

Do Servers Matter on Mastodon? Data-driven Design for Decentralized Social Media

ANONYMOUS AUTHOR(S)

When trying to join Mastodon, a decentralized collection of interoperable social networking servers, new users face the dilemma of choosing a home server. Using trace data from millions of new Mastodon accounts, we show that new accounts are less likely to remain active on the network's largest general instances compared to others. Additionally, we observe a trend of users migrating from larger to smaller servers. Addressing the challenge of onboarding and server selection, the paper proposes a decentralized recommendation system for server using hashtags and the Okapi BM25 algorithm. This system leverages servers' top hashtags and their frequency to create a recommendation mechanism that respects Mastodon's decentralized ethos. Simulations demonstrate that such a tool can be effective even with limited data on each local server.

CCS Concepts: • **Human-centered computing** → **Social networking sites**; *Collaborative and social computing*; • **Information systems** → *Retrieval models and ranking*; • **Applied computing** → *Document searching*.

Additional Key Words and Phrases: decentralized online social networks

ACM Reference Format:

Anonymous Author(s). 2018. Do Servers Matter on Mastodon? Data-driven Design for Decentralized Social Media. In . ACM, New York, NY, USA, 15 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Following Twitter's 2022 acquisition, Mastodon—an open-source, decentralized social network and microblogging community—saw an increase in activity and attention as a potential Twitter alternative [6, 15]. While millions of people set up new accounts and significantly increased the size of the network, many newcomers found the process confusing and many accounts did not remain active. Unlike centralized social media platforms, Mastodon is a network of independent servers with their own rules and norms [18]. Each server can communicate with each other using the shared ActivityPub protocols and accounts can move between Mastodon servers, but the local experience can vary widely from server to server.

Although attracting and retaining newcomers is a key challenge for online communities [14, 182], Mastodon's onboarding process has not always been straightforward. Variation among servers can also present a challenge for newcomers who may not even be aware of the specific rules, norms, or general topics of interest on the server they are joining [3]. Various guides and resources for people trying to join Mastodon offered mixed advice on choosing a server. Some suggest that the most important thing is to simply join any server and work from there [13, 30], while others have created tools and guides to help people find potential servers of interest by size and location[11, 31].

Mastodon's decentralized design has long been in tension with the disproportionate popularity of a small set of large, general-topic servers within the system [20]. Analysing the activity of new accounts that join the network, we find that users who sign up on such servers are less likely to remain active after 91 days. We also find that many users

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

who move accounts tend to gravitate toward smaller, more niche servers over time, suggesting that established users may also find additional utility from such servers.

In response to these findings, we propose a potential way to create server and tag recommendations on Mastodon. This recommendation system could both help newcomers find servers that match their interests and help established accounts discover “neighborhoods” of related servers to enable further discovery.

2 BACKGROUND

2.1 Empirical Setting

The Fediverse is a set of decentralized online social networks which interoperate using shared protocols like ActivityPub. Mastodon is a software program used by many Fediverse servers and offers a user experience similar to the Tweetdeck client for Twitter. It was first created in late 2016 and saw a surge in interest in 2022 during and after Elon Musk’s Twitter acquisition.

Mastodon features three kinds of timelines. The primary timeline is a “home” timeline which shows all posts from accounts followed by the user. Mastodon also supports a “local” timeline which shows all public posts from the local server and a “federated” timeline which includes all posts from users followed by other users on their server. The local timeline is unique to each server and can be used to discover new accounts and posts from the local community. On larger servers, this timeline can be unwieldy; however, on smaller servers, this presents the opportunity to discover new posts and users of potential interest.

Discovery has been challenging on Mastodon. Text search, for instance, was impossible on most servers until support for this feature was added on an optional, opt-in basis using Elasticsearch in late 2023 [24]. Recommendation systems are currently a somewhat novel problem in the context of decentralized online social networks. Trienes et al. [32] developed a recommendation system for finding new accounts to follow on the Fediverse which used collaborative filtering based on BM25 in an early example of a content discovery system on Mastodon.

Individual Mastodon servers can have an effect on the end experience of users. For example, some servers may choose to federate with some servers but not others, altering the topology of the Fediverse network for their users. At the same time, accounts can only map to one specific server. Because of Mastodon’s data portability, users can move their accounts freely between servers while retaining their followers, though their post history remains with their original account.

2.2 The Mastodon Migrations

Mastodon saw a surge in interest in 2022 and 2023, particularly after Elon Musk’s Twitter acquisition. In particular, four events of interests drove measurable increases in new users to the network: the announcement of the acquisition (April 14, 2022), the closing of the acquisition (October 27, 2022), a day when Twitter suspended a number of prominent journalists (December 15, 2022), and a day when Twitter experienced an outage and started rate limiting accounts (July 1, 2023). Many Twitter accounts announced they were setting up Mastodon accounts and linked their new accounts to their followers, often using tags like #TwitterMigration[6] and driving interest in Mastodon in a process La Cava et al. [15] found consistent with social influence theory.

Some media outlets have framed reports on Mastodon [9] through what Zulli et al. [36] calls the “Killer Hype Cycle”, whereby the media finds a new alternative social media platform, declares it a potential killer of some established platform, and later calls it a failure if it does not displace the existing platform. Such framing fails to take systems like

the Fediverse seriously for their own merits: completely replacing existing commercial systems is not the only way to measure success, nor does it account for the real value the Fediverse provides for its millions of active users.

