits such as transparency, decentralization, and security. To that end, a thorough study of the papers on current BC-based RSs is presented along with a synthesized comprehensive taxonomy. Furthermore, a modular RS architecture, encompassing on-chain and off-chain storage and computation processes, is designed. This paper also includes a thorough discussion on the validity of the proposed architecture, BC limitations concerning RSs, and the derivation of a rich set of pointers to future research directions.

1. Introduction

Modern Recommender Systems (RSs) are used in various present-day industries, such as e-commerce, retail, media, to name but a few. Businesses using RSs focus on creating a personalized user journey and an enhanced experience, which results in increased customer retention and sales. Businesses also use recommendations to speed up searches and make it easier for users to access content relevant to them. Furthermore, it allows companies to position ahead of their competitors and eventually increase their earnings. Without a doubt, providing such an added value to users by including recommendations is appealing. RSs are one of the most used and mature use cases of data science and artificial intelligence. To date, researchers have developed many RSs techniques in the pursuit of maximizing the effectiveness of these systems in various industrial domains [1]. However, these developers are facing many challenges, mainly related to performance, security, and privacy [2,3].

An RS seeks to estimate and predict user content preference and interest in specific items. The system draws from data usage history, aiming at making suggestions based on the user's interest. Most techniques behind RSs employ data sources to learn more about user preferences, making good use of explicit or implicit feedback. The more feedback RSs can collect, the more accurate the prediction can be. However, as RSs collect large amounts of data and user information, they are prone to security and privacy issues.

For example, traditional RSs have to address the problems of privacy, trust, risk of data loss, risk of data tampering, and centralized related issues like single point of failure and scalability. Traditional RSs can address these issues directly, however some new technologies lend themselves as a convenient approach to address these issues holistically.

Modern RS architectures attempt to benefit from trending technologies like Artificial Intelligence (AI), Machine Learning (ML), and cybersecurity. One such important technology is Distributed Ledger Technology (DLT). A DLT implements a ledger in which a network of decentralized nodes can store data. The goal is to create a tamper-proof ledger and to distribute its control among all participants. In essence, DLTs are data structures where participants can write transactions and functions to manipulate them. Although each DLT may be using different data models and technologies, all DLTs use three well-known pillars, (i) public-key cryptography, (ii) distributed Peer-to-Peer (P2P) networks, and (iii) consensus mechanisms [4]. The most famous implementation of DLT is Blockchain (BC).

The decentralization feature, on top of the cryptography, makes BC provide better protection than other systems. Other than security,

* Corresponding author.
E-mail address: Loubna.Mekouar@um6p.ma (L. Mekouar).

https://doi.org/10.1016/j.comcom.2022.01.020
Received 17 September 2021; Received in revised form 21 December 2021; Accepted 29 January 2022
Available online 7 February 2022

BC has various benefits making it gain attraction from researchers as it offers an immutable, transparent, and reliable data storage structure [5]. Offering this added value, BC is becoming the go-to technology in multiple sectors, like banking, healthcare, and the supply chain.

One promising technology advancement to explore is integrating BC technology within RSs. Indeed, BC can be incorporated to assure cryptographic safety. BC can be used in conjunction with legacy techniques of RSs to provide solutions that guarantee user's privacy and introduces the possibility to perform computations without disclosing the input data [6]. The integration will result in the prevention of misuse of personal data and fraud. Many benefits are expected from the introduction of BC within RSs. For example, the problems presented earlier can be addressed by the inherent features of the BC. Indeed, the problems of privacy, trust, risk of data loss, risk of data tampering, and centralized related issues can be addressed respectively by the following BC features: encryption, transparency, persistency, tamper-proofing, and distribution.

However, challenges are also expected in terms of effective architectural integration, implementation, and performance. Integrating BC improvements to RSs will motivate people to be less conservative in sharing their information, boosting the RS benefits across many sectors.

The research objectives of this article include the following:

- Survey the literature to extract important features of RSs and BC technology with a focus on their potential integration.
- Create a taxonomy for BC-based RSs.
- Develop an architecture that captures different RS procedures and their integration within the BC.
- Provide recommendations on the developed taxonomy and the performed architecture validation.
- Discuss various implementation aspects related to BC-based RSs.
- Identify open research problems.
- Identify area transformations and future trends.
- Provide a comprehensive bibliography on BC-based RSs.

The adopted methodology comprises the following:

- Develop the search protocol.
- Search in recent outstanding and comprehensive surveys of the literature.
- Develop classifications of related work.
- Develop the taxonomy.
- Present a discussion on good practices, challenges, and limitations in the surveyed literature.
- Identify patterns in existing BC-based RSs.
- Develop a generic architecture for BC-based RSs.
- Present recommendations on various important aspects.
- Identify and propose future work.
- Conclude with open research questions.

A systematic protocol was adopted in search of the articles to be surveyed. The academic search engines used were *IEEE Xplore digital library*, *ScienceDirect*, and *ACM Digital Library*. Several combinations of the following keywords were used in the search process: recommender systems, decentralized RSs, privacy-preserving RS, service recommendation, machine learning and deep learning RSs, RS algorithms, recommendation models, blockchain, smart contracts, hyperledger, distributed ledger technology, oracles, security and applications, such as Internet of Things (IoT), smart cities, health informatics, connected vehicles, autonomous driving, financial transactions, and trade. To maintain the currency of the surveyed publications, as part of this investigation, the dates were restricted to be between 2016 and 2021. The search related to RSs comprised only survey articles, while the search for investigations that combines BC and RSs included research and survey articles. The initial set included 36 surveys and 32 research articles. Then, the initial set was systematically reduced to 25 surveys and 18 research articles by removing out-of-scope papers. All papers in the reduced set included significant information as mapped onto the protocol keywords.

A variety of survey papers that explore different aspects of RSs were identified in the literature. In [7,8], the authors explored applications, challenges, classifications, open issues, and future opportunities of BC-based RSs. In [7], the focus was on investigating privacy solutions using BC for RSs. Furthermore, a holistic review with emphasis on security and privacy challenges was presented in [8]. In [9–12], RSs were discussed specifically in social networks, and multiple techniques were introduced. Other survey papers also investigated specific domains, like e-commerce in [2], mobile application in [13], scientific paper research in [1], smart cities in [14], cloud services in [15], and music recommendation in [16]. However, other survey papers focused on the techniques used regardless of the domain or use-case of the system. In [17,18], RSs based on deep learning were discussed, while in [3] the topic was RSs and edge computing. Various other techniques or architectures were investigated as well, like multi-criteria review-based RSs in [19], evaluation of Collaborative Filtering (CF) in [20], parallel and distributed CF in [21], sequence-aware RSs in [22], and context-aware RSs in [23]. A systematic review of the use of ML algorithms in RSs was presented in [24].

This article is organized so that Section 2 presents background information on RSs and the BC. In Section 3, modern RSs are surveyed for intrinsic aspects and open issues that may benefit from an integration with the BC. To motivate the importance of BC-based RS, Section 4 presents a use-case scenario. Section 5 surveys BC-based RSs and presents the synthesized taxonomy. A modular architecture for BC-based RSs is proposed in Section 6. Section 7 presents a thorough discussion that comprises validating the developed architecture, identifying limitations of smart contracts in the context of RSs, and presenting area transformation and future trends. Section 8 concludes the paper.

2. Background

2.1. Recommender systems

With the vast flow of information on the web, users seeking certain services or products can be faced with a large number of options to select from; rendering the decision-making process a complex one. Information filtering is an important process that alleviates this problem and assists users in making satisfying decisions with ease. This filtering procedure is handled by RSs that have become an integral part of a large variety of industrial applications [25]. Based on user preferences, RSs provide personalized suggestions, improving the overall user experience. To this end, RSs are tasked with providing either predictions or recommendations [19]. When tasked with prediction, an RS determines the likelihood of a user liking a particular item by predicting the rating with which the user will rate this item. When tasked with providing recommendations, an RS generates a list of items that the user will most probably like. Recommendations generated by an RS can include lists of movies [26], articles [27], songs [28], and much more [29,30].

To generate recommendations, RSs rely on the collection of user data from which insight to user preferences on items can be deduced. The data collected may be based on implicit interactions of the users, such as their historical behavior (percentage scrolled of a webpage, browsing history, etc.), or explicit interactions such as user-submitted ratings. The large data quantities collected are uploaded to a server where data pre-processing and representation are performed. The pre-processed data representation is then used as an input to a recommendation algorithm. Computations are then done by the algorithm and the resulting recommendations are sent back to the users. Therefore, the general framework of an RS consists of: data collection and centralized storage, data pre-processing and representation, computation by a certain algorithm, and finally recommendation [9, 31].

Depending on how the RS algorithms generate recommendations, they can be categorized into various types. For instance, Content-Based (CB) recommendation algorithms focus on finding similarities among items based on their content [32]. Given the interaction of a user with a specific number of items, a user profile can be established and used to identify their preferences. By knowing the preferences of each user, items with similar features or content to those positively rated by the users can be then recommended to them. Additional data in the user profile can be exploited by CB algorithms such as user demographic information, age, education, gender, or nationality.

A different and popular approach is CF algorithms, where an assumption is taken that users with similar ratings to specific items have similar preferences and that users who have agreed in the past will also agree in the future [12,33]. As opposed to CB algorithms, CF approaches use information about similarities with other users. Therefore, CF algorithms rely only on item ratings submitted by users of the recommender to compute user similarities; hence the term "collaborative" refers to the need for collaboration of the users to filter information and generate recommendations.

While CB and CF approaches have been widely adopted, each approach has its advantages and drawbacks. To mitigate these weaknesses, hybrid recommendation techniques integrate two or more algorithms to make use of their complementary strengths and gain improvement in performance over a single recommendation technique. Many strategies to develop hybrid recommendation algorithms have been studied in the literature, such as combining the results of a CB RS and CF RS, boosting CF algorithms with CB features or vice-versa, and others [20,34]. Although hybridization procedures may provide performance enhancement, they could come at the expense of increased system complexity [35].

There exist several other types of RSs such as knowledge-based, knowledge graph-based, and context-aware RSs. Knowledge-based recommendation relies on knowledge of how specific items may match specific needs of users and can then search through their databases of items to find and recommend items that can satisfy user needs. This approach is mainly desirable over CB or CF approaches when no sufficient ratings for specific items are available, which is the case for items such as cars, houses, or financial services [36]. Hence, knowledge-based recommenders may prompt users for information of what the result should look like and then reason about the relationship between items and the need of users. A more recent approach is knowledge graph-based recommenders that rely on the extraction of the embedding of a knowledge graph; a heterogeneous graph of nodes, where each node is an entity and the edges between nodes represent their relationships. This helps capture user–user, item–item, or user–item relationships and is useful for recommendation processes [37]. Another type of RSs is context-aware recommenders. As opposed to the aforementioned types of recommendation algorithms that only consider user or item information, context-aware RSs also incorporate contextual information into their process, which can impact the relevance of recommendations [38,39]. Contextual information may be formed of several elements such as the physical and emotional state of users, locations, time, and more [40].

2.2. The blockchain

BC technology utilizes a peer-to-peer network to establish a distributed ledger of blocks that store data in a synchronized and verifiable way across the network. This technology allows the creation of a secure and **tamper-proof** log of transactions that are recorded in chronological order. While it was originally conceived as the enabling technology for Bitcoin [41], a trustless cryptocurrency, BC has since then unbound itself from this original purpose, finding its way as a promising solution to various industrial problems beyond cryptocurrencies. For instance, since BC allows payments to go through without any central intermediary, such as banks, it could be used to enable several financial services [42]. Due to its set of appealing characteristics, BC technology is expected to receive wide adoption in healthcare [43], supply-chain management [44], and many more industrial applications where the sharing of data among several parties is required [45–47].

Being a distributed architecture in nature that is maintained by several nodes, BC achieves **decentralization**. It was introduced as a decentralized data and transaction management technology that does not require the trust of stakeholders and where no central third party is in control of the data. Several nodes which in principle do not trust each other usually make up a BC system. The transfer of digital assets between nodes on the BC is recorded by a data structure referred to as a transaction. Each transaction records the number of inputs, outputs, and addresses of the assignments. These transactions are then stored in another data structure referred to as a block. Each block contains a record of transactions, as well as a header containing block-specific information [48].

While centralized systems require the trust of a central authority that has access and control over the data, BC guarantees the acquisition of trust through consensus mechanisms. These mechanisms ensure the state of the ledger is updated with the agreement, or consensus, of all participating nodes, and that transactions and blocks are validated before being added to the network. The performance of the BC is reliant on the performance of the adopted consensus strategy of which many exist in the literature, each having different characteristics in terms of speed, scalability, data consistency, and robustness [49]. The updating of the ledger through these mechanisms requires rigorous computations, a process commonly known as mining. The consensus allows maintaining the decentralized design of the BC, and eliminating the need to trust any central intermediary.

Two other exceptional characteristics of BC technology are its immutability and irreversibility properties. In a BC, the blocks are chained together using a cryptographic hashing function. Specifically, the header of each block contains the hash of the previous block. Linking the blocks together using cryptographic pointers makes the BC tamper-proof since any changes to the previous block will require changes to the block's header. Therefore, any attempt in tampering with the data stored in a block will require tampering with all connected blocks that come after it [50]. Additionally, given that blocks and transactions are always being validated by miners, any attempt in tampering with the data on the BC would be detected by the nodes. Since each node has its copy of the ledger, an adversary would have to perform the same modification to half of the copies in the network for it to be validated by the miners. However, modifying all these blocks is an almost infeasible task as it requires a huge amount of resources. This highlights the **persistence** feature of BC networks. Another important feature of BC is the concept of Smart Contract (SC). Smart contracts offer a lot of features and flexibility to execute decentralized logic. A SC is a deterministic program that includes an arbitrary executable script and a data model that is saved in the BC [4]. Through exposing public functions, a smart contract interacts with the users of the BC to offer predefined business logic.

