Proceedings of the Eighteenth International AAAI Conference on Web and Social Media (ICWSM 2024)

Understanding and Improving Content Moderation in Web3 Platforms

Wenrui Zuo[1], **Raul J Mondragon**[1], **Aravindh Raman**[3], **Gareth Tyson**[1,2]

[1]Queen Mary University of London
[2]Hong Kong University of Science and Technology (GZ)
[3]Telefónica Research
w.zuo@qmul.ac.uk, r.j.mondragon@qmul.ac.uk, Aravindh.raman@telefonica.com, gtyson@ust.hk

Abstract

There have been numerous recent attempts to "decentralize" social media platforms, loosely referred to as "Web3". Such ideas, often underpinned by blockchain solutions, offer decentralized equivalents of well known services (*e.g.,* forums, social networks, video sharing sites, microblogs). One particularly challenging function to implement in such an architecture is *content moderation*, due to the lack of central control. Consequently, they often rely on user-controlled moderation, whereby each user must create their own personal block list to filter out content they do not wish to see. This paper presents the first study of user-controlled moderation on one exemplar Web3 social microblogging platform, `memo.cash`. Based on a dataset covering 391K posts, we study the factors that lead users to filter each other. We find that the most crucial factor is the level of activity on the platform, rather than the presence of things like hate speech. We also show that the followership network plays a pivotal role in determining user visibility on the platform, further influencing who gets filtered. This leads us to design tooling to automate this moderation process on a per user basis. We model this as a recommendation problem, and experiment with a number of state-of-the-art recommender engines. We show that our system can generate effective personalized filter lists for users.

1 Introduction

In traditional centralized online social networks, users must trust service providers. For example, users of Facebook must rely on them to enforce moderation policies, maintain reliable operations and keep their data safe. Unfortunately, there are few avenues to recourse when this trust is breached, as vendor lock-in makes it difficult for users to easily migrate to alternative providers. Indeed, there are numerous cases of large social network providers failing their users, *e.g.,* via leakage of users' personal data (Lam, Chen, and Chen 2008; Yang, Qu, and Cudré-Mauroux 2018).

Due to this, we have witnessed an increasing push towards "decentralization", often referred to as Web3. Such ideas, often underpinned by blockchain solutions, offer decentralized equivalents of well-known services (*e.g.,* forums, social networks, video sharing sites, microblogs) (Zuo et al. 2023a). Such platforms lack any single point of control and, instead, mediate access through a set of public protocols and smart contracts. One particularly challenging function to implement in such am architecture is *moderation*, due to the lack of central control. Consequently, they rely on user-controlled moderation, whereby individual users must create their own personal block lists to filter out content they do not wish to see. This, however, creates a significant overhead for users and makes it necessary for users to view at least one post before they can decide to block the target user. We argue that this new approach to moderation offers an interesting point on the design space, and highlights a challenge that any future decentralized web solutions must overcome.

With this in mind, we focus on one exemplar Web3 platform called `memo.cash`. This is a microblogging platform, constructed on top of the Bitcoin Cash (BCH) blockchain. On `memo.cash`, all user interactions are permanently recorded on a public ledger, removing any form of central control. This includes posts, follow actions, and post liking — all are recorded within transactions. The action that we are most interested in is `memo.cash`'s "mute" feature. Through muting, users (referred to as the mute issuer) can selectively filter other users (the muted user) from their timelines without unfollowing them, or deleting their posts. In this user-level moderation system, every user participates as a moderator, and muting serves as the sole moderation method to minimize or reduce the visibility of harmful content. As a result, moderators create a mute list, which consists of a compilation of muted users openly showcased on their profiles, ensuring complete moderation transparency. In the event of dissatisfaction with controversial mute actions, users can initiate appeals or review processes by sending direct messages to the user who create mute list, or by publicly posting their appeals. We focus on `memo.cash` because, to the best of our knowledge, it is the sole Web3 platform that implements user-to-user moderation. In contrast, other Web3 platforms like Steemit, Noise.cash, and Read.cash adopt a less decentralized approach, where community moderators and admins are involved in content moderation.

This empowerment grants individual users a heightened level of control over their online experiences, allowing them to determine the content they wish to engage with and fostering a significant degree of autonomy. We argue that this creates the perfect use case to study user-controlled content

moderation. Consequently, we gather a dataset covering 27K users, 391K posts, 1.5M social interactions (including 88.4K followings, 375K likes and 6.2K mutes) on memo.cash. The data covers a span of five years, from April 2018 (the platform's inception) to June 2023.