Mastodon’s approach to onboarding has also changed over time. In much of 2020 and early 2021, the Mastodon developers closed sign-ups to their flagship server and linked to an alternative server, which saw increased sign-ups during this period. They also linked to a list of servers on the “Join Mastodon” webpage [17], where all servers are pre-approved and follow the Mastodon Server Covenant which guarantees certain content moderation standards and data protections. Starting in 2023, the Mastodon developers shifted toward making the flagship server the default when people sign up on the official Mastodon Android and iOS apps [25, 26].

2.3 Newcomers in Online Communities

Onboarding newcomers is an important part of the life cycle of online communities. Any community can expect a certain amount of turnover, and so it is important for the long-term health and longevity of the community to be able to bring in new members [14, 182]. However, the process of onboarding newcomers is not always straightforward.

The series of migrations of new users into Mastodon in many ways reflect folk stories of “Eternal Septembers” on previous communication networks, where a large influx of newcomers challenged the existing norms [4, 10]. Many Mastodon servers do have specific norms which people coming from Twitter may find confusing, such as local norms around content warnings [18]. Variation among servers can also present a challenge for newcomers who may not even be aware of the specific rules, norms, or general topics of interest on the server they are joining [3]. Mastodon servers open to new accounts must thus be both accommodating to newcomers while at the same ensuring the propagation of their norms and culture, either through social norms or through technical means.

2.4 Recommendation Systems and Collaborative Filtering

Recommender systems help people filter information to find resources relevant to some need [22]. The development of these systems as an area of formal study harkens back to information retrieval (e.g. Salton and McGill [27]) and foundational works imagining the role of computing in human decision-making (e.g. Bush [1]). Early work on these systems produced more effective ways of filtering and sorting documents in searches such as the probabilistic models that motivated the creation of the okapi (BM25) relevance function [23]. Many contemporary recommendation systems use collaborative filtering, a technique which produces new recommendations for items based on the preferences of a collection of similar users [12].

Collaborative filtering systems build on top of a user-item-rating ($U - I - r$) model where there is a set of users who each provide ratings for a set of items. The system then uses the ratings from other users to predict the ratings of a user for an item they have not yet rated and uses these predictions to create a ordered list of the best recommendations for the user’s needs [5, 86-87]. Collaborative filtering recommender systems typically produce better results as the number of users and items in the system increases; however, they must also deal with the “cold start” problem, where limited data makes recommendations unviable [16]. The cold start problem has three possible facets: bootstrapping new communities, dealing with new items, and handling new users [29, 311-312]. In each case, limited data on the entity makes it impossible to find similar entities without some way of building a profile. Further, uncorrected collaborative filtering techniques often also produce a bias where more broadly popular items receive more recommendations than more obscure but possibly more relevant items [35]. Research on collaborative filtering has also shown that the quality of recommendations can be improved by using a combination of user-based and item-based collaborative filtering [28].

Although all forms of collaborative filtering use some combination of users and items, there are two main approaches to collaborative filtering: memory-based and model-based. Memory-based approaches use the entire user-item matrix to make recommendations, while model-based approaches use a reduced form of the matrix to make recommendations. This is particularly useful because the matrix of items and users tends to be extremely sparse, e.g. in a movie recommender system, most people have not seen most of the movies in the database. Singular value decomposition (SVD) is one such dimension reduction technique which transforms a $m \times n$ matrix M into the form $M = U\Sigma V^T$ [19]. SVD is particularly useful for recommendation systems because it can be used to find the latent factors which underlie the user-item matrix and use these factors to make recommendations.

While researchers in the recommendation system space often focus on ways to design the system to produce good results mathematically, human-computer interaction researchers also consider various human factors which contribute to the overall system. Crucially, McNee et al. argued “being accurate is not enough”: user-centric evaluations, which consider multiple aspects of the user experience, are necessary to evaluate the full system. HCI researchers have also contributed pioneering recommender systems in practice. For example, GroupLens researchers Resnick et al. [21] created a collaborative filtering systems for Usenet and later produced advancements toward system evaluation and explanation of movie recommendations [7, 8]. Cosley et al. [2] created a system to match people with tasks on Wikipedia to encourage more editing. This prior work shows that recommender systems can be used to help users find relevant information in a variety of contexts.

2.5 Evaluation of Recommendation Systems

Evaluating recommender systems can be tricky because a measure of good performance must take into account various dimensions[34]. A measure of accuracy must be paired with a question of “accuracy toward what?” Explainability requires a transparent means of showing the user why a certain item was recommended.

It is often important to both start with an end goal in mind and to keep evaluation integrated throughout the entire process of creating a recommender systems, from conceptualization to optimization. There are several considerations to keep in mind such as the trade-off between optimizing suggestions and the risks of over-fitting. For example, a system designed to create suggestions with the highest propensity that the user will like the recommendations may struggle with a reduced diversity of its suggestions.

Recommender systems can be evaluated using three board categories of techniques: offline, online, and user studies. Offline evaluation uses pre-collected data and a measure to describe the performance of the system, assuming there is insufficient relevance to the difference in time between when the data was collected and the present moment. Online evaluation uses a deployed, live system, e.g. A/B testing. In this case, the user is often unaware of the experiment. In contrast, user studies involve subjects which are aware they are being studied.

3 DATA

Mastodon has an extensive API which allows for the collection of public posts and account information. We collected data from the public timelines of Mastodon servers using the Mastodon API with a crawler which runs once per day. We also collected account information from the opt-in public profile directories on these servers.