Given the openness and **transparency** of the BC, where the complete ledger can be seen by every node of the network, privacy leakage could be of concern. To maintain **privacy** and protect the assets of users, cryptography is at the heart of BC. Specifically, BC relies on the concept of asymmetric encryption, also known as public-key cryptography, to enable its secure operation. This type of cryptography allows users to verify their identities to the whole network without disclosing who they are. This is feasible through the provision of a public and a private key to the user. Each user in the network possesses a digital wallet that is secured using the user's private key and is accessible using digital signatures generated by that private key. Each transaction is signed using the user's private key that is secretly kept by the user. The public key serves as the address of the user. Through the public key hash, they can be uniquely identified in the network, enabling an **authenticity** feature [51]. Also, the public key is changed with each

Table 1
The coverage of target information by the explored survey articles.

Reference	Year	Techniques	Architectures	System Properties	Performance Indicators	Implementations	Open Issues
[8]	2022	✓	✓	✓	✓	✓	✓
[7]	2021	✓	✓	✓			✓
[55]	2021	✓	✓	✓	✓	✓	✓
[56]	2021	✓			✓		✓
[2]	2020	✓		✓	✓	✓	
[3]	2020	✓	✓		✓	✓	✓
[14]	2020	✓			✓		✓
[16]	2020	✓	✓		✓	✓	✓
[23]	2019	✓		✓	✓	✓	✓
[1]	2019	✓		✓	✓		✓
[18]	2019	✓		✓	✓	✓	✓
[19]	2019	✓	✓	✓	✓		✓
[9]	2018	✓		✓	✓	✓	✓
[17]	2018	✓		✓			✓
[22]	2018	✓			✓	✓	✓
[11]	2018	✓		✓	✓	✓	✓
[20]	2018	✓			✓	✓	
[24]	2018	✓		✓	✓	✓	✓
[15]	2017	✓			✓		✓
[10]	2017	✓			✓	✓	✓
[13]	2016	✓	✓		✓	✓	✓
[57]	2016	✓	✓		✓	✓	
[21]	2016	✓			✓	✓	✓

transaction to preserve user **anonymity** and privacy. Hence, users can be known to the network through their public keys but can transact anonymously using their private keys [52].

BC systems can be roughly classified into three types: public, private, or consortium, based on the application scenario they are used in [53,54]. Each type has distinct characteristics that are tailored for the requirements of specific applications. Public BCs are completely decentralized and *permissionless*, meaning anyone is allowed to join the network and can participate in adding new blocks or accessing current blocks, reading and writing transactions, and participating in the consensus mechanism. Public BCs can be used to build various kinds of large-scale decentralized applications and are adopted by most cryptocurrencies. By contrast, private BCs are controlled by a central authority and are *permissioned*, whereby only identified and trusted participants of a single organization are allowed to join the network. The private type of BC network is specifically attractive for usage as a single-enterprise solution in organizations such as banks, where control over the data is desired. Consortium BCs are popular among industrial communities and are suitable to record cross-organizational transactions, helping maintain transparency. Like private BCs, consortium BCs are permissioned and only allow authorized organizations to join, but are partially decentralized instead of being controlled by a central entity.

3. Intrinsic blockchain aspects in modern recommender systems

To identify intrinsic BC aspects in RSs, a careful review of the selected survey articles is carried out with a focus on RS techniques (collaborative-based, knowledge-based, etc.), architectures (main building blocks), system properties (security, transparency, reliability, etc.), performance indicators, implementation aspects (datasets, tools, etc.), and open issues that are yet to be further investigated. The coverage of the search targets by the explored survey articles is shown in Table 1. The findings of the exploratory survey and the identified intrinsic BC aspects in modern RSs are presented in this section.

3.1. Modern recommender systems

An RS is currently an indispensable part of intelligent mass online services that include a large number of users and items. Modern large-sized service providers, such as Facebook, YouTube, and Amazon, rely heavily on the capabilities of their RSs in the success of their businesses. Throughout the advancement in RSs, the techniques used to accurately and efficiently determine the recommendations have significantly evolved. Throughout the explored literature, the following techniques were heavily mentioned [1,3,9,10,13–15,17,19]:

- Content-Based RS
- Collaborative Filtering
- Demographic-Based Filtering (DBF)
- Knowledge-Based Filtering (KBF)
- Hybrid mechanisms

Classification-based, Matrix Factorization (MF), social network-based, context-based, multi-criteria, review-based, graph-based, tag-based, location-aware, sequence-aware techniques were also presented in the literature [1,10,18,22].

Modern RS techniques are exhibiting appealing features and seem to have accumulated considerable strength over the years. On the system level, recommendation techniques demonstrate improved accuracy, scalability, security, Quality of Service (QoS), Quality of Experience (QoE), responsiveness, and decreased operation cost [15]. An example of improved accuracy and QoE, is context-aware recommendation techniques that incorporate contextual data, thus, coming up with a more accurate recommendation for a user's situation at a specific time [17]. RSs still face many limitations, and challenges regarding their techniques, which will be discussed later in Section 3.3.

One important aspect, that was studied throughout the explored articles, is modern RSs architectures. Investigations revealed a focus on exploring what components these architectures consist of, what are the procedures that precede the recommendation architecture flow to prepare data, and more. The explored surveys are almost in consensus that data collection and preprocessing are two essential preparatory steps before implementing an RS technique [13]. In the data collection step, users' and items' data are gathered. In the preprocessing step, data is cleaned and prepared for the later processing stages within the RS. Moreover, the preprocessing step ensures that input data is in an appropriate format and enables the generation of accurate recommendations. Preprocessing data by removing outliers aids in increasing the reliability of the system. Examples of data preprocessing for CF are the generation of implicit ratings and data integration [13].

For the most part, the architecture of an RS is highly dependent on the domain the system is being used in. For example, in [16], the researchers studied how to generate mobile music recommendations for runners based on their location, the architecture included multiple essential hardware devices to collect and analyze data. In another example, some RSs can collect data from users' surroundings like location, time, and climate and base the recommendation upon the perceived environment to generate personalized recommendations [14].

Fig. 1 presents an overview of the recommendation process in modern RSs and highlights some of their important aspects. The figure attempts to confirm the fact that RSs are critical and integral subsystems of mainstream online service providers, such as Netflix, Amazon, and YouTube whose number of users and items are in billions. As a black box, a modern RS communicates with users to input queries, ratings, tracked behavior, human and social attributes, and possibly their QoE. Furthermore, the system outputs one or more recommendations. The return of an RS is characterized by its attained performance as featured by a spectrum of indicators that include reliability, recall, precision, and accuracy. A typical RS allocates significant storage for its user profiles and items. Processes within an RS may comprise content analysis, filtering, similarity computations, profile

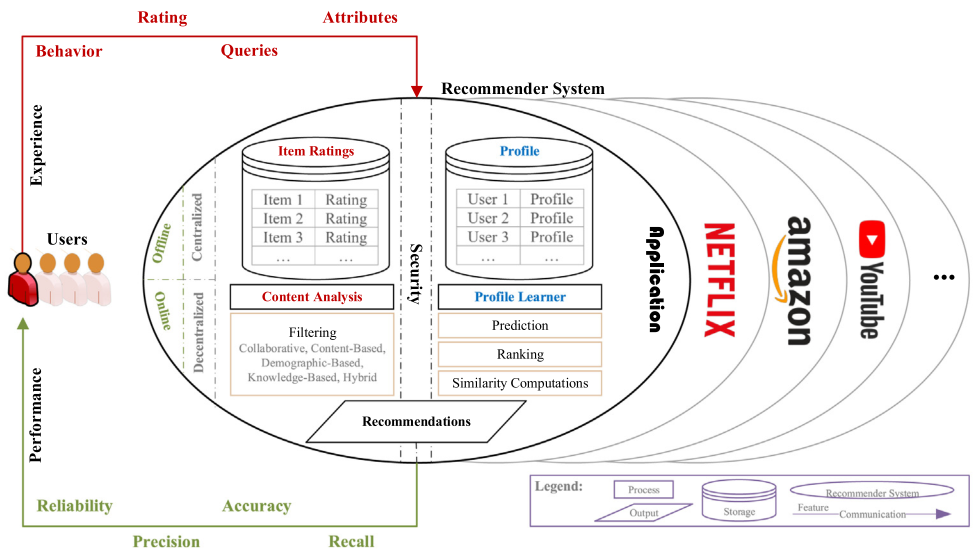

Fig. 1. An Overview of the recommendation process in modern RSs.

learning, prediction, ranking, and finally generating recommendations. Without a doubt, security plays a major role in RSs as related to privacy, confidentiality, availability, authentication, and more. The figure presents additional RSs characteristics that comprise online/offline and decentralized/centralized processing options.

The assessment and evaluation of an RS can be performed in two ways, namely offline or online. Offline analysis, a popular and relatively cheap approach, is conducted on standard datasets that might consist of large amounts of real data from previously deployed systems. In cases where no datasets have been published, artificially generated synthetic datasets are used. Online analysis is followed when an RS is deployed in a live system. To this end, tests are conducted with a focus on assessing the user behavior in reaction to different recommendations. Online analysis is important to evaluate metrics related to users, vendors, or technical aspects of the system. A/B testing is one of the most commonly used online evaluation techniques identified in the surveyed literature [14].

After discussing various ways of evaluating RSs, it is inevitable to talk about notable implementation aspects throughout the surveyed articles, the most important of which are datasets. Going through all the survey articles we came across multiple datasets that are commonly used throughout offline evaluation. Table 2 lists the cited datasets, their types, a summary of their content, the number of their occurrences, and a link to their locations. Evaluations using real-time data were rarely identified in the literature and were only reported within limited use cases. For example, in [13], analysis was based on real-time recordings of the user behavior as collected from a mobile application of a video platform in China. Within existing datasets, the most common available information is as follows:

- Item: The subject to be recommended. It may be a product, movie, or any piece of information.
- User: An entity that interacts with the RS application through implicit and explicit rating on items and in return receives recommendations for new items.
- Rating: An expression of preference by a user in regards to items. Ratings can be recorded in many forms depending on the application.
- Size: How many users, ratings, and items a dataset consists of. Example: MovieLens 100K Dataset has 100K movie ratings from 1000 users on 1700 movies.

3.2. Intrinsic blockchain aspects

Applications of RSs aim at attaining important intrinsic aspects that guarantee effective use, such as, security, privacy, reliability, decentralization, and trust. Such characteristics are prioritized according to business usage, context, and user interactions. Many of the identified intrinsic aspects of RSs can intersect with or can be improved by adopting BC technology.

When it comes to RSs, security is no less important than any other system property and it touches at all sides of accessing information assets. In addition, there are various security concerns in RSs such as authenticity and privacy. For authenticity, users often assume that the rating information and other feedback provided by the system are fair and honest. But in reality, this assumption may be inaccurate as in some business applications recommendations might be biased, like in favor of an advertisement. Findings from the evaluated research show some practices to deliver stable, accurate, and personalized recommendations. Legacy knowledge-based mechanisms usually offer high security but at low response time. On the other hand, demography-based techniques improve scalability and accuracy while being oblivious to trust. Hybrid techniques are usually developed to enjoy offering high accuracy and scalability at the same time [15].

Privacy is another critical security concern of RS. More specific to RSs, privacy means that the system should not leak any information beyond what can be derived from the output. The output recommendations should not allow for identifying user attributes. From the explored surveys, RSs use abundant amounts of personal data of users which poses privacy concerns [1]. Yet, ML, alongside other modern recommendation algorithms, has introduced techniques that aid computers to learn users' preferences and offer personalized recommendations [24].

In the context of RSs, decentralization and privacy exhibit important interrelated aspects. Peer-to-peer models were reported to focus on achieving improved privacy by distributing user information, such as user profiles and preferences. In some investigations, effective decentralized trust-based RS frameworks were reported to rely on probing product standing, client preference and seller credibility, and scalability and computational complexity of the system. In [21], it is confirmed that models that use disjoint datasets showed security improvements when using distributed systems.

Table 2
Cited datasets, types, contents, links, and the number of their occurrences (No.) in the explored survey articles.