We begin by exploring features that correlate with the issuing of mutes (Section 3). To achieve this, we build a set of classifiers to distinguish between muted and non-muted users. We find that the most crucial factor is the platform action count, rather than the presence of things like hate speech. This is drawn from the observation that action count ranks first when modeling the classification between muted and non-muted users, while the count of hate words ranks considerably lower (26th) among all features. We also find that, in addition to an individual user's characteristics, their position within the followership network plays a pivotal role in determining their visibility on the platform, further influencing their muting behavior. Consequently, we group users into communities based on their followership network and measure the frequency of muting within their communities (Section 4). We show that a significant proportion of users tend to mute others within their own communities. This leads us to conjecture that we could design tooling to automate the muting process. Specifically, our goal is to generate personalized mute lists for users, alleviating the human workload (Section 5). We model this as a recommendation problem, and experiment with a number of state-of-the-art recommender engines. We show that our system can generate effective personalized mute lists for users. Instead of relying solely on muting unwanted users after being annoyed, the recommendation system suggests possibly muting users based on the mute behavior of the users themselves and their followers. Our work has important implications for not only Web3 platforms, but also other social networks that rely on user-driven moderation.

Our study makes the following key contributions:

- We investigate the most important features that result in a user being muted on memo.cash. We build a set of models to distinguish between muted and non-muted users. We find that the presence of hateful text does not strongly drive muting behavior. In fact, it is ranked 26th in terms of feature importance. Instead, the number of social actions performed by users is the most significant factor. This could be attributed to the fact that active users are exposed to more users, increasing the likelihood of being muted by some individuals.
- We investigate the impact of the followership network structure on mute behavior. We find that muting users in other communities on the followership network is rarely seen, as users tend to mute others who belong to the same community. Upon dividing users into 11 main communities based on their followership network, we find that 60.9% of mutes occur within the same community. The reason may be that users are more likely to encounter and mute other users from within their own community, making it less common for mutes to extend across different communities.
- We propose a decentralized and personalized mute recommendation system for individual users to create top-n mute lists. We employ unique labels (1, 2, 3, 4) to quantify users' willingness to mute. Our dataset comprises 6,242 mute events (labels). We then evaluate various recommendation techniques, including Random, Most Popular, SVD, UserKNN, LightFM, and CNN models, based on these 6,242 labels. LightFM stands out, achieving strong performance with a high F1 score (0.76) and low MAE (1.62) on average in the top-10 mute list scenario.

2 Dataset Overview

2.1 memo.cash and Dataset

memo.cash: primer. memo.cash operates as a microblogging service running on the BitcoinCash (BCH) blockchain, where all data is stored on-chain. In terms of user-facing functionality, memo.cash is similar to other microblogs such as Twitter. The platform's creator describes it as "both a protocol and a front-end application". A user's interactions with other users are recorded as transactions on the BCH blockchain and stored in the OP_RETURN field (Zuo et al. 2023b). This means that all user's data, including posts, following, and likes, is permanently stored without the possibility of censorship. Users can access the data directly from the blockchain store, with a web-based frontend for convenient presentation.

One notable feature of memo.cash is its user-controlled moderation capability known as *muting*, which is similar to the blocking on Twitter. When a user mutes another user, the muted user's posts becomes hidden to the issuer of the mute. Importantly, the act of muting is recorded on the blockchain via the OP_RETURN field, ensuring transparency and immutability.

Dataset collection. We obtain the dataset from the memo.cash web frontend using a web crawler. This provides a convenient way to access the underlying blockchain-stored data. The dataset covers the time period from April 2018, when the platform was created, to June 2023. It consists of information from approximately 27,630 users and includes a total of 1.4M social actions recorded on the platform. These social actions encompass 391K posts, 375K likes, and 6,240 mutes. A summary of the dataset is in Table 1.

Our initial step involves extracting user identifiers from the webpage. Subsequently, we collect all user-related data, including profile information, follower lists, mute lists, and associated metadata. Following the user data extraction, we proceed to gather all publicly available posts listed on each user's profile. This encompasses the post text itself, as well as the number (lists) of "likes" received from other users for each individual post.