Mastodon Profiles: We collected accounts using data previously collected from posts on public Mastodon timelines from October 2020 to August 2023. We then queried for up-to-date information on those accounts including their most recent status and if the account had moved as of February 2024. Through this process, we discovered a total of 1,784,438 account created between August 14, 2020 and January 1, 2024. We then filtered out accounts which were bots (19,014

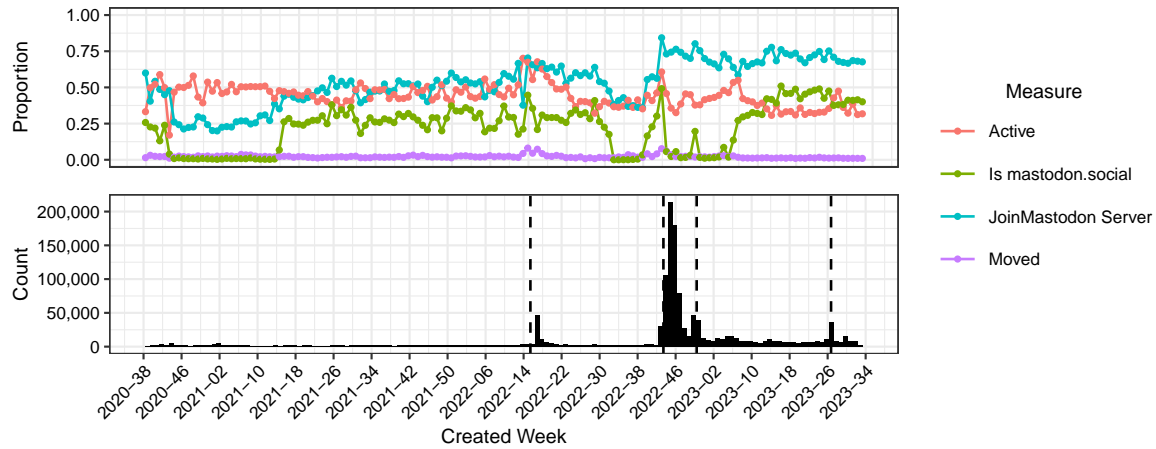


Fig. 1. Accounts in the dataset created between January 2022 and March 2023. The top panels shows the proportion of accounts still active 45 days after creation, the proportion of accounts that have moved, and the proportion of accounts that have been suspended. The bottom panel shows the count of accounts created each week. The dashed vertical lines in the bottom panel represent the announcement day of the Elon Musk Twitter acquisition, the acquisition closing day, a day where Twitter suspended a number of prominent journalist, and a day when Twitter experienced an outage and started rate limiting accounts.

accounts), had been suspended (163,721 accounts), had been marked as moved to another account (41,111 accounts), had been limited by their local server (6,062 accounts), had no statuses (140,870 accounts), or had posted their last status on the same day as their account creation (391,173 accounts). This gave us a total of 1,067,546 accounts which met all the filtering criteria. Note that because we got updated information on each account, we include only accounts on servers which still existed at the time of our profile queries and which returned records for the account.

Tags: Mastodon supports hashtags, which are user-generated metadata tags that can be added to posts. Clicking the link for a tag shows a stream of posts which also have that tag from the federated timeline, which includes accounts on the same server and posts from accounts followed by the accounts on the local server. We collected 1,206,779 statuses posted by 67,604 accounts on 723 unique servers from between May to July 2023 which contained at least one hashtag.

4 ANALYSIS AND RESULTS

4.1 Survival Model

Are accounts on suggested general servers less likely to remain active than accounts on other servers?

4.1.1 Kaplan–Meier Estimator. Using 28,258 accounts created from May 1 to June 30, 2023, we create a Kaplan–Meier estimator for the probability that an account will remain active based on whether the account is on one of the largest general instances¹ featured at the top of the Join Mastodon webpage or otherwise if it is on a server in the Join Mastodon list. Accounts are considered active if they have made at least one post after the censorship period 91 days after account creation.

4.1.2 Mixed Effects Cox Proportional Hazard Model. We also construct a Mixed Effects Cox Proportional Hazard Model

¹mastodon.social, mstdn.social, mastodon.world, mas.to, mastodon.online, mastodonapp.uk, universeodon.com, masto.ai, c.im, social.vivaldi.net, mstdn.party, ohai.social

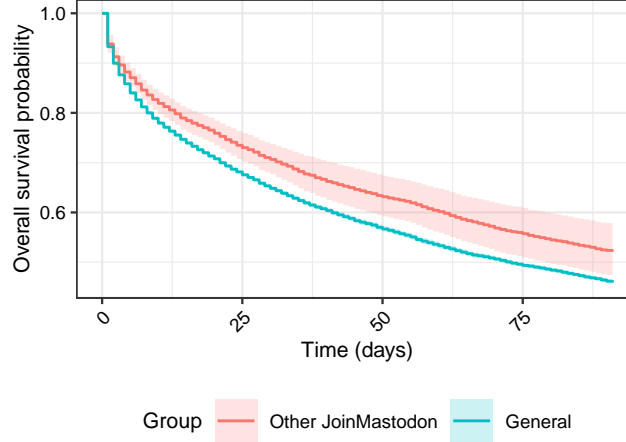


Fig. 2. Survival probabilities for accounts created during May 2023.

