Web3Recommend

Decentralised Web3 social recommendations with trust and relevance balance

Rohan Madhwal
Delft University of Technology
Delft, The Netherlands
R.Madhwal@student.tudelft.nl

Johan Pouwelse
Delft University of Technology
Delft, The Netherlands
J.A.Pouwelse@tudelft.nl

***Abstract*—Web3 is an emerging decentralised alternative to the conventional, centrally governed model of the Internet. By leveraging decentralised protocols on shared infrastructure, Web3 is able to facilitate trusted interactions between strangers on the internet, without the need for profit-driven trusted intermediaries. However, the proliferation of massive amounts of data in Web3 platforms, or the "information overload problem" makes it increasingly hard for users to find content that they like or find useful. In the centralised model of the internet "Social Recommender Systems" are used to solve this problem and hence, increase user engagement. Generating recommendations in decentralised networks is a non-trivial problem because of the relative ease with which a single malicious user can generate multiple fake or "sybil" identities, allowing them to unfairly manipulate the ranking of items, thus leading to low quality, untrustworthy recommendations. Further, the spam created by sybil identities places a large burden on Web3 platforms due to resource wastage, potentially further degrading user experience. In this paper, we present Web3Recommend, a novel distributed, social recommendation system that balances trust and relevance by combining a graph-based content recommendation algorithm inspired by systems used in Twitter with MeritRank, a decentralised reputation scheme. Our systems ranks items using a personalised SALSA algorithm which involves Monte Carlo estimation of recommendations through random walks performed in a sybil-resistant manner. Our system's design relies on decay parameters which balance the trustworthiness (sybil-resistance) and relevance of the generated recommendations. In this paper, we present an end-to-end implementation of Web3Recommend which can be plugged into any Web3 platform. As a proof-of-concept, we integrate our system with MusicDAO, an open-source Web3 music sharing platform to generate personalised, real-time recommendations. Further, our experiments demonstrate the trust-relevance balance of recommendations against multiple adversarial strategies and feasibility of the system in large-scale systems.**

I. Introduction

The cardinal objective of a social media platform that aims to be successful and vibrant is an active and engaged user base. Achieving user engagement boils down to presenting the most attractive and relevant content to each user. However, popularity and success is a double edged sword since the abundance of users and content on these platforms floods users with huge amounts of information and hence poses a great challenge in terms of information overload. While search capabilities slightly alleviate the problem, often users are unable to express keywords that convey requirements about the type of content they would be interested in. Further, users tend to have diverse taste and the quality of content may be subjective depending on the user searching for it, therefore, beyond simple searching capabilities, personalisation is also required to make the content attractive and relevant to each user.

A Social Recommender System is an intelligent system that filters the massive amounts of information on social media platforms and recommends useful items and information to users based on their personalised needs which are inferred through unique explicit and implicit interactions within the social network. In this way, Social Networks and their Recommender Systems tend to have a symbiotic relationship since the quality of recommendations catered to users allows the networks to grow in size, which in turn provides more interactions, allowing higher quality recommendations.

GraphJet is an example of a graph based Social Recommender system used for generating content recommendations in Twitter. GraphJet is able to provide personalized, real-time content recommendations for Twitter users, i.e. for a given user, it is able to recommend tweets that the user may be interested in based on the user's history and social interactions. To serve these recommendations, a personalized SALSA algorithm is run on a bi-partite graph of interactions between users and tweets. The system assumes that the entire graph can be stored in the memory of a single server.

While GraphJet and similar existing Social Recommender systems rely on cenralised servers to serve recommendations to users, Web3 platforms involve direct interactions between users without any third-party intermediaries. Thus, a Social Recommender for a Web3 platform requires a significantly different design from existing systems in order to cater recommendations to its users.

In Web3 platforms, the generation of recommendations also faces unique challenges compared to traditional centralized Internet models in the form of sybil attacks, which undermine the trustworthiness of recommendations. In II, we further explain the sybil attack and the problem of generating trusted

recommendations in Web3 platforms.