2.2 Motivation

We observe 6,239 mutes in total, with 417 users initiating mutes and 2,215 users receiving mutes. We observe that, among individuals who have been muted, 91.3% of them have also received at least one like for their posts. In fact, muted users receive an average of 384.7 likes (median 85),

Users	Content
user id*	user-post content
user activities*	likes*
following users*	
followers*	
mute receivers*	
mute issuers*	

* includes creation/action time.

Table 1: Dataset description

demonstrating how challenging it is to attain total consensus on moderation. Even among those users who have been muted by more than 5 people, 35.4% also receive likes on their posts. Hence, we argue that achieving a consensus on who should be muted is challenging.

3 Exploring Reasons for Muting

To explore which features are most determinate in triggering an account to be muted, we next train a number of classifiers to distinguish between muted and non-muted accounts. We then explore their feature importance.

3.1 Modelling Users Features

Our dataset contains 27,630 total users, with 2,215 users receiving at least one mute (label=1) and 25,415 non-muted users (label=0). We then build models to predict whether a given user would be muted or not. We extract two feature sets, covering both user-based and content-based features, shown in Table 2. In total, we extract 406 features, which consist of 9 user-based features and 397 content-based features taken from the post's text. Among the content-based features, there are 13 text-based features and 384 BERT embedding dimensions.

The user-based features include the days since user creation, the number of followers, etc. We also consider text content. A higher toxicity scores indicate that text has a greater toxicity. A higher sentiment score suggests the text is more positive. To understand how different emotions are expressed by various sets of users, we analyze the sentiment of text using VADER (Valence Aware Dictionary for Sentiment Reasoning) (Hutto and Gilbert 2014) to find the sentiment expressed by the users. Typical thresholds for classifying sentences as either positive, neutral, or negative are determined by the polarity (compound) score. The compound score is computed by totalling the valence scores of each word in the lexicon, where valence denotes the sentiment intensity score and is rated on a scale from '–4' (extremely negative) to '4' (extremely positive), with an allowance for '0' (neutral). These valence scores tend to be normalized to be between -1 (most extreme negative) and +1 (most extreme positive) We also count the number of hate words using the HateBase API. This captures specific properties related to post semantic information and toxicity.

We also use BERT embeddings to identify the semantics information that a user publishes. We use a pre-trained Bert model (Mozafari, Farahbakhsh, and Crespi 2020) to embed

User-Based	Features Description
creation_days	Number of days since one user created
profile_creation	If one user has set profile
action	Number of action of one user
followers	Number of followers of one user
following	Number of following of one user
topic_count	Number of following topics of one user
Issuing mutes	Number of mutes issued of one user
balance	Total bitcoin balance in one user's wallet
transaction_count	Number of transaction for one user
Content-Based	
post_count	Number of posts published by one user
comment_count	Total number of comments for one user's post
comment_avg	Average comments count per post for one user
like_count	Total number of likes for one user's post
like_avg	Average number of likes per post for one user
star_count	Total number of stars for one user's post
star_avg	Average number of stars per post
rewards_count	Total post rewards for one user
rewards_avg	Average rewards per post for one user
sentiment_score	Average sentiment score per post for one user
toxicity_score	Average toxic score per post for one user
profanity_score	Average profanity score per post for one user
threat_score	Average threat score per post for one user
sexually_score	Average sexually explicit score for one user
hateword_count	Total number of hate words for one user's post
hateword_avg	Average hate words count per post for one user
Bert_1	The 1st dimension (Bert)vector for user's post
Bert_2	The 2nd dimension (Bert)vector for user's post
...	
Bert_387	The 387th dimension vector for user's post

Table 2: Modelling features list

the text into 384 dimensions of vectors for each post. Once the individual post's vector is obtained, the mean value of all posts' vectors from one `memo.cash` account is used as a feature. If a user does not post, all 384 dimensions will be set to 0. Ultimately, one user has 384-dimension vectors to represent published posts (called a user-content vector), and we define each dimension of the user-content vector as a Bert-involved feature: Bert_1 for the first dimension of the user-content vector, Bert_2 for the second dimension of the user-content vector and so on.

3.2 Model Configuration

We next use the above features to classify users who receive one or more mutes. Thus, the target variable is a user being muted or not.