Table 1. Coefficients for the Cox Proportional Hazard Model with Mixed Effects. The model includes a random effect for the server.

term	estimate	exp.coef	p.value
Join Mastodon	0.115	(0.97, 1.3)	0.117
General Servers	0.385	(1.07, 2.01)	0.017
Small Server	-0.245	(0.66, 0.92)	0.003

$$h(t_{ij}) = h_0(t) \exp \begin{pmatrix} \beta_1 \text{Join Mastodon} \\ + \beta_2 \text{General Servers} \\ + \beta_3 \text{Small Server} \\ + b_j \end{pmatrix}$$

where $h(t_{ij})$ is the hazard for account i on server j at time t , $h_0(t)$ is the baseline hazard, β_1 is the coefficient for whether the account is on a server featured on Join Mastodon, β_2 is the coefficient for whether the account is on one of the largest general instances, β_3 is the coefficient for whether the account is on a small server with less than 100 accounts, and b_j is the random effect for server j .

We again find that accounts on the largest general instances are less likely to remain active than accounts on other servers, while accounts created on smaller servers are more likely to remain active.

4.1.3 Logistic Regression. First, we calculate a continuous measure for the generality of the server based on the item-item similarity between servers. We then use this measure to predict whether an account will remain active after 91 days using a logistic regression model.

The results of this analysis again suggests that the generality of the server is negatively associated with the likelihood that an account will remain active after 91 days.

	(1)
(Intercept)	−0.416 (0.092)
generality	−0.788 (0.100)
Num.Obs.	25 322
AIC	28 029.8
BIC	28 046.1
Log.Lik.	−14 012.898
RMSE	0.43

Table 2. Exponential family random graph models for account movement between Mastodon servers. Accounts in Model A were created in May 2022 and moved to another account at some later point. Accounts in Model B were created at some earlier point and moved after October 2023.

	Model A		Model B	
	Coef.	Std.Error	Coef.	Std.Error
Sum	−9.529	***0.188	−10.268	***0.718
Nonzero	−3.577	***0.083	−2.861	***0.254
Smaller server	0.709	***0.032	0.629	***0.082
Server size (outgoing)	0.686	***0.013	0.655	***0.042
Open registrations (incoming)	0.168	***0.046	−0.250	0.186
Languages match	0.044	0.065	0.589	0.392

4.2 Moved Accounts

Do accounts tend to move to larger or smaller servers?

Mastodon users can move their accounts to another server while retaining their connections (but not their posts) to other Mastodon accounts. This feature, built into the Mastodon software, offers data portability and helps avoid lock-in.

To corroborate our findings, we also use data from thousands of accounts which moved between Mastodon servers, taking advantage of the data portability of the platform. Conceiving of these moved accounts as edges within a weighted directional network where nodes represent servers, edges represent accounts, and weights represent the number of accounts that moved between servers, we construct an exponential family random graph model (ERGM) with terms for server size, open registrations, and language match between servers. We find that accounts are more likely to move from larger servers to smaller servers.

5 PROPOSED RECOMMENDATION SYSTEM

How can we build an opt-in, low-resource recommendation system for finding Fediverse servers?

Based on these findings, we suggest a need for better ways for newcomers to find servers and propose a viable way to create server and tag recommendations on Mastodon. This system could both help newcomers find servers that match their interests and help established accounts discover “neighborhoods” of related servers.

5.1 Constraints and Evaluation

The decentralized web presents unique challenges for recommendation systems. Centralized recommendation systems can collect data from all users and use this data to make recommendations. However, this is less desirable on the decentralized web, where data is spread across many servers and users may not want to share their data with a central authority. Instead, I propose a system where servers can report the top hashtags by the number of unique accounts on the server using them during the last three months. Such a system would be opt-in and require few additional server resources since tags already have their own database table. Because each server only reports aggregated counts of publicly posted hashtags, this also reduces the risk of privacy violations.

In the Mastodon context, the cold start problem has two possible facets: there is no information on new servers and there is also no information on new users. New servers are thus likely prone to falling for popularity bias: there is simply more data on larger servers. A common strategy to deal with new users is to ask for some initial preferences to create an initial workable user profile. In the case of this system, we ask the user to provide a set of tags which they are interested in. We then use these tags to find the top servers which match these tags.

I plan to evaluate the system in part using the accounts which moved between servers. Based on their posting history (e.g. hashtags), can the recommendations system predict where they will move to?

As my recommender system operates under the assumption that smaller, more topic-focused servers are better, it follows that a diverse set of niche results which only match a few tags are more helpful than a set of results which match a larger and more broad set of tags. The system therefore presents results sorted in a manner which encourages a higher diversity of results.

One current limitation of my system is that it does not account for the relationship between tags, e.g. “union” and “unions” are essentially the same tag and “furry” and “fursuit” are highly related tags which are in similar areas of embedded space. In future revisions, I hope to account for the relationship between similar tags and pull the top servers from clusters of highly related tags with the top priority going to clusters based on their number of selected tags. This system could be implemented efficiently in $O(nt)$ time given a minimum cluster size of t .