Dataset name	Type	Contents	Link	No.
MovieLens	Movie Rating	280K users; 27M ratings; 1.1M tag applications; 58K movies	[58]	10
Bookcrossing	Book Rating	278,858 users; 1,149,780 ratings; 271,379 books	[59]	5
LastFM	Music Rating	943,347 tracks; 522,366 unique tags	[60]	5
Delicious	Social Bookmarking	16,105 instances; 500 attributes; 983 labels	[61]	5
Epinions	Trust Social Network	75,879 nodes; 508,837 edges	[62]	4
Flixter	Movie Rating	2.5M nodes; 7.9M edges	[63]	4
Netflix	Movie or Series Rating	15K titles; 5B ratings	[64]	4
Amazon	E-Commerce	303 different datasets	[65]	3
Jester	Video Clips	73,496 users; 4.1M continuous ratings; 100 jokes	[66]	3
EachMovie	Movie Rating	72,916 users; 2,811,983 ratings; 1,628 movies	[67]	3
Yahoo! Music	Music Rating	1.8M users; 717M ratings; 136K songs	[68]	3
Audioscrobbler	Music Rating	20K real people profiles	[69]	3
Foursquare	Location-based Social Network	60M+ Point of Interest (POI); 941 venue categories; 1M+ fresh tips; 18.5M+ fresh photos; 1M+ tastes; 23K+ chains; 7.6M+ venues	[70]	3
Facebook	Social Network	2.91B users	[71]	3
IMDB	Movie Rating	8,313,921 titles; 13,813,945 genres	[72]	2
Wikipedia	Machine Learning Research	Miscellaneous	[73]	2
Gowalla	Location-based Social Network	196,591 nodes; 950,327 edges	[74]	2
Flickr	Image	105,938 nodes; 2,316,948 edges	[75]	2
metacritic.com	Films; TV shows; music albums; video games; formerly books	Not available	[76]	2
rottentomatoes.com	Movie and Critic Reviews	30M unique visitors a month	[77]	2
Yelp	Business Reviews	2,189,457 users; 8,635,403 reviews; 160,585 businesses; 200K pictures; 1,162,119 tips	[78]	2
Google Research Datasets	Miscellaneous	Miscellaneous	[79]	2
Spotify	Music Rating	190M users; 1M playlists; 100K episodes; 50K h of audio; 600M+ transcribed words; 40M tracks	[80]	2
GeoLife	Location	182 users; a period of over three years; 17,621 trajectories; 1.2M km; 48K+ h	[81]	2
CiteULike	Scientific Papers	5,551 users; 16,980 items; 46,391 tags; 44,709 citations	[82]	2
WS-DREAM	Web Services Research	3 sets of data: QoS, log, and review datasets; 30+ QoS prediction approaches	[83]	2
Stackoverflow	Posts Rating	2,916,211 web; 434,212 movies; 327,622 audio; 255,330 data; 88,063 texts; 23,189 images	[84]	1
Tencent	Image	17,609,752 training and 88,739 validation image URLs; 11,166 categories	[85]	1
SNAP	Networks	Miscellaneous	[86]	1
YFCC100M	Image and Video	100M images and videos	[87]	1
WeFi	Location	Enables building datasets through connecting to millions of IoT devices and hotspots	[88]	1
YouTube-M	Video Rating	237K segments; 1,000 classes	[89]	1
HetRec	Miscellaneous	2K users; 105K bookmarks; 92,800 artist listening records; 86K ratings	[90]	1
MyPersonality	Social Science Research	6M users	[91]	1
South Tyrol Suggests	Context-Aware Suggestions	2,535 users	[92]	1
Whrrl	Loan Rating	Enables building datasets through connecting to 2M USD in online loans; 1,400 warehouses; 3 lenders; 550M+ USD commodities tokenized	[93]	1

(*continued on next page*)

Table 2 (*continued*).

Dataset name	Type	Contents	Link	No.
Yahoo!	Miscellaneous	Miscellaneous	[68]	1
Million Song	Song Lyrics	1,019,318 users; 384,546 MSD songs	[94]	1
TiVo data	Advertising	Millions of TV shows, movies, live events, web series, music albums and tracks, etc.; Metadata: moods, tones, themes, weighted keywords, age descriptors, popularity scores, etc.	[95]	1
Orkut	Community Data	3,072,441 nodes; 117,185,083 edges	[96]	1
NIPS Conference Papers 1987–2015 Dataset	Words	11,463 instances; 5,812 attributes; 56,643 web hits	[97]	1
DianPing	Social Network	147,918 users; 11,123 restaurants; 2,149,675 ratings	[98]	1
Walmart.com	Store Data	4,607,680 instances	[99]	1
RecSys Challenge	Miscellaneous	Miscellaneous	[100]	1
Retailrocket	E-Commerce	2,756,101 events; 2,664,312 views; 69,332 add to carts; 22,457 transactions; 1,407,580 unique visitors	[101]	1
Microsoft	Miscellaneous	Miscellaneous	[102]	1
MSNBC.com Web	Page visits	989,818 users; 5.7 average visits per user; 10 to 5K URLs per category	[103]	1
Caltech Pedestrian	Object Images; a list of related datasets	250K frames; 137 min long segments; 350K bounding boxes; 2,300 pedestrians	[104]	1
Outbrain	Click Prediction	700M users; 2B page views; 16.9M clicks; 560 sites	[105]	1
AOL	User Search	650K users; 20M web queries	[106]	1
Adressa	News	A Week of data collection: 15,514 users; 923 articles (in Norwegian); average article length of 518.6 words; available as a 10 weeks version	[107]	1
MovieTweetings	Movie Rating	71,690 users; 921,015 ratings; 37,998 items	[108]	1
Mobile App Retrieval	Mobile App Review	43,041 mobile application; 1,385,607 user reviews	[109]	1
Frappe	Context-aware App Usage Log	957 users; 96,203 entries; 4,082 applications	[110]	1
Global Terrorism	Terrorist Attacks	200K terrorist attacks	[111]	1
TV Audience	TV viewing habits	3K users; 217 channels; during a period of 4 months in 2013	[112]	1
Chicago Entree	Restaurant Interactions	50,672 instances; 104,999 web hits	[113]	1
CiteSeer	Labeled Network	3.3K nodes; 4.5K edges	[114]	1
ACM	Miscellaneous	Miscellaneous	[115]	1
DBLP	Bibliographic Information	5.8M+ publications	[116]	1
German Credit	Financial Credit Risk	1,000 credit-histories; 700 successful repayments; 300 defaults	[117]	1
Drug Review	Patient Reviews on Drugs	215,063 instances	[118]	1
User QoS Values	Web Service Research	339 service users; 5,825 actual web services; 1,974,675 response times and throughput records	[119]	1

Interesting concepts were identified in the literature as related to promoting trust as an important intrinsic aspect of RSs. The authors in [17] demonstrated the adoption of trust relationships between users, rather than between a user and the system. An interesting idea to ensure that users trust the used RS is transforming the data gathered from users into meaningful and easily accessible information to illustrate the reasons for choosing such recommendations—there is little research in this domain [19].

3.3. Suggested open issues

In the current era of extreme dependability on online services, virtual online presence is being a part of every single activity we do. Due to the current Covid-19 pandemic, for example, shopping, medical consultations, learning, and way much more activities demonstrated heavy reliance on online services. The explored survey articles include a variety of suggested important open issues related to RSs with a focus on those that can call for the use of BC technology.

With the high usage of online tools, saving user data is essential to every service provider. Yet, to be transparent about how personal data is used, who is it shared with, and what is it used for, are important issues. Many active online users have concerns about their data being used to train AI models, and many raise the question: "are online service providers transparent on the purpose behind using user data?" As related to RS, the question remains on to what extent online service providers are transparent about their adopted techniques and what kind of recommendations is being drawn out of the user's online behavior and interactions? All the presented questions are valid and can lead firmly to considering transparency-related matters and their potential enforcement within modern RSs [24].

While investigating the literature, we noticed a very mere focus on users' choice of their access rights. For example in [3], the problem of exploitation of existing items vs. exploration of new items in RSs was raised as an open issue, although it was not at all mentioned that users may have the right to choose the balance according to their best interest rather than the interest of the service provider. While in [19], it was suggested to produce explanations for users declaring why these specific items are being recommended to increase the user trust in the recommendation in particular and the system in general. For example, "We think you might like movie X because you have shown interest in movie Y", and in the same paper it was raised that there is a lack of research investigations discussing such concerns.

With the evolution of RSs and their techniques, we can still witness challenges and accordingly open issues in terms of quality versus cost, privacy, and security. An example of a cost-related challenge is the narrow use of recommendation techniques for specific contexts that minimizes the reusability of outcomes. Still, increasing the effectiveness of any recommendation technique leads to implementation challenges in terms of design, memory, and computations. Adopting hybrid mechanisms usually leads to increased complexity and operational cost [22]. Effective RSs must not only look into user's past interactions or behavioral information but also consider the user's live surroundings as the recommendations are being generated like time information or location [9]. Privacy and security metrics challenge the scope of access the RS is allowed to have. To that end, saving such kind of private context-tracking data requires soliciting the permission of the user. This again highlights the importance of customizing RSs algorithms according to users' preferences. The findings of our exploratory survey are summarized in Table 3. The table summarizes modern RS domains, techniques, and noted important open issues. The identified open issues are related to accuracy, performance, cold start, sparsity, scalability, security, privacy, trust, and standardization.

4. Use case: RSBC international bank

Recommender systems implementation in financial services companies is making a significant impact on the sector. RSs allow personalization of user experience, improving the overall customer satisfaction, enabling cost minimization and an increase in revenues. Indeed, several banks have already been using a recommender system or are building one. Examples of such banks include Emirates NBD (UAE), BBVA (Spain), Banorte (Mexico), CitiBanamex (Mexico), and HSBC [120].

Consider the case of the fictitious RSBC International Bank. The bank Chief Executive Officer (CEO) is convinced about the benefits of using a recommender system to help sell new services and products. Still, he/she is worried about the security issues related to such an approach. In particular, the CEO wants to guarantee the following security features: privacy, trust, integrity, confidentiality, and availability. The Chief Technology Officer (CTO) suggested using a BC-based RS solution as this will address these security issues inherently and provide more flexibility and transparency than a traditional approach. The CTO explains that only authenticated users can access the blockchain by using a permissioned or consortium blockchain, which guarantees privacy, confidentiality, and trust. The BC will also ensure the integrity of the stored data and its persistence. Being distributed among several nodes in the different branches of the bank, the BC guarantees availability as every node has a complete copy of the data, preventing Denial of Service (DoS) attacks. The recommendation algorithms will execute as smart contracts, also stored on the BC, which will increase the transparency of the system and allow for more flexibility in the introduction of new recommendation techniques. Furthermore, the CTO explains to the CEO that other well-known financial institutions are already using BCs for other purposes, ripping the benefits of this technology, which makes using it for a RS even more reasonable. Such financial institutions include J.P. Morgan, PayPal, and the Bank of China [121]. Convinced, the CEO approved the use of a BC-based RS.

5. Blockchain-based recommender systems

Several investigations in the literature have proposed BC-based RS frameworks. In what follows, we present an exploratory survey of existing frameworks that have been proposed in the literature, and provide an overview of their operation and functionalities. The exploration enables the synthesis of a taxonomy of BC-based RS system properties.

5.1. Review of existing blockchain-based recommender systems

We classify the BC-based RS in the literature into two categories: on-chain and off-chain computations. In the on-chain computations scenario, SCs perform the required computations to generate recommendations, while in the off-chain case, computations are executed by third parties and the data storage is on-chain which provides better computational capabilities. We also summarize and present the surveyed literature.

5.1.1. On-chain computation

The idea behind using the BC infrastructure to address privacy concerns in RSs was proposed in [6], where the authors proposed the storage of the encrypted user data on the BC and the usage of secure multiparty computation to compute recommendations. The authors propose an efficient solution in three core processes of the retail business: providing recommendations, self-checkout, and payment while the data owner has access to the input data. BC and beacon technology [122] are combined to provide a secure shopping experience. In [123], a BC-based RS for e-commerce is proposed using the framework provided in [6]. In the proposed system, user data such as demographic attributes, shopping history, favorite products, preferences, and habits are recorded on the BC. This arrangement benefits from the encryption provided by BC technology to avoid data tampering, enabling companies to compute recommendations without having access to raw user data. Additionally, the proposed recommender offers transparency whereby users can specify the way their data is used for computations through SCs. The user can then request recommendations. As the company can handle different user profiles, their data can be used by a CF algorithm to compute recommendations in the BC. However, in both [6,123], the BC part of the system was presented as a black box with no discussion on implementation aspects.

In [124], a platform for decentralized recommendations was proposed using SCs that run on top of a BC. The proposed architecture tackles the drawbacks of centralized RS architecture, where users need to trust a central authority that controls the decisions on how the user ratings are used to rank items, and also can tamper with user data to recommend specific items. The platform offers users the ability to register using a pseudonym, avoiding disclosure of personal information. Upon registration, users can create and add new items to the framework, rate items that have been already added, and compute the score of a specific item. These functionalities are implemented and executed through several Ethereum SCs. The authors considered simple functions that users can choose from for score computation; specifically the scoring average and weighted score average. These simple functions are not capable of delivering the same performance as traditional RSs. This issue was addressed in [5] where a trust-based CF recommender that relies on BC was proposed. The system incorporates SCs that users can interact with via a metamask to append their ratings. The authors distribute the process of CF between on-chain and off-chain computation. SCs are used to compute the similarity matrix of user-based and item-based CF. This matrix is then sent to an off-chain model that is used to compute predictions. The proposed system was evaluated in terms of gas usage, where extremely large values were observed due to the calculation of the similarity matrices. A federated learning approach was proposed in [125], where a QoS-aware BC-based RS for web services was proposed. Conventional web service RS does not provide QoS guarantees, which considers factors such as price,

Table 3
Summary of state-of-the-art recommender system domains, techniques, and suggested open issues.