Model selection. We experiment with several classifiers to predict a given user's likelihood of being muted or not on `memo.cash`: Random Forest (RF) (Breiman 2001), Logistics Regression (LR) (Wright 1995), Gradient Boosted Decision Trees (GBDT) (Neelakandan and Paulraj 2020) and Support Vector Machines (SVM) (Hearst et al. 1998). This enables us to identify the most appropriate model for prediction. Apart from traditional machine-learning algorithms, we also use Graph Convolutional Networks (GCNs) (Zhang et al. 2019). As an input, we use the followership graph, and each node has 406 features as listed in Table 2.

Models	Hyperparameters	Value
LR	”penalty”	[’l1’,’l2’]
	”C”	[0.001,0.01,0.1,1,10,100]
SVM	”gamma” (’kernel’ = rbf)	[1e-3, 1e-4]
	”C” (’kernel’ = rbf)	[1, 10, 100, 200]
	”C” (’kernel’ =linear)	[1, 10, 20,30, 40, 100]
RF	”n_estimators”	[50, 200, 300,700]
	”max_features”	[’auto’, ’sqrt’, ’log2’]
GBDT	”learning_rate”	[0.5, 0.075, 0.1, 0.15, 0.2]
	”min_samples_split”	np.linspace(0.1, 0.5, 4)
	”min_samples_leaf”	np.linspace(0.1, 0.5, 4)
	”max_depth”	[3,5,8]
	”max_features”	[”log2”,”sqrt”]
	”criterion”	[”friedman_mse”, ”mae”]
	”subsample”	[0.5, 0.6, 0.8, 0.9, 1.0]
GCNs	”epochs ”	[100, 200, 500]
	”learning rate”	[0.01, 0.05, 0.1]
	”node embedding”	[100, 200, 300]

Table 3: Models hyperparameters

Models	Accuracy	Precision	Recall	F1 Score
LR	0.98	0.67	0.45	0.53
SVM	0.98	0.80	0.28	0.42
RE	0.98	**0.85**	**0.55**	**0.62**
GBDT	0.98	0.64	0.39	0.49
GCNs	0.93	0.60	0.38	0.46

Table 4: Model results with corresponding scores in accuracy, precision, recall and f1 score.

Parameter configuration & Training. We use grid search to select the optimal hyperparameters for each model. The hyperparameters for the models is listed in Table 3. We train all models using an 80:20 train:test split.

3.3 Models Performance

Model performance. Table 4 presents the performance of all models. The best f1 score (0.62) is achieved with the Random Forest algorithm. The traditional models outperform GCN (f1 score of 0.46), and the main cause of this may lie in over-smoothing (Dai, Guo, and Feng 2020). This refers to a situation where, during the training process, the node representations become too similar or indistinguishable across different nodes in the graph. This makes it hard to distinguish between muted and non-muted users, especially for the imbalanced and small-sized `memo.cash` dataset.

Feature importance. We next assess the factors driving users’ engagement with the user-controlled moderation system on `memo.cash`. To visually represent the importance of individual features, Figure 1 presents a histogram depicting the feature importance. We include the top 30 crucial features utilized in the Random Forest modeling. Action count is the most essential feature in classifying muted and non-muted accounts. Unsurprisingly, this indicates that users active on the platform tend to be muted by others. By comparison, the Perspective API scores are not as important.

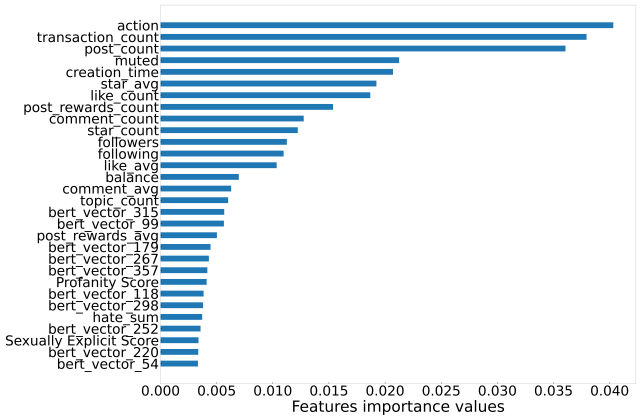

Figure 1: Top 30 Features importance histogram using Random Forest.