5.2 Recommendation System Design

We use Okapi BM25 to construct a term frequency-inverse document frequency (TF-IDF) model to associate the top tags with each server using counts of tag-account pairs from each server for the term frequency and the number of servers that use each tag for the inverse document frequency. We then L2 normalize the vectors for each tag and calculate the cosine similarity between the tag vectors for each server.

$$tf = \frac{f_{t,s} \cdot (k_1 + 1)}{f_{t,s} + k_1 \cdot (1 - b + b \cdot \frac{|s|}{avgstl})}$$

where $f_{t,s}$ is the number of accounts using the tag t on server d , k_1 and b are tuning parameters, and $avgstl$ is the average sum of account-tag pairs. For the inverse document frequency, we use the following formula:

$$idf = \log \frac{N - n + 0.5}{n + 0.5}$$

where N is the total number of servers and n is the number of servers where the tag appears as one of the top tags. We then apply L2 normalization:

Table 3. Top five servers most similar to hci.social

Server	Similarity
sigmoid.social	0.341
sciences.social	0.274
arvr.social	0.249
openedtech.social	0.233
akademienl.social	0.182

$$tf \cdot idf = \frac{tf \cdot idf}{\|tf \cdot idf\|_2}$$

We then used the normalized TF-IDF matrix to produce recommendations using SVD where the relationship between tags and servers can be presented as $A = U\Sigma V^T$. We then use the similarity matrix to find the top servers which match the user's selected tags. We can also suggest related tags to users based on the similarity between tags, $U\Sigma$.

5.3 Applications

5.3.1 Server Similarity Neighborhoods. Mastodon provides two feeds in addition to a user's home timeline populated by accounts they follow: a local timeline with all public posts from their local server and a federated timeline which includes all posts from users followed by other users on their server. We suggest a third kind of timeline, a *neighborhood timeline*, which filters the federated timeline by topic.

We calculate the pairwise similarity between two servers with TF-IDF vectors A and B using cosine similarity:

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

For an example of how this might work in practice, consider a use-case for someone who is a researcher in the field of human-computer interaction. They might be situated with an account on `hci.social`, but also interested in discovering posts from account on similar related servers. We can use the similarity matrix to find the top five servers most similar to `hci.social`, which is shown in Table 3.

5.3.2 Tag Similarity. We also calculate the similarity between tags using the same method. This can be used to suggest related tags to users based on their interests.