Ref.	Year	Title	Domains	Techniques	Open Issues
[8]	2022	Blockchain-based recommender systems: Applications, challenges and future opportunities	Energy, healthcare, e-commerce, IoT, social networks, e-learning	CB, CF, MF, AI and ML, adversarial ML, deep learning, federated learning	BC protocols' sustainability, lack of regulations, interoperability
[7]	2021	A Survey of privacy solutions using blockchain for recommender systems: Current status, classification and open issues	Online services	CB, CF, hybrid	Privacy problems, blockchain applications
[55]	2021	A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks	Tourism, movie, fashion, music, electronic, business, product, news	CB, CF, deep learning, federated learning, Adversarial machine learning (AML), adversarial training, Generative Adversarial Networks (GANs).	Bridging the gap between attack and defense models, choice of recommendation model, scalability, performance evaluation
[56]	2021	Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison	Movie, joke	CB, CF	Quality of recommendation
[14]	2020	Recommender systems for smart cities	Filter relevant information, upgrading relations between stakeholders and civil society, assist tech platforms in decision making	CB, CF, social-based, KBF, DBF, context-aware	Evaluation
[3]	2020	Convergence of recommender systems and edge computing: a comprehensive survey	IoT devices	CB, CF, KBF, Hybrid Filtering (HF)	Cold start, exploration, exploitation, security, privacy
[2]	2020	A systematic study on the recommender systems in E-Commerce	E-commerce, Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), Business-to-Business (B2B)	CB, CF, DBF, HF, KBF	Adaptivity, serendipity, risk, novelty, privacy
[16]	2020	Mobile music recommendations for runners based on location and emotions: The DJ-Running system	Music, sports, location, emotion	Nearest Neighbor Search algorithm (NNS), CF, user-based filtering, similarity-based Filtering	Not mentioned
[23]	2019	Progress in context-aware recommender systems—an overview	Facebook, Google, YouTube, Bing News, Music, books, food, health, articles	CF, MF, Tensor Factorization (TF), Latent Dirichlet Allocation (LDA), Learning to Rank (LTR), Singular Value Decomposition (SVD)	Bias, cold start, sparsity, dimensionality, adaptivity, contextual modeling, privacy, availability, copyrights
[19]	2019	Multi-Criteria Review-Based recommender system–The State of the Art	Hotels	CB, CF, KBF, HF	Accuracy
[1]	2019	Scientific paper recommendation: a survey	Economic, education, scientific research	CB, CF, graph-based, hybrid	Cold start, sparsity, scalability, serendipity, privacy, standardization
[18]	2019	Deep Learning Based recommender system: A Survey and New Perspectives	Computer vision and speech recognition, games and self-driving cars	deep learning, CB, MF, CF, TF	Hyperparameter Tuning, interpretability
[9]	2018	A Survey of CF-based recommender systems: from traditional methods to hybrid methods based on social networks	Social networks	Not mentioned	Sparsity, dimensionality
[17]	2018	A Survey of recommender systems based on deep learning	Not mentioned	CB, CF, deep learning, HF, context-aware	Performance, scalability, explainability, attention mechanisms in Deep Learning, modeling techniques
[22]	2018	Sequence-Aware recommender systems	Media, e-commerce, apps, music, learning, Ads, Web, click prediction, financial	Deep learning, Markov Models, Reinforcement Learning, Recurrent Neural Networks	Bias, sparsity, scalability, complexity, rule base size, mining, hyperparameter configuration
[11]	2018	Objectives and State-of-the-Art of Location-Based Social Network recommender systems	Wi-Fi	GPS-enabled devices Base Stations	Cold start, sparsity, diversity
[24]	2017	The use of ML algorithms in recommender systems: A systematic review	Movies, social, academic, news, e-commerce, web-pages, documents, music, books, health, images, tourism, games, pictures, clothing, email, jobs, restaurant, ads, elections, jokes, mobile phones	CB, CF, Ensemble, K Means, Support Vector Machines (SVM), Bayesian, Decision Tree, Matrix Factorization	Performance, data storage, modeling techniques

(*continued on next page*)

Table 3 (*continued*).

Ref.	Year	Title	Domains	Techniques	Open Issues
[15]	2017	Cloud services recommendation: reviewing the recent advances and suggesting the future research directions	Cloud recommender system	CF, DBF, KBF, hybrid	Cold start, scalability, dimensionality, security, trust, quality
[10]	2017	Recommender systems for online and mobile social networks: a survey	Online and mobile social networks	CF, CB, Graph-Based Filtering (GBF), context-aware, social-aware, tag-based, location-aware	Context awareness
[13]	2016	Location-aware and personalized CF for web service recommendation	Mobile Apps	CB, CF	Complexity, data storage
[21]	2016	Parallel and Distributed CF: A Survey	Web Recommendation, movie/TV recommendation, information/document recommendation	CF, CB, data mining context-aware, hybrid, social filtering, MF	Sparsity, scalability

reliability, and response time. Additionally, to preserve their privacy in centralized recommenders, users prefer not to disclose their ratings of a specific web service they have used. Hence, it is usually difficult to obtain QoS values for web services. This proposed system stores the user's QoS values on a BC to preserve their privacy. A federated MF algorithm is then used to build a recommendation model without having access to the user's raw data.

Additionally, a BC-based framework for collaborative learning of ML models was proposed in [126]. The proposed architecture offers an alternative way of crowdsourcing data for retraining ML models, which may be supervised or unsupervised, by initiating SCs with users that can use the model deployed on the BC for inference. By using this publicly shared model, the SC allows users to submit their data to the BC, which is in turn used to improve the model's performance incrementally. The authors state that such a model could be applied to RSs. Evaluations of this framework were later presented in [127], where the results showed improvements in the shared model's accuracy on the testset over time.

5.1.2. Off-chain computation

Instead of performing computations on-chain using smart contract functionalities, various works have proposed on-chain data storage and off-chain computations. This offers more flexibility in terms of what type of computations can be executed, in addition to memory resources and computational capability. For instance, the exchange of information sharing was made through the BC in [128,129], where a BC-based RS framework was presented as an approach to provide privacy over user-sensitive data in CF recommenders. This proposed framework ensures the security and privacy of the user's data by adopting a user-centric approach. The consent of the user is required to be able to utilize their data for the system. When a user requests a recommendation, the data needed by the recommender is first requested from users through a manager. Upon approval from other users, their data is sent from the cloud and this transaction is then recorded on the BC. CF is then used to compute recommendations in an off-chain manner. A similar approach is proposed in [117], where a BC-based credit score recommender was proposed. The recommender is intended for financial institutions and relies on BC for storing information of prospective borrowers. When credit score recommendation is required, this data is fetched from the BC for off-chain computation using a Long Short Term Memory (LSTM) recurrent neural network model.

Recommendations in the pharmaceutical industry were targeted in [118], where a BC-based RS for pharmaceutical drugs based on a dataset of drug reviews was proposed. In this respect, the approach proposes a ML-based RS that analyzes reviewers' sentiments for various types of pharmaceutical drugs. The system collects drug reviews from users and stores them efficiently using BC and an off-chain storage database. This data is then fetched by a recommendation module that pre-processes the textual data and provides a list of top drugs based on users' comments. This model is retrained as new comments are submitted by users of the system.

A BC-based service recommendation scheme that supports data sharing among cloud platforms was proposed in [130]. Data sharing among cloud servers gives each owner of the server more data to train their recommender on, leading to improved performance. To protect the security and privacy of data, a distributed ledger approach rather than a centralized one was adopted. The proposed system model is composed of a user layer, at which users access web services and generate data consequently, a data-sharing layer, at which a consortium BC is utilized to share data among the different cloud services providers, and a data layer, at which cloud servers are used to store large files, such as images and videos, in a shared cloud. A security analysis of confidentiality and tamper-proofing and a performance evaluation of the accuracy of the proposed system were presented.

Communication and computational costs are usually a drawback in BC-based RSs. From the BC side, this cost is in the transactions and verifications. From the RS's side, this cost is imposed by the computations that consider all users at once. In [131], a framework for efficient BC-based CF was introduced to mitigate these costs. The architecture first clusters users together through a Locality Sensitive Hashing (LSH) clustering mechanism. This mechanism creates a hash of the user profile, given his ratings of items, and groups users with similar hashes into similar clusters. Hence, when computing recommendations, the recommender does not consider all the ratings of other users but only the ratings of users with a similar profile, thus decreasing computational time in item score computation. Users send their ratings to an InterPlanetary File System (IPFS), which is used for off-chain storage, and append their metadata (such as the hash of this data) to a BC. The architecture stores information off-chain and only shares the hashes instead to reduce the total information in the transactions. To compute recommendations, the corresponding metadata is extracted from the BC and is then used to retrieve the data stored off-chain. In this regard, secure multiparty computation is applied between users in the same cluster, which computes similarities. By comparing to conventional, centralized RS architecture, the proposed architecture achieved similar performance measured by the Mean Absolute Error (MAE) but provided much better computational efficiency in a privacy-preserving manner.

The authors in [132] proposed another work relying on off-chain computation. The focus of the work was to leverage the inherent advantages of BC, combined with LSH and local-differential privacy techniques, to provide a privacy-preserving recommender system. In addition, the authors proposed to use both on-chain and off-chain storage where the data is stored on the IPFS while the data hash is stored in the BC.

A knowledge-graph-based RS that relies on on-chain data storage was proposed in [133]. Knowledge graph-based deep learning RSs benefit from the collection of interlinked information available in knowledge graphs and the ability of deep learning techniques to learn

Table 4
Application domain, motivations, proposed solutions, performance metrics, and implementation aspects of the surveyed BC-based RSs.

Ref.	Year	Application Domain	Problem	Solution	Performance Metrics	Implementation Aspects	
						Technology	Dataset
[132]	2021	General	• Privacy-Preserving	• Local sensitive hashing • Local differential Privacy	• Precision, Recall, F-score	Not Mentioned	MovieLens 100K MovieLens 20M
[5]	2020	Movie Recommendation	• Restriction on the functions that can be used in computation	• Proposed the computation of the similarity matrix on-chain • Proposed off-chain computation of the rest of the process	• Gas Cost • Precision, Recall, F-score	Not Mentioned	MovieLens 100K
[117]	2020	Credit Score Recommendation	• Security attacks in financial institution systems • Centralization of user information	• Propose a BC-based architecture for credit score recommendation • Record hashes of user data on the BC	• Accuracy, Error Rate • Area Under the Curve • Precision, Recall, F-Measure • Communication Cost (bytes) • Computation Cost (s)	Ethereum SC	German credit
[118]	2020	Drug Recommendation	• Lack of security in the drug supply chain	• Propose a BC-based smart drug delivery system • Propose RS that recommended top drugs based on user comments	• Training loss & accuracy • Response time (ms) • Latency (ms)	Java/NodeJS	Drugs dataset
[130]	2020	Web Service Recommendation	• Issue of privacy in data-sharing between cloud servers	• Record data shared between different servers on a consortium BC	• Mean Absolute Error • Root Mean Squared Error • Gas Cost	Ethereum SC	WS-DREAM
[128,129]	2019, 2020	Not mentioned	• Control of central authority over user-sensitive data • Ability of central authority to manipulate user data	• Store user data on a temporary storage which can be accessed by the RS • Record transactions between the temporary storage and the RS on the BC	• Processing time (ms)	Not Mentioned	Not mentioned
[125]	2019	Web Service Recommendation	• Privacy concerns in centralized RSs • Lack of user credibility assessment in centralized RSs	• Propose a decentralized federated matrix factorization algorithm • Propose a consensus process based on Practical Byzantine Fault Tolerance	• Mean Absolute Error • Root Mean Squared Error	Ethereum SC	WS-DREAM
[124]	2019	Not mentioned	• Need of trust between users and central authority • Ability of central authority of tampering with user data • Unawareness of users to methods used in computations	• Propose a BC-based architecture for general-purpose RS • Implement RS functionalities using SCs • Allow users to specify method used in rating computations	• Gas Cost • Throughput • Scalability	Ethereum SC	Not mentioned
[131]	2019	Movie Recommendation	• Communication and computation overhead • Possibility of data-tampering in centralized RS	• Propose the usage of LSH for grouping similar users into smaller clusters • Record hash of user ratings on the BC when being sent to off-chain storage • Implement secure multiparty computation for computing recommendations	• Disclosure Risk (DR) • Mean Absolute Error	Ethereum SC	MovieLens
[133]	2019	Work Task Recommendation	• Challenges of constructing knowledge graphs • Centralization that permits data tampering	• Propose a crowdsourcing voting scheme for knowledge-graph construction • Use BC-powered SCs for secure and decentralized data storage • Update knowledge-graph with daily user votes stored on the BC	• User Satisfaction	Ethereum SC	Not mentioned
[134]	2019	Web Service Recommendation	• Ability of malicious users to submit false data	• Propose a BC-based approach for verification of user reliability	• Mean Absolute Error • Ninety Percent Relative Error	Ethereum SC	User QoS values
[6,123]	2016, 2016	E-Commerce	• Access of companies to highly-sensitive user data • No consideration for user-privacy in RSs	• Use BC for multi-party computation of user data	• Not mentioned	Not Mentioned	Not mentioned

non-linear relationships between users and items. However, knowledge graphs are known to be tedious and time-consuming to construct given the complex procedure required in their development. This becomes more challenging when the system is centralized, raising serious security concerns. In [133], a decentralized knowledge graph-based deep learning RS is proposed, where the knowledge graph is built via crowdsourcing voting. The proposed system is formed by two on-chain and off-chain parts. In the on-chain part, users perform voting through SCs that store the submitted data to the BC, which is then used to update the knowledge graph. In this off-chain part, the graph embeddings are extracted and fed as input to a deep learning RS for predictions. The proposed system was implemented to be used as an RS for enterprise leaders, that assists them in recommending work tasks for their employees. In this regard, the crowdsourcing voting mechanism was implemented using SCs, where the employees would submit static attributes related to their skills and background, and dynamic attributes that are their daily work reports. After being stored on the BC, the dynamic attributes are used to update the knowledge graph, a process that was run over a period of 5 months and resulted in 23,100 votes. At the end of this period, the knowledge graph was constructed, and the graph embeddings were extracted to be used as input to the deep learning RS. This resulting model was then used by enterprise leaders to recommend tasks for their employees. The performance of the model was later evaluated through questionnaire surveys, after being used for 3 months, and was found to be 83.3% satisfactory.

A consortium BC-based user authentication mechanism was proposed in [134] along with a decentralized BC-based MF approach for QoS prediction. Conventional QoS prediction algorithms do not consider the credibility of users when predicting QoS values for web services. It is thus possible for malicious users to submit false QoS values for a specific web service, which they would like to market. To eliminate such malicious users, the proposed method uses the homomorphic hash and the Byzantine agreement within its procedure, which help in the identification and elimination of malicious users. Users of the recommender are given addresses of which the hash is stored in the BC. Before making predictions, users who submit QoS values are verified by other users by comparing their hash value to the one stored in the BC. Compared to other works in the literature that consider user credibility in QoS prediction, the proposed method showed the best performance measured by the MAE and the Ninetieth Percentile Relative Error (NPRE).