3.4 Users Posting

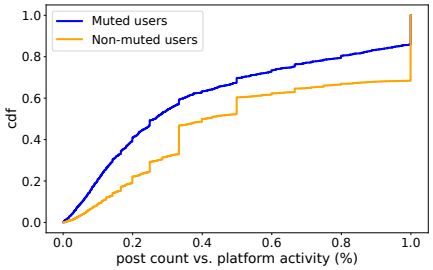

Figure 2: CDF of the post count of individual users as a ratio of the total platform activity count.

The previous section has shown that a user’s posting frequency is a strong indicator of their likelihood of being muted. Thus, we next take a closer look at the posting activity. Figure 2 plots the proportion of users’ post counts relative to their total platform activity (which covers all social actions performed by the user on the platform, including posting, following, and muting others). Our findings reveal that 19.6% of muted users and 33.1% of non-muted users exhibit a posting action vs. platform activity ratio greater than 0.8. This implies that there are only a few users who prioritize posting content over engaging in social interactions with others. This phenomenon can perhaps be attributed to the greater average platform engagement of muted users.

In addition to analyzing post frequency, we look at the posting time interval. This provides valuable insights into user posting engagement. Figure 3 plots each user’s post count vs. their average post time intervals (in seconds). On average, muted users have a post time interval of 215.3 seconds (median 34.0 seconds). In contrast, users who have not been muted typically display a longer average time between each post, at 437.7 seconds, and a median interval of 63.2 seconds. This discrepancy suggests that users who consistently post at shorter intervals are more actively engaged on the platform. Their frequent exposure may increase their visibility and likelihood of being moderated by other users.

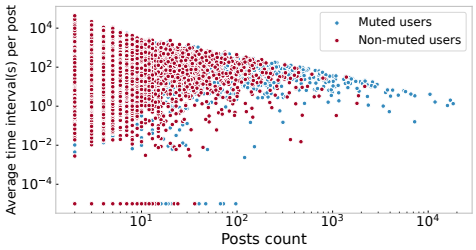

Figure 3: The number of posts and the mean time intervals (in seconds) between posts for both muted and non-muted individual users.

Furthermore, muted users' post content experience a lack of engagement, such as likes, tips, or replies, on 60.5% of their posts (as opposed to 47.0% for non-muted users). We are unclear about the specific reasons motivating the posting behavior of these muted users. However, the combination of frequent, short-interval posting and a significant proportion of posts receiving no feedback leads us to suspect the presence of low-quality or irrelevant content. This could be a contributing factor prompting users to resort to muting actions.

4 Exploring the Followership Network

The previous section revealed that the post visibility and follower count has an impact on the liklihood of being muted. As such, we next examine how the followership network structure influences user muting. We first segment users into communities based on the followership graph, before exploring if users mute within or outside of their community.

4.1 Followership Community Detection

First, a directed unweighted network is constructed using the followership data. In this network, each node represents a user, and the directed link refers to a follower relationship. The followership network comprises of 12,676 nodes and 60,809 links on `memo.cash`. We then use the Louvain Method (De Meo et al. 2011; Blondel et al. 2008) to perform community detection on the network. For context, Figure 4 plots a network vizualisation of the communities.

It is worth noting that the resolution value used in the Louvain method can influence the number of communities identified (Gao et al. 2018; Adam, Delvenne, and Thomas 2018). As the resolution value rises, the number of communities tends to increase gradually. Notably, when the resolution value reaches 0, all nodes are assigned to a single community. In past research (Adam, Delvenne, and Thomas 2018), a resolution value of 1.0 has frequently yielded a satisfactory quantity of communities. We therefore adopt this value. Despite adjusting the resolution value, certain users are not assigned to any of the identified communities, resulting in scattered communities consisting of individual users. These unaffiliated nodes are excluded, and each user is assigned a label corresponding to their respective community, and we have identified a total of 6 such communities. Out of the 72 communities detected, 61 of them consist of fewer than 10 users. Figure 5 illustrates the number of users belonging to the main communities. Out of the 11 main communities that are detected, Community A boasts the highest user count at 2,950, followed by Community B and Community C as the second and third largest, with 2,490 and 1,891 users, respectively.

Figure 4: Visualization of the mute graph. Each node is assigned a color based on its corresponding community within the followership network, while the size of each node is determined by the number of issued mutes.