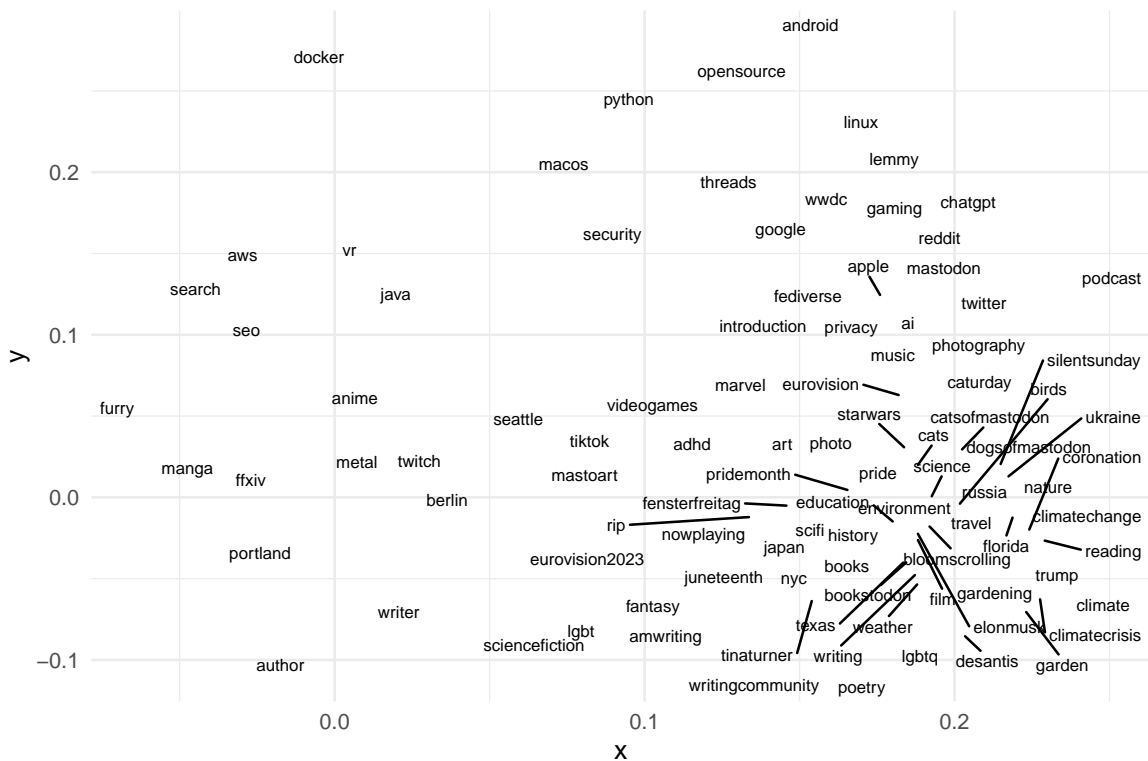


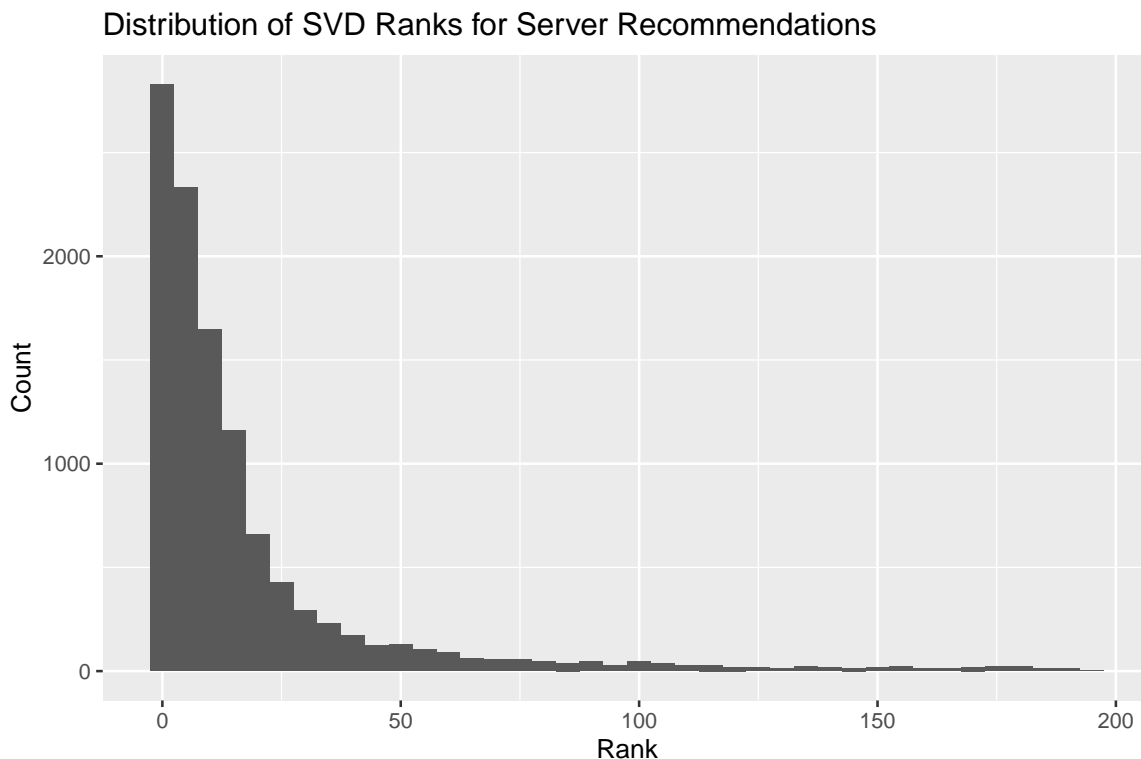
Fig. 3. 100 popular hashtags visualized in two dimensions using a principal component analysis (PCA) on the transformed singular value decomposition (SVD) matrix.

5.3.3 Server Discovery. Given a set of popular tags and a list of servers, we build a recommendation system where users select tags from a list of popular tags and receive server suggestions. The system first creates a subset of vectors based on the TF-IDF matrix which represents the top clusters of topics. After a user selects the top tags of interest to them, it suggests servers which match their preferences.

5.4 Evaluation

5.4.1 Server Recommendations for Users (Offline).

Time-based. For evaluation, we plan to use data from posts on accounts during a different time period from the one we used to train the recommender system. The goal of the system is to suggest the best servers for these accounts.



The SVD system predicts the server with a median rank of 8 and a mean rank of 18.

Movement-based. In parallel with the analysis of server survival, we also take an interest in users who moved servers since we can assume that these users found a server they liked better than their original server. We can use the recommender system to predict where these users will move to and use these predictions to evaluate the system.

5.4.2 Online Evaluation. I have also given some thought to online evaluation. Could we use an alternative version of the front-end to produce recommendations for interesting servers from existing accounts?

5.4.3 Robustness to Limited Data. A challenge for a federated recommendation system like we propose is that it needs buy in from a sufficient number of servers to provide value. There is also a tradeoff between the amount of tags to expose for each server and potential concerns about exposing too much data.

We simulated various scenarios that limit both servers that report data and the number of tags they report. We used rank biased overlap (RBO) to then compare the outputs from these simulations to the baseline with more complete information from all tags on all servers [33]. In particular, we gave a higher weight to suggestions with a higher rank, with weights decaying by a factor of $k^{0.80}$. Figure 4 shows how the average agreement with the baseline scales, which take the top 256 tags from each server.

6 DISCUSSION

This work provides a first step toward building a recommendation system for finding servers on the Fediverse based on empirical evidence of trace data from thousands of Fediverse newcomers. We find that servers matter and that users