Table 4 summarizes the surveyed literature on BC-based RSs. The table highlights the application domain, motivation, proposed solutions, performance metrics, and implementation aspects.

5.2. Taxonomy synthesis

Recently, there has been a growing interest in using the decentralized architecture of BC in building RSs. The main attraction of BC has been to benefit from the multiple appealing characteristics of BC and

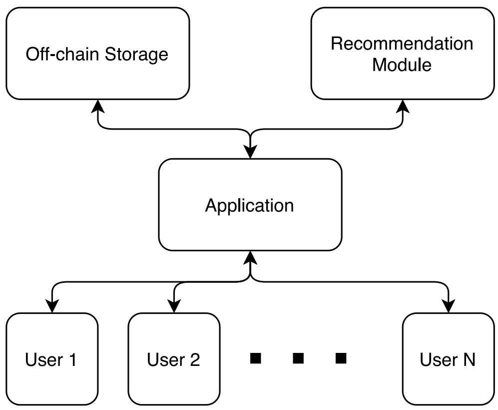

Fig. 2. Traditional RS architecture.

overcome the previously discussed drawbacks of traditional centralized RSs. BC-based RSs aim to attain the following features:

- **Transparency:** Users are no longer oblivious to how item scores are being predicted. Items, users, and score computing functions are visible to all users of the RS while preserving users' privacy.
- **Decentralization:** The decentralized architecture of BC eliminates the need for users to trust the central authority that would usually have control of user data.
- **Tamper-proofing:** Since BCs are composed of blocks linked together with hash functions, they are tamper-proof. Hence, a BC is suitable for storing data (or its hash) about user ratings to avoid any attempts at tampering with these ratings for the purpose of recommending specific items over others.
- **Persistence:** The user ratings and SCs are always available on the chain and cannot be removed. Therefore, the removal of user-submitted ratings is not possible.
- **Customization (through smart contracts):** A customization feature can be implemented to allow users of the RS to choose which function or method to be used when computing item scores.
- **Authentication:** Another feature that is important in BC-based RSs is the detection of malicious users that can create fake profiles and submit false ratings for specific items. Authentication can be achieved using permissioned or consortium blockchain.
- **Privacy:** Preserving user privacy is important for user-centric RSs where the data used in computation is sensitive. These types of recommenders are important for companies that aim to provide personalized recommendations, and thus create detailed user profiles that contain their preferences, behavior, and needs. For instance, this type of data could be a user's search history that they would not like someone to have access to. This issue is alleviated in BC-based RSs where opaque pseudonyms are used to refer to users, hence, achieving privacy.

Another critical design choice that intervenes when using BC in RSs is the type of nodes. Indeed, nodes participating in a BC system can be of two types: full nodes or lightweight nodes. Full nodes have a full copy of the BC and participate in the functioning of the BC, which means that a full node will participate in validating transactions and blocks. These operations are in general resource intensive. Lightweight nodes, on the other hand, do not keep a copy of all the BC and hence are suitable for resource-constrained nodes like in an IoT environment where devices have severe computation and memory limitations. The type of nodes to use in a BC-based RS depends on the target application context and properties of the participating entities.

Table 5 highlights the targeted features in the proposed BC-based RSs of the surveyed literature, in addition to whether recommender computations and data storage are done on-chain or off-chain. We note that several proposed architectures rely on both on-chain and off-chain storage, where the hash of the user data is stored on-chain and the actual user data is stored on specific off-chain data servers from which data is retrieved by its hash.

6. A modular architecture for blockchain-based recommender systems

Inspired by the surveyed literature, a modular architecture for RS that incorporates BC technology is presented in this section. The proposed architecture aims at being flexible and modifiable to serve application-specific requirements. Firstly, traditional RS architectures are presented with an emphasis on their own centralized data storage and computations. Then, the developed BC-based architecture is detailed. The presentation of the different architectures includes describing how each of the two procedures of feedback collection and storage, and recommendation computation are performed.

6.1. Traditional recommender system architecture

The typical traditional recommender architecture is composed of an application that receives data from the users and provides them with recommendations by using two entities as shown in Fig. 2. These entities include a centralized storage server and a recommendation module.

The procedure for collecting and storing user-submitted data (e.g., explicit rating), or data collected from user activity (e.g., implicit rating) is shown in Fig. 3. This figure shows two sequence diagrams in a typical traditional RS architecture. Two important processes are described: feedback collection and recommendation computation.

6.1.1. Feedback collection and storage

- **Feedback Submission (1a):** The data collection process begins by receiving data from users. Explicit feedback such as ratings can be submitted by users of a specific application via their User Interface (UI) and is handled by the application. Implicit feedback based on user activity may also be collected from the users at this stage.
- **Feedback Storage (1b):** The application receives user data and sends it for storage in a centralized storage server that most traditional RS adopt.

6.1.2. Recommendation computation

- **Recommend (2a):** For the recommendation process to start, a recommend event is generated to trigger the recommendation process. This triggering can happen every specified duration of time, when a significant amount of user data becomes available, or upon user request or activity, for example when the user interacts with an item.
- **Get Data (2b):** The application engine sends a request for the retrieval of the collected data.
- **Data (2c):** The centralized storage server sends the collected user data to the application.
- **Get Recommendation (2d):** The application transfers the data received from the storage server to the centralized computation server in the recommendation module.
- **Compute Recommendation (2e):** At this stage, data pre-processing and computations can be performed according to the fixed and chosen recommendation method such as CF, CB filtering, or others.
- **Recommendation (2f):** Upon completion of the computation process, the recommendation algorithm forwards its output, specifically the generated recommendations, to the application.
- **Recommendation (2g):** The application finally presents the corresponding recommendations to the user.

Table 5
Taxonomy of Blockchain-based Recommender Systems.

Ref.	Year	Transparency	Decentralization	Tamper-proofing	Persistence	Customization	Authentication	Privacy	Recommender computations		Data storage	
									On-Chain	Off-Chain	On-Chain	Off-Chain
[132]	2021		✓	✓	✓			✓		✓	✓	✓
[5]	2020		✓	✓				✓	✓		✓	
[117]	2020	✓	✓	✓			✓			✓	✓	✓
[118]	2020		✓	✓						✓	✓	✓
[130]	2020		✓	✓	✓			✓		✓	✓	✓
[128,129]	2019, 2020		✓		✓		✓	✓		✓	✓	✓
[125]	2019		✓				✓	✓	✓		✓	
[124]	2019	✓	✓	✓	✓	✓	✓		✓		✓	
[131]	2019		✓	✓				✓		✓	✓	✓
[133]	2019	✓	✓	✓	✓		✓			✓	✓	
[134]	2019		✓				✓			✓	✓	✓
[6,123]	2016, 2016		✓		✓			✓	✓		✓	

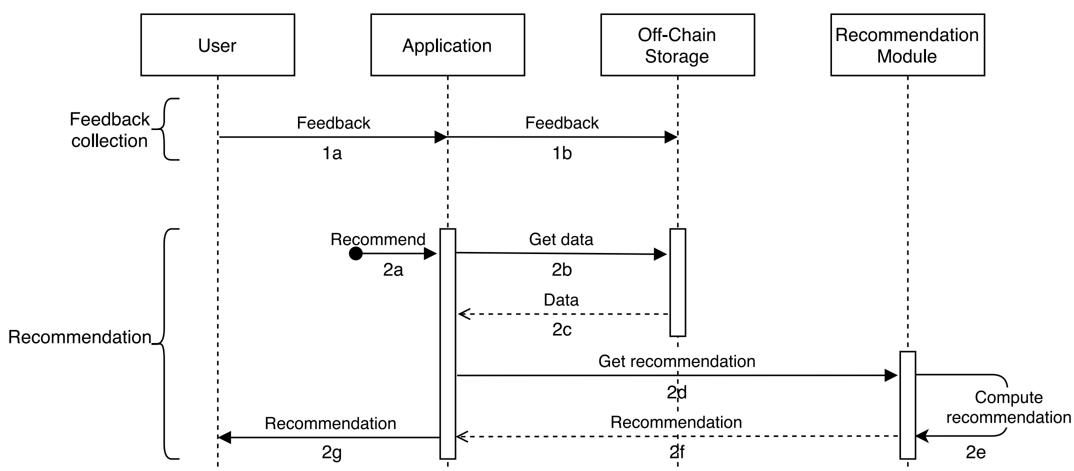

Fig. 3. Sequence diagram in a typical traditional RS architecture.

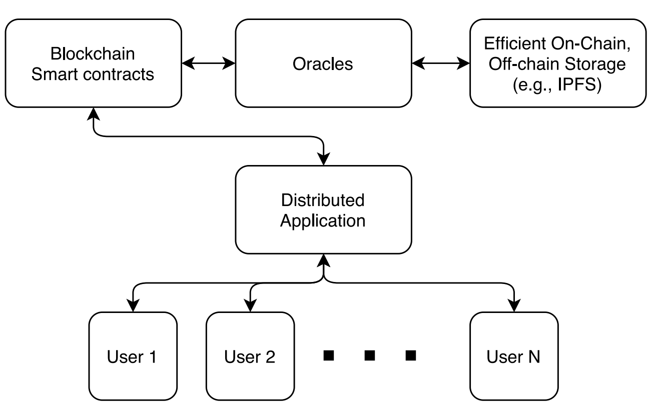

Fig. 4. A typical Blockchain-based RS architecture.

6.2. Blockchain-based recommender system architecture

In the following subsection, we propose a BC-based RS architecture. Instead of the centralized approach that most traditional RSs adopt, we propose to use an efficient BC-based storage procedure that relies on the data stored on the BC as well as on decentralized persistent storage. In addition, trusted oracles are used to aid SCs with on-chain and off-chain computations.

The architecture is composed of an application that transfers data between users and three entities for storage and computation as shown in Fig. 4. These entities include BC smart contracts, oracles, and on-chain and off-chain storage services.

As SCs are not able to pull data that is stored off-chain, there will be a need to rely on oracles that will push the data to the BC so that it could be used by the smart contract. According to [135], an oracle is an interface that delivers data from external data outside the BC to a smart contract to consume. Oracles are controlled by SCs to perform specific tasks. It is also possible for SCs to distribute computations between on-chain nodes and the oracles. In the case of off-chain storage and computation, the SCs perform a validation step on the results provided by the oracles to maintain trust. As the oracles are generally arranged in clusters, a simple and efficient technique is to use majority voting among the oracles and to hold the cluster head responsible for performing the required action while the other oracles of the cluster can serve as watchdogs.

Fig. 5 shows three sequence diagrams for the BC-based architecture. In what follows, we provide a description of the typical operations of the architecture for feedback collection and storage, as well as for recommendation computation.

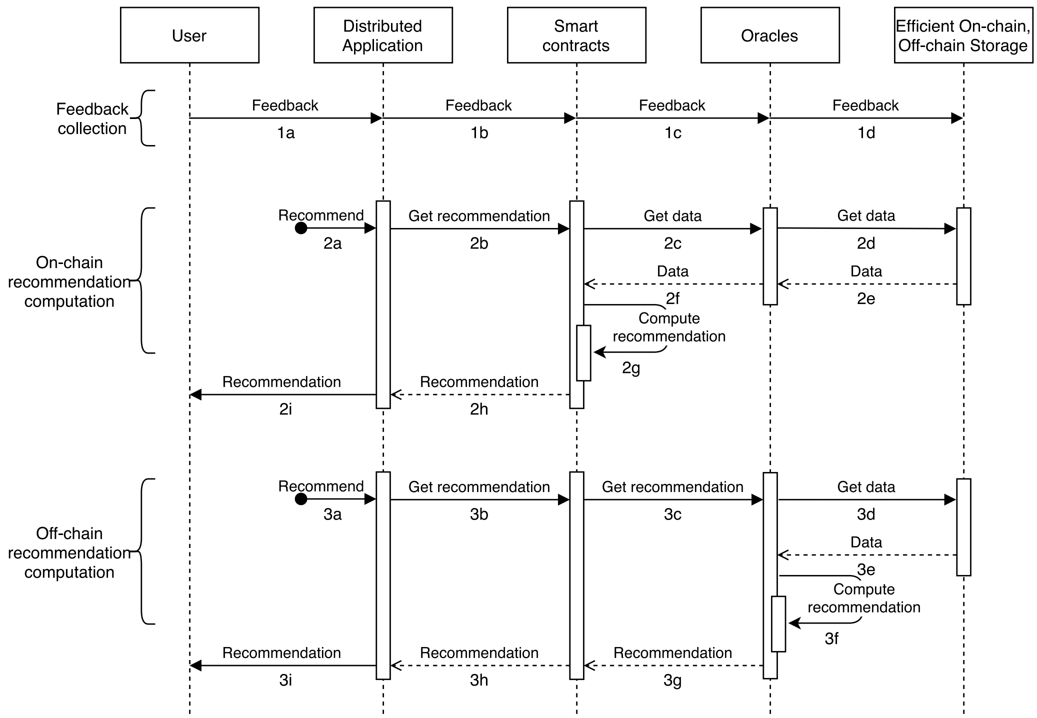

Fig. 5. Sequence diagram in a typical Blockchain-based RS architecture.