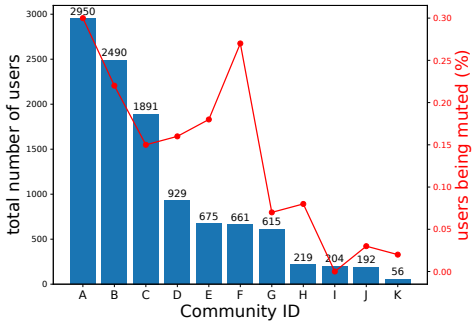

Figure 5: The distribution of users per community.

4.2 Measuring Localization of Mutes

Isolation index. We next examine whether the muting is localized within communities. We adopt the approach proposed by Massey Denton (Massey and Denton 1988). They defined the isolation of a specific community (e.g., community C) as follows:

$$P_{\partial} = \sum_{i=1}^{n} \frac{c_i}{C} \frac{c_i}{T_i} \quad (1)$$

In our case, c_i represents the number of users in community C who mute users in community i, C denotes the total number of users belonging to community C, and T_i represents the total count of users muting to community i. Each user is assigned to the community detected based on their followership network. Consequently, $\frac{c_i}{C}$ signifies the probability that a randomly selected user from community C will

mute users in community i, while $\frac{c_i}{T_i}$ represents the fraction of users muting to community i that belongs to community C. This quantifies the contribution of a specific community to the overall number of muted users in community i. Thus, P_∂ represents the probability that a user from community C will randomly issue a mute to someone within the same community in community i. Equation (1) provides a direct measure of the level of isolation among users in different communities (C) when it comes to mute behavior. A large value P_∂ (close to 1) indicates highly segregated communities, while smaller values suggest more integrated communities.

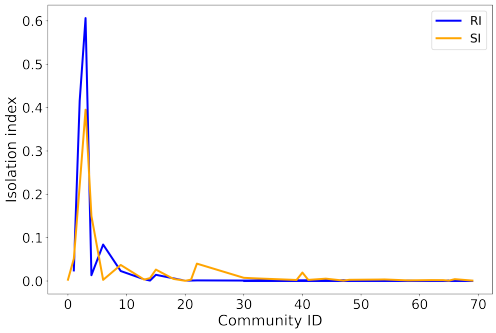

Figure 6: Real isolation index (RI) value and random simulation index (SI) for muting behavior of detected communities.

Results. To assess whether mute behavior is localized within communities, we calculate the real emperical isolation index (RI) for each community in our dataset. These values are then compared to the simulated isolation index (SI) derived from random simulations. As a baseline, we use the Erdős-Rényi model to simulate the scenario where mute behavior between any two communities is random. To evaluate the isolation of mute behavior, we depict the comparison between RI and SI in Figure 6. This demonstrates that RI distribution has a more skewed distribution: 9 out of 12 communities have RI values larger than expected based on our random simulation framework. This suggests that users are more inclined to mute users within their own community. Indeed, 60.9%, of the mutes originate from users belonging to that same community. This finding can be attributed to users' tendency to engage with content posted by users in the same community. This suggests that users in the same community might be well placed to assist with recommending mutes to each other. Therefore, in the subsequent section, we explore ways to automate the recommendation of mutes, exploiting community features.

5 Mute Recommendations

As discussed earlier, the users' platform activity, particularly their posting behavior, plays a pivotal role in muting (Section 3.3 and 3.4). Further, the followership graph is important as it impacts post visibility on the platform. Hence, in we next leverage both individual users' mute lists and the mute lists of their followers to propose a tool for generating personalized mute lists.

5.1 Dataset

Data summary. Within `memo.cash`, there are 6,242 mutes, with 417 users initiating mutes and 2,215 users receiving them. Notably, 1,421 mutes are initiated by users who are also muted by their followers, indicating strong negative feedback. Additionally, 8.4% of users have been both muted and liked by the same user. This underscores the challenge of achieving consensus regarding moderation.

Labels and features. In light of this situation, we generate labels to quantify interactions from one user to another based on muting, following, and liking. These labels quantify users' willingness to mute, thereby basing to design of the mute recommendations tool. Table 5 presents the meaning of each label, taking into account three key behaviors: a user's self-initiated mute actions, the muting actions taken by the user's followers, and the user's engagement in liking posts.

To perform recommendation modeling, we feed each pair of users (the one issuing the mute and the one receiving it). This involves creating modeling features that combine the user attributes of both the mute issuer (User A) and the mute receiver (User B) in one mute pair (User A to User B), along with the label associated with this pair of mutes. We analyze six user attributes: the total number of followers, the total number of likes on their posts, and the total number of mutes received for the mute issuer and mute receiver. Subsequently, we utilize these six user features for each pair of users to predict the likelihood of muting from one user to another, and we generate the top 5, 10, and 20 mute recommendations for individual users.