-Insert Text About Merit Rank here-

In this paper, we address the challenges associated with generating recommendations in Web3 platforms by presenting Web3Recommend, a novel distributed, social recommendation system. Our approach integrates a graph-based content recommendation algorithm inspired by systems used in Twitter with MeritRank, a decentralized reputation scheme. By leveraging personalized SALSA algorithm with sybil-resistant random walks, we aim to strike a balance between trustworthiness and relevance in recommendations.

Web3Recommend contributes to the existing body of research by providing an end-to-end implementation that can be seamlessly integrated into any Web3 platform. Additionally, we demonstrate the trust-relevance balance of recommendations and the feasibility of our system through experiments conducted in large-scale Web3 environments.

The following are the key features and main contributions of Web3Recommend:

1) **Based on Random Walks**
   Web3Recommend uses personalized ego-centric random walks to perform monte carlo estimations of recommendations in the system. By enhancing these random walks with decay parameters from MeritRank Web3Recommend is able to provide a balance of trust and relevance in its recommendations. Random Walks allow us to create a system that is simple and understandable, yet also sufficiently expressive to generate relevant, trustworthy recommendatins. Further, random walks act as "social proof" for the recommendations, allowing users to better understand why certain items were recommended, leading to higher user engagement.
2) **Each node stores the entire interaction graph between users and items which is synchronized using gossiping mechanism**
   While readers might find the idea of storing the entire graph in a node strange, graph partitioning remains a complex problem and maintaining a distributed graph adds communication cost to the system. Further, in a P2P Web3 platform, users are not expected to be always available/online, thus, similar to GraphJet, Web3Recommend assumes that all nodes store the entire bi-partite graph in memory. We further demonstrate that this is a reasonable assumption by showing that upto a billion edges can be reasonably stored in less than 8GB of memory with our compact graph serialization techniques
3) **Bootstrapping mechanisms**
   Web3Recommend includes two bootstrapping mechanisms: 1) A similarity based mechanism for finding similar users, allowing new users to find users they can trust 2) A personalized page rank for creating a circle of trust which can be used for recommending relevant items to users who haven't consumed many items and thus don't have any existing edges in the interaction graph

II. Problem Description

The decentralized nature of Web3 platforms, coupled with the pseudonymity and anonymity they offer, creates an environment where sybil attacks can occur more easily. Unlike in centralized systems, where user identities are typically verified, Web3 platforms prioritize user privacy and allow users to operate under multiple pseudonyms. This makes it challenging to establish the authenticity and credibility of users and their preferences.

Sybil attacks have been extensively studied in the context of decentralized systems. Researchers have identified the adverse effects of sybil attacks on various aspects of Web3 platforms, including reputation systems, voting systems, and collaborative filtering recommender systems.

For example, in the domain of reputation systems, a P2P system may be compromised by a sybil attack as the attacker is able to favourably alter reputation scores by creating multiple forged identities which attest to each other's trustworthiness.

When it comes to recommendation systems, sybil attacks pose a significant challenge in generating trustworthy recommendations. In centralized recommender systems, traditional approaches such as collaborative filtering rely on user preferences and similarity to generate recommendations. However, in the Web3 context, the presence of sybil identities disrupts the accuracy and reliability of these techniques. Fake identities can artificially boost the rankings of specific items or flood the system with spam, leading to a deterioration in recommendation quality and user experience.

Recommender systems in centralized networks are relatively more secure from Sybil attacks due to the centralized nature of the system. Centralized systems often require user authentication and verification, making it more difficult for malicious actors to create multiple fake accounts to manipulate the ranking of items. Additionally, centralized systems have the advantage of being able to monitor user behavior and detect anomalies such as unusual activity patterns or highly repetitive actions, which could indicate the presence of a Sybil attack. This detection is possible because of the large number of skilled attendants dedicated to maintaining and improving system capabilities in centralized systems. These attendants can develop measures to prevent Sybil attacks, such as restricting the number of user actions that can be performed within a certain time frame, or introducing identity verification requirements.

In contrast, creating a trustable and reliable decentralized recommender system for Web3 platforms is a challenging task due to the lack of centralized infrastructure, making it easier for malicious users to create and control multiple identities, manipulate the ranking of items, and compromise the trustworthiness of recommendations. Therefore, the creation of decentralized recommender systems requires new approaches that can address the challenges of decentralized networks, including sybil attacks, limited resources, and lack of trust among users.