6.2.1. *Feedback collection: Efficient on-chain/off-chain storage*

- **Feedback Submission (1a):** Similarly to how the process starts in the centralized scenario, data is received from users by the distributed application.
- **Feedback Storage (1b):** The application then forwards the user-submitted data to the SCs.
- **Feedback Storage (1c)** and **(1d):** After the hash of the user-submitted data is stored on the BC as a transaction guaranteeing tamper-proofing (on-chain storage), the data is stored in the storage server in an off-chain manner. An IPFS can be used as a storage server where objects are retrieved by their SHA256 hash. This storage is not like a typical centralized storage server from which data can be retrieved based on its location, but rather a file-sharing system where objects can be retrieved by their hash. Hence, efficient BC-based storage can be achieved. Also, the fact that the provided storage is content addressable makes it impossible to change the content without changing the content identifier hence reinforcing the immutability provided by the BC. To maintain trust, the smart contract validates the operations of the oracles.

6.2.2. *Recommendation computation*

The typical procedure for computations and providing recommendations to users that rely on the data stored on the BC can be done in two ways: on-chain and off-chain. These are described below:

1. **On-Chain Recommendation Computation**

    - **Recommend (2a):** For the recommendation process to start, a recommend event is generated to trigger the recommendation process. This triggering can happen every specified duration of time, when a significant amount of user data becomes available, or upon user request or activity.
    - **Get Recommendation (2b):** The application asks for a recommendation. This request triggers a smart contract to start the recommendation process.
    - **Get Data (2c):** The smart contract instructs the oracles to retrieve the required transactions stored on the off-chain storage (e.g., IPFS) to compute the recommendation.
    - **Get Data (2d):** The oracles get hashes of the transactions previously recorded from the BC to be used with the off-chain storage.
    - **Data (2e):** Using the hashes, the corresponding relevant user-submitted data is retrieved from the storage service.
    - **Data (2f):** The required data is then made available to the SCs. The smart contract performs a validation step to check the validity of the provided data.
    - **Compute Recommendation (2g):** The data is used by the BC-powered SCs to compute the recommendations. Any choice of recommendation algorithm can be used at this stage subject to the implemented SCs and their limitations.
    - **Recommendation (2h):** When computations are performed successfully, the smart contract forwards the recommendations to the application.
    - **Recommendation (2i):** The recommendations obtained from the smart contract are finally transferred to the corresponding users.

2. **Off-Chain Recommendation Computation**
    The procedure for off-chain computations and providing recommendations to users is shown in Fig. 5.

    - **Recommend (3a):** Same tasks as in 2a.
    - **Get Recommendation (3b):** Same tasks as in 2b.
    - **Get Recommendation (3c):** The smart contract will send a request to the oracles to retrieve the relevant user data and compute the recommendation.
    - **Get Data (3d):** The oracles get hashes of the transactions previously recorded from the BC to be used with the off-chain storage.
    - **Data (3e):** Using the hashes, the corresponding relevant user-submitted data is retrieved from the storage service and sent to the oracles.

- **Compute Recommendation (3f):** In this scenario, the oracles handle the pre-processing and computation procedures that are normally done on centralized servers in traditional RSs. Any choice of recommendation algorithm can be used at this stage. The smart contract performs a validation step to check the results of the oracles computation.
- **Recommendation (3g):** When computations are performed successfully, the oracles push the output of the recommender to the smart contract.
- **Recommendation (3h):** The recommendations are forwarded to the application.
- **Recommendation (3i):** The application finally transfers recommendation to the corresponding users.

7. Discussion

The proposed architecture promises significant improvements to traditional RSs by integrating BC technology. To check the validity, a procedure is adopted that includes iterating through the surveyed BC-based RS articles to check their conformance with the proposed modular architecture. Also, the limitations of SCs in the context of RSs are thoroughly discussed. Then, the transformations and future trends in the area are explored.

7.1. Architecture validation

The BC-based RSs surveyed in Section 5 are broadly classified into two categories: on-chain and off-chain computation approaches. Our proposed architecture for BC-based RSs is designed to support both approaches. In the on-chain computation approach, the recommendation algorithm computations are done on BC nodes before returning the results to the RS users. This approach, adopted in [5,6,123–125], is feasible through our proposed BC-based RS architecture, where on-chain recommendation computation is performed by following steps **2a** till **2i** of Fig. 5. Additionally, the reliance on SCs and oracles in our proposed architecture allows for flexibility in terms of customization of choice of computational methods as proposed in [124], or load distribution of computation (on and off-chain) as proposed in [5].

In the off-chain computation approach, adopted in [117,118,128–133], an external server is used to avoid on-chain computational limitations. These approaches require an off-chain storage server from which data can be retrieved by their hash previously stored on the BC. In this regard, such off-chain computation approaches are feasible through our proposed architecture by following steps **3a** till **3i** of Fig. 5. An efficient on-chain/off-chain storage procedure is adopted in the architecture, followed by pre-processing and computations on oracles. The usage of oracles for computation in step **3f** also allows for the use of computationally-heavy approaches such as deep learning [128,129,133]. Therefore, our proposed architecture provides a general framework that covers the main aspects presented in the architectures of the surveyed literature in terms of full reliance on on-chain computation and storage or distribution between on and off-chain resources, as shown previously in Table 5. Our architecture covers all possible scenarios and can be further tuned for specific desired customization.

7.2. Limitations of smart contracts

While SCs offer several benefits, they also suffer from some critical limitations.

Amending and Termination: The fact that SCs are immutable brings challenges to their usage in RSs. Specifically, this challenge is significant to companies that frequently update their recommendation pipelines to improve performance. Suppose a company used to adopt a specific method in computing item similarities but wants to change it to a novel advanced method. Another situation is when bugs or unintended coding errors are discovered in the code of the deployed recommender, making it deliver invalid recommendations. Traditional programming allows developers of this company to easily modify the code in which their recommender is written and change the way computations are being made, or stop the service temporarily. This option is not available in SCs, however, making it difficult for businesses to update and control their recommenders [136]. The company can of course deploy a new contract and redirect the new requests to it, however, special attention has to be paid to avoid misconfigurations.

Automated Nature & Objectivity: SCs are relentlessly and automatically executed, without any human intervention. Indeed, this feature is useful for various use cases, but also has its drawbacks when it comes to certain recommenders. For instance, consider an RS for financial institutions that recommend whether or not to give a client a financial loan. To get this loan, the client needs to satisfy specific requirements, otherwise, the smart contract will automatically decline the loan request. In real-world scenarios, the client might be a loyal client to the financial institution and has a good history of paying back his loan but did not satisfy all of the requirements. In this case, the financial institution may accept the request of the client to preserve the long-term relationship, despite not having all the requirements met. This is not possible with automatically executable SCs [136]. Hence, this automated nature may not suit some businesses in the real world.

Scalability: SCs run on BC networks which have limitations in terms of transaction processing rate and latency. At present, Ethereum can process approximately 15 to 25 transactions per second. According to [137], VISA can handle 65,000 transactions per second. This has been criticized in [138] where it is claimed that the processing rate of VISA does not exceed 1,700 transactions per second. In both cases, compared with the capabilities of centralized financial systems such as VISA, the processing rate of Ethereum is quite low, because of the time the consensus protocol, transaction validation, and decentralized execution of programs take. High latency occurs due to the low transaction throughput. It is therefore difficult to build large-scale systems that can perform in real-time [139,140]. This applies to the case of large-scale RSs of large social media platforms that rely on advanced processing capabilities. Approaches such as a parallel Proof of Work (PoW) [141] and state-machine replication [142] have been proposed to improve scalability. Also, new systems are moving away from PoW to Proof of Stake (PoS) that is faster. In addition, the use of sharding improves scalability. Both of these are to be used in the new version of Etherium, Serenity.

Performance Issues: Despite benefiting from the immutability, tamper-proofing, and transparency of the BC, this comes at the expense of computational and storage capabilities [143]. Hence, RSs that rely on SCs for on-chain computation do not promise great performance, due to the limited number of inputs they can handle, in addition to the type of computations that can be done, as was observed in the results in [124,144]. This problem is not only limited to RSs but is encountered with all kinds of BC-based applications. A possible resolution to this issue is the off-chaining of data storage and computation [145]. However, careful validation has to be performed on off-chain activities to maintain trust.

Security Vulnerabilities: The nature of the operation of SCs make them vulnerable to various security issues that malicious miners may exploit [135,146]. Examples include timestamp dependence, where miners can manipulate timestamps of blocks for their benefits, or mishandled exceptions where SCs call one another, and failure of a contract to return a value to a contract that called it can lead to security vulnerabilities.

Trust of Smart Contract Programmer: A key issue in the adoption of SCs is the need to rely on a technical expert (programmer) that will handle the implementation of the smart contract, that is turning what parties have agreed on in the contract into machine-readable code. However, programmers may not be able to fully capture the details of legal documents which can lead to the development of a smart contract with terms that parties did not originally agree to. Additionally, the reliance on a third party to write the smart contract also comes with the threat of malicious activities in the implementation [147].

The Requirement of Deterministic Results: In a sizeable permissionless BC, two miners may simultaneously solve two different puzzles and propagate two new potential blocks to the network nodes. The nodes will receive different blocks, thus creating two concurrent chains. These are referred to as forks. The occurrence of forks in the chain is a form of non-determinism. The deterministic aspect in the network is enforced by the consensus, which converges with certainty to a single chain. Consequently, the results of SCs need to be deterministic so that their execution yields consistent results across all nodes. Hence, SCs cannot rely on non-deterministic data and require all data passing through them to be deterministic. Thus SCs are not able to do floating-point operations due to their non-deterministic nature, which significantly limits the computational capabilities of SCs [144].

Reliance on Oracles: SCs are not able to pull data that is stored off-chain. Instead, for applications that assume that the smart contract will rely on off-chain resources, there will be a need to rely on third parties called oracles that will push the data to the BC so that it could be used by the smart contract. However, this reliance on oracles not only dilutes the advantages of decentralization of SCs, but it will also introduce an issue where oracles might face failure and go out of service [135].

Incorporating Desired Ambiguity: SCs are objective and are designed to be able to deal with every situation that they may face. However, in real textual contracts, parties may intentionally add ambiguous terms to consider unanticipated future scenarios, where a current solution is not known, and parties would want to resolve it at the time of occurrence. In the case of SCs, incorporating such desired ambiguity is simply not possible since they are executed based on pre-defined rules [136].

Lack of Standards and Regulations: BC and SCs are new technologies and lack standards and regulations. This makes their deployment for real-world usage comes with various operational challenges and high-security risks. Therefore, the regulation of smart contract usage is needed to avoid malicious activities [143,148].

7.3. Area transformation and future trends

The study of the various investigations, that attempted the integration of BC technology within RSs, enabled the identification of significant improvement opportunities and pointers to research directions that shape up important area transformations and future trends. In Table 6, a generalized set of pointers to future work directions is listed with mapping to their relevant references. Indeed, the exploration of the use of BC technology within RSs is rapidly growing. Many investigations called for improving validation, performance, analysis, evaluation, scalability, and security. Other important pointers included recommending the investigation of how to improve trust [6,123], control over personal data [124], and the integration of modern ML techniques [118].

Also, among the identified limitations, performance issues, and the reliance on oracles seem as exciting research directions. Indeed, RSs tend to handle a large amount of user-related data that need to be processed to extract user preferences and predict user needs and expectations. This processing needs to be performed relatively quickly to react to user interaction with the proposed items and reflect new market trends. Handling large amounts of data, and performing intensive processing, are not strong points of BC. Using external aid, by using oracles service, helps in alleviating such a burden. However, relying on oracles breaks the trust assumptions of the BC as oracles' activities are not subject to the consensus mechanism of the BC.

8. Conclusion

This paper investigates the adoption of Blockchain technology in recommender systems. We have surveyed state-of-the-art RSs surveys. The goal was to identify RSs limitations and open issues, and study their integration with the BC. After a thorough study of relevant BC-based RS papers, we have synthesized a taxonomy of these systems. We have also proposed a modular architecture that allows an efficient on-chain and off-chain activities combination. The proposed architecture allows a tradeoff between the benefits of the BC and its limitations, and forms a foundation for further development of BC-based RS systems. We have also included a discussion on the validity of the proposed architecture, a description of SCs limitations, and identified several improvement opportunities that require further investigation.

Acronyms

AI	Artificial Intelligence.
B2B	Business-to-Business.
B2C	Business-to-Consumer.
BC	Blockchain.
C2C	Consumer-to-Consumer.
CB	Content-Based.
CEO	Chief Executive Officer.
CF	Collaborative Filtering.
CTO	Chief Technology Officer.
DBF	Demographic-Based Filtering.
DLT	Distributed Ledger Technology.
DoS	Denial of Service.
GBF	Graph-Based Filtering.
HF	Hybrid Filtering.
IoT	Internet of Things.
IPFS	InterPlanetary File System.
KBF	Knowledge-Based Filtering.
LDA	Latent Dirichlet Allocation.
LSH	Locality Sensitive Hashing.
LSTM	Long Short Term Memory.
LTR	Learning to Rank.
MAE	Mean Absolute Error.
MF	Matrix Factorization.
ML	Machine Learning.
NPRE	Ninetieth Percentile Relative Error.
P2P	Peer-to-Peer.
POI	Point of Interest.
PoS	Proof of Stake.
PoW	Proof of Work.
QoE	Quality of Experience.
QoS	Quality of Service.
RS	Recommender System.
SC	Smart Contract.
SVD	Singular Value Decomposition.
TF	Tensor Factorization.

CRediT authorship contribution statement

Loubna Mekouar: Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition. **Youssef Iraqi:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Validation, Supervision. **Issam Damaj:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Resources, Visualization, Project administration. **Tarek Naous:** Investigation, Writing – original draft, Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by Zayed University, United Arab Emirates, under the startup grant award R20055.

Table 6
Mapping (✓) of pointers to future work to the investigations on BC-based RSs.