5.2 Experimental Setup

We frame the task as a recommendation problem, in order to generate personalized top 5, 10, and 20 mute lists for individual users, based on users' muting, following, and liking experience.

We compare several state-of-art recommendation techniques to determine the best-fit mute recommendation model, including Random Model, Most Popularity Model, Singular Value Decomposition(SVD), User-Based k-Nearest Neighbor(UserKNN), Light Factorization Machine(LightFM) as well as Convolutional Neural Network(CNN). Data preprocessing and convolutional layer normalization are common techniques in deep learning that improve model performance (Jung et al. 2019; Zhang and Izquierdo 2023). We thus employ four different CNN model variations: CNN-baseline, CNN-preprocess, CNN-normalised, and CNN-preprocess+normalised, which involve distinct combinations of data preprocessing and convolutional layer normalization. In the following, we summarize the above algorithms and explain why we consider them for our investigation, as well as detailing the parameters used in Table 6.

Label	Implication	Event Count
4	Very negative feedback: User A has muted user B, and user A's follower has done the same. Additionally, user A has not liked any user B posts.	525
3	Moderately negative feedback: While user A has muted user B, their followers have not taken such action. Nevertheless, user A has never liked user B's posts.	1,825
2	Somewhat negative feedback: User A has muted user B, and their follower has also done the same. However, user A has liked at least one of user B's posts.	896
1	Slightly negative feedback: Despite user A muting user B, user A has also liked a post by user B.	2,996

Table 5: Label definitions for modeling mute recommendation.

Evaluation procedure. Our experiments follow a 80-20 hold-out split (Rendle et al. 2020). We perform five-fold cross validation and present the mean values.

Evaluation metrics. We employ standard accuracy and error metrics frequently used in the recommendation literature. Concerning accuracy metrics, we compute the Normalized Discounted Cumulative Gain (nDCG), Mean Reciprocal Rank (MRR) at mute list lengths of 5, 10 and 20. It is important to note that previous research (Rajarajeswari et al. 2019; Anelli et al. 2022) has largely favored error metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for evaluation purposes. Both nDCG and MRR have a scale of 0 to 1, with 1 denoting optimal performance and 0 indicating bad performance. In contrast, RMSE and MAE typically assume positive real values, with a lower value signifying superior model performance. Additionally, we assess the performance of the models using standard information retrieval metrics like accuracy, recall, precision, and F1. Consequently, we adopt the aforementioned ranking metrics to appraise the efficacy of top-n (n=5,10,20) mutes recommendation tasks.

5.3 Model Performance

Overall performance. Figure 7 presents the average evaluation metrics for the different models when suggesting the top 5,10,20 mute lists per user. We find that LightFM consistently exhibits strong performance across all metrics and for all top-n recommendations, particularly in the top-10 mute list scenario. Notably, when examining RMSE and MAE metrics, we observe that the values do not significantly fluctuate with changes in n.

This is evident from the high F1 score (0.76) and low MAE (1.62) achieved for the top-10 recommendations. In contrast, the SVD model exhibits bad prediction performance with an F1 of just 0.28. Moving on to the nDCG and MRR metrics, LightFM demonstrates good performance for the top-10 mute list, with nDCG at 0.91 and MRR at 0.56. Our `memo.cash` mute-list recommendation system is susceptible to the cold start issue, since our models are constructed using data from only 50.7% of `memo.cash` users. This explains why LightFM outperforms other models, as it is adept in handling the cold start problem by enabling recommendations even with limited user data.

All CNN models exhibit subpar results. Specifically, the F1 score for the CNN-baseline is 0.16, while CNN-preprocess and CNN-normalized achieve only 0.23 and 0.25, respectively. The best-performing variant, CNN-normalized+preprocess, still falls short with an F1 score of just 0.28. These scores are significantly lower than those attained by the other models. We attribute this to the small training set size. Unlike traditional models, CNNs tend to require larger dataset sizes, making them unsuitable for our context.