To address the problem of sybil attacks, several research efforts have proposed solutions based on decentralized repu-

tation systems, cryptographic techniques, and social network analysis. For instance, Yu et al. (2008) proposed a trust-aware recommendation algorithm that utilized a decentralized reputation mechanism to combat the influence of sybil identities. The approach incorporated trust ratings from reputable users to filter out fake or malicious recommendations.

Achieving a balance between trust and relevance in recommendations is another critical challenge in Web3 platforms. While combating sybil attacks is crucial, it is equally important to deliver recommendations that align with the user's preferences and interests. Existing research has explored the trade-off between trust and relevance in recommender systems. For example, Massa and Avesani (2007) introduced a trust-aware collaborative filtering algorithm that considered both the trustworthiness of recommendations and the personalized relevance to users.

By addressing the issues caused by sybil attacks and addressing the trade-off between trust and relevance, Web3Recommend contributes to the advancement of recommendation systems within the Web3 ecosystem.

III. Background

The system presented in this paper relies on the (incremental) computation of personalized PageRank and SALSA augmented with principles from MeritRank. We also build on top of the GraphJet recommender system by Twitter. In this section, we provide a quick review of these methods.

A. PageRank

One of the most widely known ranking systems in the world is Google's PageRank which is still used (along with other algorithms) in order to rank websites for user queries on Google. PageRank determines a rough estimate of the relative importance of a website by computing a ranking for every web page. The underlying assumption of PageRank is that a website that is more important is more likely to receive links from other websites than a website that is less important.

The calculation of PageRank of a website can be simplified to the below equation:

$$\sum \frac{PageRank\ of\ Inbound\ Link}{Number\ of\ Outgoing\ Links\ on\ that\ Page} \quad (1)$$

PageRank can also be viewed as the stationary distribution of a random walk in a graph of websites connected by hyperlinks, which at each step with probability ϵ jumps to a random node and with probability $1 - \epsilon$ jumps to a randomly chosen node through an edge from the current node. **Personalized Page Rank**

Despite PageRank being a well studied problem, making some simple assumptions on the structure of data layout and network leads to a dramatic improvement in its runtime using the Monte Carlo estimation technique.

B. SALSA

C. GraphJet

D. Sybil Attacks

E. MeritRank

F. Distributed Network and Web3

IV. System Design

Web3Recommend is a Recommender System designed to provide recommendations in any application running on a distributed network. Hence, we use a peer to peer architecture which assumes each user operates their own node and interacts with items being shared inside the network. The central data structure in the network is the **TrustNetwork** graph which stores information about node to node and node to item relationships across the entire network. Recommendations in the system are generated by performing random walks inside this network. Each node maintains a personal copy of a TrustNetwork and updates to the network are synchronized through a timestamp biased edge gossiping mechanism which ensures that recommendations are based on recent, global information inside the network. The system design also includes a simple bootstrapping mechanism which allows new users to find similar users in the network, however, it is worth noting that this bootstrap mechanism can be exploited by malicious users and in a real application, we assume that users are able to bootstrap through social discovery of trusted peers or through the provision of trustworthy nodes by the application itself. The following is an in-depth description of the various components of the system:

A. TrustNetwork

B. Recommendation Algorithm

C. Timestamp Biased Edge Gossiping

D. Bootstrap

- Graph Storage: High Level Design consists of each node storing the entire network's state in the form of two graphs: a node to node network which describes trust relationships between nodes in the network and a node to song network which describes each node's song preferences calculated as a proportion of the song's play count. These graphs are then synchronized between nodes using a gossiping mechanism. - Baseline Recommendation System: Influenced by GraphJet, runs on the node to song network in the form of a personalized SALSA, create illustration to show how it works, songs that have the most song visits receive highest ranking - Modified Trustworthy Recommendation System: Exploration probability added to the random walk, beta decays from merit rank - Custom Compact Serialization - Gossiping Mechanism - Bootstrap

V. Dataset and Experiments

- Background about what we need to prove, how the dataset was constructed - Large Sybil Attack - Similarity