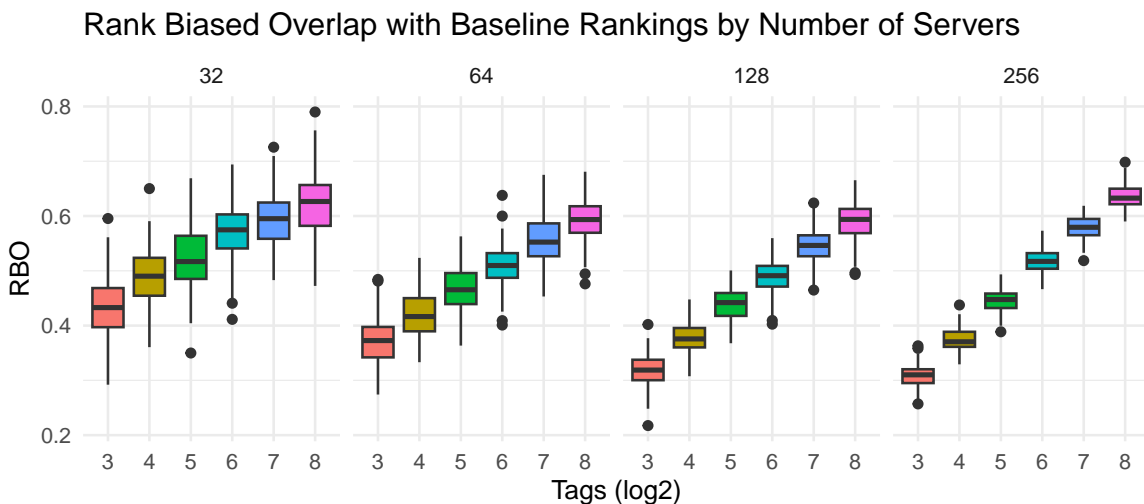


Fig. 4

tend to move from larger servers to smaller servers. Our recommender system considers constraints in a novel context where data is decentralized and privacy is a major concern. We propose a federated recommendation system which can be implemented with minimal resources and which can provide value to users by helping them find servers which match their interests.

The analysis can also be improved by additionally focusing on factors lead to accounts remaining active or dropping out, which a particular focus on the actual activity of accounts over time. For instance, do accounts that interact with other users more remain active longer? Are there particular markers of activity that are more predictive of account retention? Future work could use these to provide suggests for ways to helps newcomers during the onboarding process.

The observational nature of the data limit some of the causal claims we can make. It is unclear, for instance, if accounts on general servers are less likely to remain active because of the server itself or because of the type of users who join such servers. For example, it is conceivable that the kind of person who spends more time researching which server to join is more invested in their Mastodon experience than one who simply joins the first server they find.

6.1 Future Work

Future work is necessary to determine the how well the recommendation system is at helping users find servers that match their interests. This may involve user studies and interviews to determine how well the system works in practice.

While the work presented here is based on observed posts on the public timelines, simulations may be helpful in determining the robustness of the system to targeted attacks. Due to the decentralized nature of the system, it is feasible that a bad actor could set up zombie accounts on servers to manipulate the recommendation system. Simulations could help determine how well the system can resist such attacks and ways to mitigate this risk.

7 CONCLUSION

Based on analysis of trace data from millions of new Fediverse accounts, we find evidence that suggests that servers matter and that users tend to move from larger servers to smaller servers. We then propose a recommendation system that can help new Fediverse users find servers with a high probability of being a good match based on their interests. Based on simulations, we demonstrate that such a tool can be effectively deployed in a federated manner, even with limited data on each local server.

REFERENCES

- [1] Vannevar Bush. 1945. As We May Think. *The Atlantic* 176, 1 (July 1945), 101–108.
- [2] Dan Cosley, Dan Frankowski, Loren Terveen, and John Riedl. 2007. SuggestBot: Using Intelligent Task Routing to Help People Find Work in Wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI '07)*. ACM, New York, NY, USA, 32–41. <https://doi.org/10.1145/1216295.1216309>
- [3] Jesus Diaz. 2022. Using Mastodon Is Way Too Complicated to Ever Topple Twitter. <https://www.fastcompany.com/90808984/using-mastodon-is-way-too-complicated-to-ever-topple-twitter>.
- [4] Kevin Driscoll. 2023. Do We Misremember Eternal September?
- [5] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. 2011. Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human-Computer Interaction* 4, 2 (May 2011), 81–173. <https://doi.org/10.1561/1100000009>
- [6] Jiahui He, Haris Bin Zia, Ignacio Castro, Aravindh Raman, Nishanth Sastry, and Gareth Tyson. 2023. Flocking to Mastodon: Tracking the Great Twitter Migration. In *Proceedings of the 2023 ACM on Internet Measurement Conference (IMC '23)*. Association for Computing Machinery, New York, NY, USA, 111–123. <https://doi.org/10.1145/3618257.3624819>
- [7] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW '00)*. Association for Computing Machinery, New York, NY, USA, 241–250. <https://doi.org/10.1145/358916.358995>
- [8] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* 22, 1 (Jan. 2004), 5–53. <https://doi.org/10.1145/963770.963772>
- [9] Amanda Hoover. 2023. The Mastodon Bump Is Now a Slump. *Wired* (Feb. 2023).
- [10] Charles Kiene, Andrés Monroy-Hernández, and Benjamin Mako Hill. 2016. Surviving an “Eternal September”: How an Online Community Managed a Surge of Newcomers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1152–1156. <https://doi.org/10.1145/2858036.2858356>
- [11] Jaz-Michael King. 2024. Mastodon Near Me.
- [12] Yehuda Koren, Steffen Rendle, and Robert Bell. 2022. Advances in Collaborative Filtering. In *Recommender Systems Handbook* (third edition ed.), Francesco Ricci, Li'or Rokah, and Bracha Shapira (Eds.). Springer, New York, NY, 91–142.
- [13] Barbara Krasnoff. 2022. Mastodon 101: How to Follow (and Unfollow) Other Accounts. <https://www.theverge.com/23519279/mastodon-instance-follow-friend>.
- [14] Robert E. Kraut, Paul Resnick, and Sara Kiesler. 2011. *Building Successful Online Communities: Evidence-Based Social Design*. MIT Press, Cambridge, Mass.
- [15] Lucio La Cava, Luca Maria Aiello, and Andrea Tagarelli. 2023. Drivers of Social Influence in the Twitter Migration to Mastodon. *Scientific Reports* 13, 1 (Dec. 2023), 21626. <https://doi.org/10.1038/s41598-023-48200-7>
- [16] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing Cold-Start Problem in Recommendation Systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC '08)*. Association for Computing Machinery, New York, NY, USA, 208–211. <https://doi.org/10.1145/1352793.1352837>
- [17] Mastodon gGmbH. [n. d.]. Servers. <https://joinmastodon.org/servers>.
- [18] Matthew N. Nicholson, Brian C Keegan, and Casey Fiesler. 2023. Mastodon Rules: Characterizing Formal Rules on Popular Mastodon Instances. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 86–90. <https://doi.org/10.1145/3584931.3606970>
- [19] Arkadiusz Paterek. 2007. Improving Regularized Singular Value Decomposition for Collaborative Filtering. *Proceedings of KDD cup and workshop* (Aug. 2007).
- [20] Aravindh Raman, Sagar Joglekar, Emiliano De Cristofaro, Nishanth Sastry, and Gareth Tyson. 2019. Challenges in the Decentralised Web: The Mastodon Case. In *Proceedings of the Internet Measurement Conference (IMC '19)*. Association for Computing Machinery, New York, NY, USA, 217–229. <https://doi.org/10.1145/3355369.3355572>
- [21] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)*. ACM, New York, NY, USA, 175–186. <https://doi.org/10.1145/192844.192905>