Reference and Year Pointers to Future Directions	[8] 2022	[7] 2021	[5] 2020	[117] 2020	[118] 2020	[130] 2020	[128,129] 2019, 2020	[125] 2019	[124] 2019	[131] 2019	[133] 2019	[134] 2019	[6,123] 2016, 2016
Improve validation					✓	✓	✓			✓	✓		✓
Improve performance	✓	✓		✓	✓			✓	✓		✓	✓	✓
Perform thorough analysis/evaluation			✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Improve scalability	✓				✓	✓			✓	✓	✓	✓	✓
Strengthen security	✓	✓					✓		✓		✓		✓
Support customization									✓				
Improve access control	✓								✓				✓
Reduce algorithmic complexity			✓	✓				✓					

References

[1] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, F. Xia, Scientific paper recommendation: A survey, IEEE Access 7 (2019) 9324–9339, http://dx.doi.org/10.1109/ACCESS.2018.2890388.

[2] P.M. Alamdari, N.J. Navimipour, M. Hosseinzadeh, A.A. Safaei, A. Darwesh, A systematic study on the recommender systems in the E-commerce, IEEE Access 8 (2020) 115694–115716, http://dx.doi.org/10.1109/ACCESS.2020.3002803.

[3] C. Sun, H. Li, X. Li, J. Wen, Q. Xiong, W. Zhou, Convergence of recommender systems and edge computing: A comprehensive survey, IEEE Access 8 (2020) 47118–47132, http://dx.doi.org/10.1109/ACCESS.2020.2978896.

[4] E. Bellini, Y. Iraqi, E. Damiani, Blockchain-based distributed trust and reputation management systems: A survey, IEEE Access 8 (2020) 21127–21151, http://dx.doi.org/10.1109/ACCESS.2020.2969820.

[5] T.-Y. Yeh, R. Kashef, Trust-based collaborative filtering recommendation systems on the blockchain, Adv. Internet Things 10 (4) (2020) 37–56.

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

[7] T.A. Abduljabbar, X. Tao, J. Zhang, X. Zhou, L. Li, Y. Cai, A survey of privacy solutions using blockchain for recommender systems: Current status, classification and open issues, Comput. J. 64 (7) (2021) 1104–1129, http://dx.doi.org/10.1093/comjnl/bxab065, arXiv:https://academic.oup.com/comjnl/article-pdf/64/7/1104/40979444/bxab065.pdf.

[8] Y. Himeur, A. Sayed, A. Alsalemi, F. Bensaali, A. Amira, I. Varlamis, M. Eirinaki, C. Sardianos, G. Dimitrakopoulos, Blockchain-based recommender systems: Applications, challenges and future opportunities, Comp. Sci. Rev. 43 (2022) 100439, http://dx.doi.org/10.1016/j.cosrev.2021.100439, URL https://www.sciencedirect.com/science/article/pii/S1574013721000769.

[9] R. Chen, Q. Hua, Y. Chang, B. Wang, L. Zhang, X. Kong, A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks, IEEE Access 6 (2018) 64301–64320, http://dx.doi.org/10.1109/ACCESS.2018.2877208.

[10] M.G. Campana, F. Delmastro, Recommender systems for online and mobile social networks: A survey, Online Soc. Netw. Media 3–4 (2017) 75–97, http://dx.doi.org/10.1016/j.osnem.2017.10.005.

[11] Z. Ding, X. Li, C. Jiang, M. Zhou, Objectives and state-of-the-art of location-based social network recommender systems, ACM Comput. Surv. 51 (1) (2018) http://dx.doi.org/10.1145/3154526.

[12] X. Yang, Y. Guo, Y. Liu, H. Steck, A survey of collaborative filtering based social recommender systems, Comput. Commun. 41 (2014) 1–10, http://dx.doi.org/10.1016/j.comcom.2013.06.009, URL https://www.sciencedirect.com/science/article/pii/S0140366413001722.

[13] J. Liu, M. Tang, Z. Zheng, X. Liu, S. Lyu, Location-aware and personalized collaborative filtering for web service recommendation, IEEE Trans. Serv. Comput. 9 (5) (2016) 686–699, http://dx.doi.org/10.1109/TSC.2015.2433251.

[14] L. Quijano-Sánchez, I. Cantador, M.E. Cortés-Cediel, O. Gil, Recommender systems for smart cities, Inf. Syst. 92 (2020) 101545, http://dx.doi.org/10.1016/j.is.2020.101545.

[15] F. Aznoli, N.J. Navimipour, Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions, J. Netw. Comput. Appl. 77 (2017) 73–86, http://dx.doi.org/10.1016/j.jnca.2016.10.009.

[16] P. Álvarez, F. Zarazaga-Soria, S. Baldassarri, Mobile music recommendations for runners based on location and emotions: The DJ-running system, Pervasive Mob. Comput. 67 (2020) 101242, http://dx.doi.org/10.1016/j.pmcj.2020.101242.

[17] R. Mu, A survey of recommender systems based on deep learning, IEEE Access 6 (2018) 69009–69022, http://dx.doi.org/10.1109/ACCESS.2018.2880197.

[18] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (1) (2019) http://dx.doi.org/10.1145/3285029, URL DOI: 10.1145/3285029.

[19] S.M. Al-Ghuribi, S.A. Mohd Noah, Multi-criteria review-based recommender system–the state of the art, IEEE Access 7 (2019) 169446–169468, http://dx.doi.org/10.1109/ACCESS.2019.2954861.

[20] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, M. Salehi, Evaluating collaborative filtering recommender algorithms: A survey, IEEE Access 6 (2018) 74003–74024, http://dx.doi.org/10.1109/ACCESS.2018.2883742.

[21] E. Karydi, K. Margaritis, Parallel and distributed collaborative filtering: A survey, ACM Comput. Surv. 49 (2) (2016) http://dx.doi.org/10.1145/2951952.

[22] M. Quadrana, P. Cremonesi, D. Jannach, Sequence-aware recommender systems, ACM Comput. Surv. 51 (4) (2018) http://dx.doi.org/10.1145/3190616.

[23] S. Raza, C. Ding, Progress in context-aware recommender systems — An overview, Comp. Sci. Rev. 31 (2019) 84–97, http://dx.doi.org/10.1016/j.cosrev.2019.01.001.

[24] I. Portugal, P. Alencar, D. Cowan, The use of machine learning algorithms in recommender systems: A systematic review, Expert Syst. Appl. 97 (2018) 205–227, http://dx.doi.org/10.1016/j.eswa.2017.12.020.

[25] F. Ricci, L. Rokach, B. Shapira, Recommender systems: introduction and challenges, in: Recommender Systems Handbook, Springer, 2015, pp. 1–34.

[26] W. Carrer-Neto, M.L. Hernández-Alcaraz, R. Valencia-García, F. García-Sánchez, Social knowledge-based recommender system. Application to the movies domain, Expert Syst. Appl. 39 (12) (2012) 10990–11000.

[27] H. Wen, L. Fang, L. Guan, A hybrid approach for personalized recommendation of news on the web, Expert Syst. Appl. 39 (5) (2012) 5806–5814.

[28] A. Nanopoulos, D. Rafailidis, P. Symeonidis, Y. Manolopoulos, MusicBox: Personalized music recommendation based on cubic analysis of social tags, IEEE Trans. Audio Speech Lang. Process. 18 (2) (2010) 407–412.

[29] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.

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

[31] Z. Yang, B. Wu, K. Zheng, X. Wang, L. Lei, A survey of collaborative filtering-based recommender systems for mobile internet applications, IEEE Access 4 (2016) 3273–3287.

[32] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, in: Recommender Systems Handbook, Springer, 2011, pp. 73–105.

[33] S. Aciar, D. Zhang, S. Simoff, J. Debenham, Informed recommender: Basing recommendations on consumer product reviews, IEEE Intell. Syst. 22 (3) (2007) 39–47.

[34] Z. Khan, N. Iltaf, H. Afzal, H. Abbas, DST-HRS: A topic driven hybrid recommender system based on deep semantics, Comput. Commun. 156 (2020) 183–191, http://dx.doi.org/10.1016/j.comcom.2020.02.068, URL https://www.sciencedirect.com/science/article/pii/S0140366419304062.

[35] E. Aslanian, M. Radmanesh, M. Jalili, Hybrid recommender systems based on content feature relationship, IEEE Trans. Ind. Inf. (2016) 1, http://dx.doi.org/10.1109/TII.2016.2631138.

[36] C.C. Aggarwal, Knowledge-based recommender systems, in: Recommender Systems, Springer, 2016, pp. 167–197.

[37] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. (2020) 1.

[38] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender Systems Handbook, Springer, 2011, pp. 217–253.

[39] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, E. Duval, Context-aware recommender systems for learning: a survey and future challenges, IEEE Trans. Learn. Technol. 5 (4) (2012) 318–335.

[40] Z.D. Champiri, S.R. Shahamiri, S.S.B. Salim, A systematic review of scholar context-aware recommender systems, Expert Syst. Appl. 42 (3) (2015) 1743–1758.

[41] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: A technical survey on decentralized digital currencies, IEEE Commun. Surv. Tutor. 18 (3) (2016) 2084–2123.

[42] M. Du, Q. Chen, J. Xiao, H. Yang, X. Ma, Supply chain finance innovation using blockchain, IEEE Trans. Eng. Manag. 67 (4) (2020) 1045–1058.

[43] L. Ismail, H. Materwala, S. Zeadally, Lightweight blockchain for healthcare, IEEE Access 7 (2019) 149935–149951.

[44] S.E. Chang, Y. Chen, When blockchain meets supply chain: A systematic literature review on current development and potential applications, IEEE Access 8 (2020) 62478–62494.
[45] J. Al-Jaroodi, N. Mohamed, Blockchain in industries: A survey, IEEE Access 7 (2019) 36500–36515.
[46] T. Alladi, V. Chamola, R.M. Parizi, K.-K.R. Choo, Blockchain applications for industry 4.0 and industrial IoT: A review, IEEE Access 7 (2019) 176935–176951.
[47] A.I. Sanka, M. Irfan, I. Huang, R.C. Cheung, A survey of breakthrough in blockchain technology: Adoptions, applications, challenges and future research, Comput. Commun. 169 (2021) 179–201, http://dx.doi.org/10.1016/j.comcom.2020.12.028, URL https://www.sciencedirect.com/science/article/pii/S0140366421000268.
[48] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, C. Rong, A comprehensive survey of blockchain: From theory to IoT applications and beyond, IEEE Internet Things J. 6 (5) (2019) 8114–8154.
[49] W. Wang, D.T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, D.I. Kim, A survey on consensus mechanisms and mining strategy management in blockchain networks, IEEE Access 7 (2019) 22328–22370.
[50] A.A. Monrat, O. Schelén, K. Andersson, A survey of blockchain from the perspectives of applications, challenges, and opportunities, IEEE Access 7 (2019) 117134–117151.
[51] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, G. Das, Everything you wanted to know about the blockchain: Its promise, components, processes, and problems, IEEE Consum. Electr. Mag. 7 (4) (2018) 6–14.
[52] J. Abou Jaoude, R.G. Saade, Blockchain applications–usage in different domains, IEEE Access 7 (2019) 45360–45381.
[53] M.S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the internet of things: A comprehensive survey, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1676–1717.
[54] J. Xie, H. Tang, T. Huang, F.R. Yu, R. Xie, J. Liu, Y. Liu, A survey of blockchain technology applied to smart cities: Research issues and challenges, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2794–2830.
[55] Y. Deldjoo, T.D. Noia, F.A. Merra, A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks, ACM Comput. Surv. 54 (2) (2021) http://dx.doi.org/10.1145/3439729.
[56] F. Fkih, Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison, J. King Saud Univ. Comput. Inf. Sci. (2021) http://dx.doi.org/10.1016/j.jksuci.2021.09.014, URL https://www.sciencedirect.com/science/article/pii/S1319157821002652.
[57] M. Elahi, F. Ricci, N. Rubens, A survey of active learning in collaborative filtering recommender systems, Comp. Sci. Rev. 20 (2016) 29–50, http://dx.doi.org/10.1016/j.cosrev.2016.05.002.
[58] GroupLens, MovieLens, 2021, https://grouplens.org/datasets/movielens/. (Accessed 25 April 2021).
[59] UF, Book-crossing dataset, 2021, http://www2.informatik.uni-freiburg.de/~cziegler/BX/. (Accessed 25 April 2021).
[60] M.S. Dataset, The last.fm dataset, 2021, http://millionsongdataset.com/lastfm/. (Accessed 25 April 2021).
[61] Delicious, Delicious: Dataset generated from the delicious site, 2021, https://rdrr.io/cran/mldr.datasets/man/delicious.html. (Accessed 25 April 2021).
[62] SNAP, SNAP, Network datasets, epinions social network, 2021, https://snap.stanford.edu/data/soc-Epinions1.html. (Accessed 25 April 2021).
[63] Flixster, Flixster, social networks, network data repository, 2021, http://networkrepository.com/soc-flixster.php. (Accessed 25 April 2021).
[64] Netflix, Watch TV shows online, watch movies online, 2021, https://www.netflix.com. (Accessed 21 August 2021).
[65] AWS, Registry of open data on AWS, 2021, https://registry.opendata.aws/. (Accessed 25 April 2021).
[66] Jester, Jester joke dataset, GroupLens, 2021, https://grouplens.org/datasets/jester/. (Accessed 25 April 2021).
[67] GroupLens, EachMovie, GroupLens, 2021, https://grouplens.org/datasets/eachmovie/. (Accessed 25 April 2021).
[68] Yahoo, Webscope, yahoo labs, 2021, https://tinyurl.com/wp7smz4z. (Accessed 25 April 2021).
[69] Datahub, Audioscrobbler data, datasets, the datahub, 2021, https://old.datahub.io/dataset/audioscrobbler. (Accessed 25 April 2021).
[70] Places, Places database, 2021, https://developer.foursquare.com/docs/places-database. (Accessed 25 April 2021).
[71] Facebook, Social network, 2021, https://www.facebook.com/. (Accessed 21 August 2021).
[72] IMDB, IMDB, 2021, https://www.imdb.com/interfaces/. (Accessed 25 April 2021).
[73] Wikipedia, List of datasets for machine-learning research, 2021, https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research. (Accessed 25 April 2021).
[74] SNAP, SNAP, Network datasets, Gowalla, 2021, https://snap.stanford.edu/data/loc-gowalla.html. (Accessed 25 April 2021).
[75] SNAP, SNAP, Network datasets, flickr image relationships, 2021, https://snap.stanford.edu/data/web-flickr.html. (Accessed 25 April 2021).
[76] Metacritic, Metacritic, movie reviews, TV reviews, game reviews, and music reviews, 2021, https://www.metacritic.com/. (Accessed 09 May 2021).
[77] R. Tomatoes, Rotten tomatoes: Movies, TV shows, movie trailers, reviews, 2021, https://www.rottentomatoes.com/. (Accessed 09 May 2021).
[78] YELP, YELP Dataset, 2021, https://www.yelp.com/dataset. (Accessed 25 April 2021).
[79] Google Research, Datasets, google research, 2021, https://research.google/tools/datasets/. (Accessed 25 April 2021).
[80] Spotify, Datasets, spotify research, 2021, https://research.atspotify.com/datasets/. (Accessed 25 April 2021).
[81] Microsoft, Geolife GPS trajectories, 2021, https://www.microsoft.com/en-us/download/details.aspx?id=52367. (Accessed 21 August 2021).
[82] GitHub, GitHub: Collaborative topic regression with social regularization, 2021, https://github.com/js05212/citeulike-a. (Accessed 21 August 2021).
[83] WS-DREAM, WS-DREAM: Towards open datasets and source code for web service research, 2021, https://wsdream.github.io/. (Accessed 21 August 2021).
[84] Stack Exchange, Stack exchange data, 2021, https://archive.org/details/stackexchange. (Accessed 21 August 2021).
[85] GitHub, GitHub: Tencent, largest multi-label image database, 2021, https://github.com/Tencent/tencent-ml-images. (Accessed 21 August 2021).
[86] Stanford, Stanford large network dataset collection, 2021, https://snap.stanford.edu/data/. (Accessed 21 August 2021).
[87] YFCC100m, YFCC100 m dataset browser, 2021, http://projects.dfki.uni-kl.de/yfcc100m/. (Accessed 21 August 2021).
[88] WEFI, WEFI, 2021, https://www.wefi.com/. (Accessed 21 August 2021).
[89] YouTube, YouTube-8M: A large and diverse labeled video dataset for video understanding research, 2021, https://research.google.com/youtube8m/. (Accessed 21 August 2021).
[90] HetRec, HetRec 2011 dataset, GroupLens, 2021, https://grouplens.org/datasets/hetrec-2011/. (Accessed 21 August 2021).
[91] myPersonality, Mypersonality database, the psychometrics centre, 2021, https://www.psychometrics.cam.ac.uk/productsservices/mypersonality. (Accessed 21 August 2021).
[92] M. Braunhofer, M. Elahi, F. Ricci, Context-aware dataset: STS - south tyrol suggests mobile app data, 2021, https://www.researchgate.net/publication/305682479_Context-Aware_Dataset_STS_-_South_Tyrol_Suggests_Mobile_App_Data. (Accessed 21 August 2021).
[93] WHRRL, WHRRL: Loan against warehouse receipt on blockchain, 2021, https://www.whr.loans/. (Accessed 21 August 2021).
[94] ACM, The million song dataset challenge, proceedings of the 21st international conference on world wide web, 2021, https://dl.acm.org/doi/10.1145/2187980.2188222. (Accessed 09 May 2021).
[95] TIVO, Data & advertising, 2021, https://business.tivo.com/products-solutions/data-and-advertising. (Accessed 21 August 2021).
[96] Orkut, The orkut dataset, 2021, https://snap.stanford.edu/data/com-Orkut.html. (Accessed 21 August 2021).
[97] UCI, UCI machine learning repository, NIPS conference papers 1987–2015 data set, 2021, https://archive.ics.uci.edu/ml/datasets/NIPS+Conference+Papers+1987-2015. (Accessed 21 August 2021).
[98] Dianping, Dianping data sets, 2021, https://lihui.info/data/dianping/. (Accessed 21 August 2021).
[99] Walmart, Dataset, 2021, https://relational.fit.cvut.cz/dataset/Walmart. (Accessed 21 August 2021).
[100] ACM, ACM recommender systems, 2021, https://recsys.acm.org/. (Accessed 09 May 2021).
[101] Retailrocket, Retailrocket recommender system dataset, kaggle, 2021, https://www.kaggle.com/retailrocket/ecommerce-dataset. (Accessed 21 August 2021).
[102] Microsoft, Microsoft research open data, 2021, https://msropendata.com/. (Accessed 21 August 2021).
[103] UCI, UCI machine learning repository, msnbc.com anonymous web data set, 2021, https://archive.ics.uci.edu/ml/datasets/msnbc.com+anonymous+web+data. (Accessed 21 August 2021).
[104] Caltech, Caltech pedestrian detection benchmark, 2021, http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/#:~:text=The%20Caltech%20Pedestrian%20Dataset%20consists,2300%20unique%20pedestrians%20were%20annotated. (Accessed 21 August 2021).
[105] Kaggle, Outbrain click prediction, kaggle, 2021, https://www.kaggle.com/c/outbrain-click-prediction/data. (Accessed 21 August 2021).
[106] Kaggle, AOL user session collection 500K, Kaggle, 2021, https://www.kaggle.com/dineshydv/aol-user-session-collection-500k. (Accessed 21 August 2021).
[107] Reclab, Reclab dataset, 2021, http://reclab.idi.ntnu.no/dataset/. (Accessed 21 August 2021).
[108] GitHub, GitHub: A live movie rating dataset collected from Twitter, 2021, https://github.com/sidooms/MovieTweetings. (Accessed 21 August 2021).
[109] D.H. Park, Data set for mobile app retrieval, 2021, https://sites.google.com/site/daehpark/Resources/data-set-for-mobile-app-retrieval. (Accessed 21 August 2021).
[110] GitHub, GitHub: factorization, 2021, https://github.com/hexiangnan/neural_factorization_machine/tree/master/data/frappe. (Accessed 21 August 2021).
[111] START, Global terrorism database, 2021, https://www.start.umd.edu/gtd/. (Accessed 21 August 2021).

[112] RecSys Lab, Datasets: RecSys lab, 2021, http://recsys.deib.polimi.it/datasets/, (Accessed 21 August 2021).
[113] UCI, UCI machine learning repository, entree chicago recommendation data set, 2021, https://archive.ics.uci.edu/ml/datasets/Entree+Chicago+Recommendation+Data. (Accessed 21 August 2021).
[114] CiteSeer, Network data repository, 2021, http://networkrepository.com/citeseer.php. (Accessed 21 August 2021).
[115] ACM, Datasets, 2021, https://dl.acm.org/artifacts/dataset. (Accessed 21 August 2021).
[116] DBLP, DBLP: How can I download the whole DBLP dataset? 2021, https://dblp.org/faq/How+can+I+download+the+whole+dblp+dataset.html. (Accessed 21 August 2021).
[117] S.B. Patel, P. Bhattacharya, S. Tanwar, N. Kumar, KiRTi: A blockchain-based credit recommender system for financial institutions, IEEE Trans. Netw. Sci. Eng. (2020).
[118] K. Abbas, M. Afaq, T. Ahmed Khan, W.-C. Song, A blockchain and machine learning-based drug supply chain management and recommendation system for smart pharmaceutical industry, Electronics 9 (5) (2020) 852.
[119] Z. Zheng, Y. Zhang, M.R. Lyu, Distributed QoS evaluation for real-world web services, in: 2010 IEEE International Conference on Web Services, IEEE, 2010, pp. 83–90.
[120] Recommender systems and applications in banking | by aymeric bouvier | genify | medium, 2021, https://medium.com/genifyai/recommender-systems-and-applications-in-banking-f0cef8f87fa6. (Accessed 23 November 2021).
[121] 81 Of top 100 companies use blockchain technology, blockdata research shows, 2021, https://forkast.news/81-of-top-100-companies-use-blockchain-technology-blockdata/. (Accessed 23 November 2021).
[122] N. Newman, Apple ibeacon technology briefing, J. Direct Data Digit. Mark. Pract. 15 (2014) http://dx.doi.org/10.1057/dddmp.2014.7.
[123] R. Frey, D. Wörner, A. Ilic, Collaborative filtering on the blockchain: a secure recommender system for e-commerce, in: Proc. 22nd Amer. Conf. Inform. Syst, 2016, pp. 1–5.
[124] A. Lisi, A. De Salve, P. Mori, L. Ricci, A smart contract based recommender system, in: International Conference on the Economics of Grids, Clouds, Systems, and Services, Springer, 2019, pp. 29–42.
[125] X. Li, E. Du, C. Chen, Z. Zheng, T. Cai, Q. Yan, Blockchain-based credible and privacy-preserving QoS-Aware web service recommendation, in: International Conference on Blockchain and Trustworthy Systems, Springer, 2019, pp. 621–635.
[126] J.D. Harris, B. Waggoner, Decentralized and collaborative AI on blockchain, in: 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, 2019, pp. 368–375.
[127] J.D. Harris, Analysis of models for decentralized and collaborative AI on blockchain, in: International Conference on Blockchain, Springer, 2020, pp. 142–153.
[128] A. Al Omar, R. Bosri, M.S. Rahman, N. Begum, M.Z.A. Bhuiyan, Towards privacy-preserving recommender system with blockchains, in: International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications, Springer, 2019, pp. 106–118.
[129] R. Bosri, M.S. Rahman, M.Z.A. Bhuiyan, A. Al Omar, Integrating blockchain with artificial intelligence for privacy-preserving in recommender systems, IEEE Trans. Netw. Sci. Eng. (2020).
[130] B. Yan, J. Yu, Y. Wang, Q. Guo, B. Chai, S. Liu, Blockchain-based service recommendation supporting data sharing, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, 2020, pp. 580–589.
[131] F. Casino, C. Patsakis, An efficient blockchain-based privacy-preserving collaborative filtering architecture, IEEE Trans. Eng. Manag. (2019).
[132] L. Lin, Y. Tian, Y. Liu, A blockchain-based privacy-preserving recommendation mechanism, in: 2021 IEEE 5th International Conference on Cryptography, Security and Privacy, CSP, 2021, pp. 74–78, http://dx.doi.org/10.1109/CSP51677.2021.9357604.
[133] S. Wang, C. Huang, J. Li, Y. Yuan, F.-Y. Wang, Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts, IEEE Access 7 (2019) 136951–136961.
[134] W. Cai, X. Du, J. Xu, A personalized QoS prediction method for web services via blockchain-based matrix factorization, Sensors 19 (12) (2019) 2749.
[135] A. Beniiche, A study of blockchain oracles, 2020, arXiv preprint arXiv:2004.07140.
[136] S.D. Levi, A.B. Lipton, An introduction to smart contracts and their potential and inherent limitations, in: Harvard Law School Forum on Corporate Governance & Financial Regulation, 2018.
[137] VISA, VISA Fact sheet, 2021, https://usa.visa.com/dam/VCOM/download/corporate/media/visanet-technology/aboutvisafactsheet.pdf. (Accessed 21 August 2021).
[138] Bitcoin News, No, visa doesn't handle 24,000 TPS and neither does your pet blockchain, 2021, https://news.bitcoin.com/no-visa-doesnt-handle-24000-tps-and-neither-does-your-pet-blockchain/. (Accessed 21 August 2021).
[139] A. Singh, R.M. Parizi, Q. Zhang, K.-K.R. Choo, A. Dehghantanha, Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities, Comput. Secur. 88 (2020) 101654.
[140] Y. Hanada, L. Hsiao, P. Levis, Smart contracts for machine-to-machine communication: Possibilities and limitations, in: 2018 IEEE International Conference on Internet Of Things and Intelligence System, IOTAIS, IEEE, 2018, pp. 130–136.
[141] S.S. Hazari, Q.H. Mahmoud, A parallel proof of work to improve transaction speed and scalability in blockchain systems, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC, IEEE, 2019, pp. 0916–0921.
[142] M. Vukolić, The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication, in: International Workshop on Open Problems in Network Security, Springer, 2015, pp. 112–125.
[143] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.-Y. Wang, Blockchain-enabled smart contracts: architecture, applications, and future trends, IEEE Trans. Syst. Man Cybern Syst. 49 (11) (2019) 2266–2277.
[144] A. Battah, Y. Iraqi, E. Damiani, Blockchain-based reputation systems: Implementation challenges and mitigation, Electronics 10 (3) (2021) http://dx.doi.org/10.3390/electronics10030289, URL https://www.mdpi.com/2079-9292/10/3/289.
[145] J. Eberhardt, S. Tai, On or off the blockchain? Insights on off-chaining computation and data, in: European Conference on Service-Oriented and Cloud Computing, Springer, 2017, pp. 3–15.
[146] A. Dika, Ethereum smart contracts: Security vulnerabilities and security tools, 2017, NTNU.
[147] C. Sillaber, B. Waltl, Life cycle of smart contracts in blockchain ecosystems, Datenschutz Und Datensicherheit-DuD 41 (8) (2017) 497–500.
[148] E. Mik, Smart contracts: terminology, technical limitations and real world complexity, Law Innov. Technol. 9 (2) (2017) 269–300.