- [22] Francesco Ricci, Li'or Rokah, and Bracha Shapira (Eds.). 2022. *Recommender Systems Handbook* (third edition ed.). Springer, New York, NY.
- [23] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389. <https://doi.org/10.1561/1500000019>
- [24] Eugen Rochko. 2023. Mastodon 4.2. <https://blog.joinmastodon.org/2023/09/mastodon-4.2/>.
- [25] Eugen Rochko. 2023. A New Onboarding Experience on Mastodon. <https://blog.joinmastodon.org/2023/05/a-new-onboarding-experience-on-mastodon/>.
- [26] Emma Roth. 2023. It's Getting Easier to Make an Account on Mastodon. <https://www.theverge.com/2023/5/1/23707019/mastodon-account-creation-twitter-alternative>.
- [27] Gerard Salton and Michael J. McGill. 1987. *Introduction to Modern Information Retrieval* (3. pr ed.). McGraw-Hill Book Comp, New York.
- [28] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. Association for Computing Machinery, New York, NY, USA, 285–295. <https://doi.org/10.1145/371920.372071>
- [29] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Springer, Berlin, Heidelberg, 291–324. https://doi.org/10.1007/978-3-540-72079-9_9
- [30] Amanda Silberling. 2023. A Beginner's Guide to Mastodon, the Open Source Twitter Alternative | TechCrunch. <https://techcrunch.com/2023/07/24/what-is-mastodon/>.
- [31] TheKinrar. 2017. Mastodon Instances. <https://instances.social/>.
- [32] Jan Trienes, Andrés Torres Cano, and Djoerd Hiemstra. 2018. Recommending Users: Whom to Follow on Federated Social Networks. <https://doi.org/10.48550/arXiv.1811.09292> arXiv:1811.09292 [cs]
- [33] William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Transactions on Information Systems* 28, 4 (Nov. 2010), 20:1–20:38. <https://doi.org/10.1145/1852102.1852106>
- [34] Eva Zangerle and Christine Bauer. 2022. Evaluating Recommender Systems: Survey and Framework. *Comput. Surveys* 55, 8 (Dec. 2022), 170:1–170:38. <https://doi.org/10.1145/3556536>
- [35] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 85–93. <https://doi.org/10.1145/3437963.3441820>
- [36] Diana Zulli, Miao Liu, and Robert Gehl. 2020. Rethinking the “Social” in “Social Media”: Insights into Topology, Abstraction, and Scale on the Mastodon Social Network. *New Media & Society* 22, 7 (July 2020), 1188–1205. <https://doi.org/10.1177/1461444820912533>

.1 Glossary

ActivityPub: A decentralized social networking protocol based on the ActivityStreams 2.0 data format.

Fediverse: A set of decentralized online social networks which interoperate using shared protocols like ActivityPub.

Mastodon: An open-source, decentralized social network and microblogging community.

Hashtag: A user-generated metadata tag that can be added to posts.

Federated timeline: A timeline which includes all posts from users followed by other users on their server.

Local timeline: A timeline with all public posts from the local server.

.2 More Evaluation

.2.1 *User Stories*. We also illustrate the potential value of such a system with three user stories:

User Story 1: Juan is a human-computer interaction researcher looking for a server to connect with colleagues and also share about his projects. He is interested in finding a server with a focus on research and technology. Juan inputs the tags “research”, “academia”, and “technology” into the system and receives a list of servers which match his interests: `synapse.cafe`, `sciences.social`, `mathstodon.xyz`, `mastodon.social`, `mastodon.education`.

User Story 2 (Arthur) just wants to connect with friends and family. For some reason, Arthur clicks every single major category and gets the suggestions: `mas.to`, `library.love`, `mastodon.world`, `mstdn.social`.

User Story 3 (Tracy) has run a niche fandom blog on Tumblr for the last eight years and is curious about migrating to the Fediverse. She inputs the tags “doctorwho”, “fanart”, and “fanfiction” and gets the suggestions: `blorbo.social`, `mastodon.nz`, `sakurajima.moe`, `toot.kif.rocks`, `mastodon.scot`.