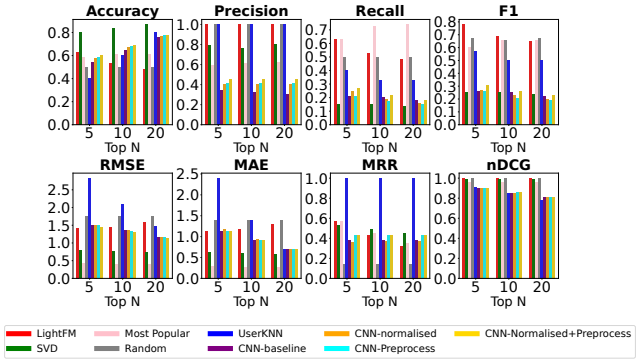

Figure 7: Mean values of evaluation metrics for all users for top-n (5,10,20) recommendation lists generated by various models.

Per-user performance. We also wish to explore how these performance metrics vary across individual users' recommendations. For example, some users may be receiving subpar recommendations, and these issues may not be evident when looking solely at average metrics.

To illustrate this, Figure 8 plots the cumulative distribution functions (CDFs) of accuracy, F1, RMSE, and MAE values for top-10 mute recommendations using various models. Our analysis indicates that for the majority of users, LightFM is the most suitable recommendation method, as only 38.8% of users exhibit F1 values <0.5, and merely 41.4% of users display accuracy values <0.5. This demonstrates a high level of accuracy in aligning the recommended mute list with the user mute actual preference. However, in terms of RMSE, approximately 20% of users exhibit values exceeding 1.7, indicating that LightFM may not be the best choice for a small portion of users.

Additionally, SVD performs well, with over 65% of users showing very low RMSE and MRR (<0.5). However, in comparison to LightFM, around 43.7% of users have F1 values lower than 0.2. This means that, despite accurately including the likely muted user for the top 10 mute list, the recommendations from SVD may not always place the most

likely muted user at the top of the mute list, potentially requiring users to scroll through the recommendations list.

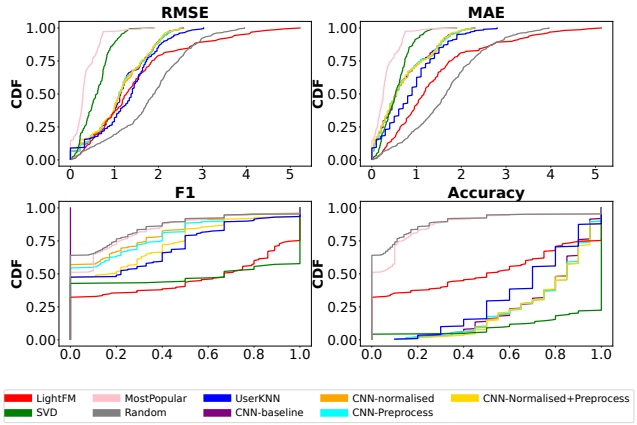

Figure 8: CDFs of evaluation metrics for individual users generated by various models.

The four CNN variant models again exhibit suboptimal performance, with 80% of users experiencing low F1 scores (<0.4), and 23% of users having RMSE values lower than 0.5. This implies that, on average, users receive mute lists that significantly diverge from their actual preferences, potentially leading to a less satisfying user experience. The Random model also exhibits poor performance, with more than 80% of users displaying low accuracy and F1 values (<0.2). Only just 9.8% of users exhibit an RMSE value <0.5, and 11.3% of users have an RMSE value <0.5. These results indicate that the recommendations are not effectively ranking muted users at the top 10 mute lists.

6 Related Work

6.1 Decentralized Social Media

Previous research has looked at decentralized social networks, encompassing network structure (Cheng, Dale, and Liu 2008), content analysis (Bin Zia et al. 2022; Cerisara et al. 2018; Lata and Gupta 2020), as well as privacy preservation (Cutillo, Molva, and Strufe 2009; Guidi, Michienzi, and Ricci 2021) and resource management (Cao 2022; Nezami et al. 2021). Decentralized social networks include federated (Anaobi et al. 2023; Trienes, Cano, and Hiemstra 2018; Raman et al. 2019), relay (Wei and Tyson 2024) and peer-to-peer (Tang et al. 2019; Dong et al. 2022) approaches. We contribute to this broad area by measuring content moderation in one such platform, `memo.cash`. More recently, there has been a surge in proposals to interlink blockchain technology with social media applications (Zuo et al. 2023b; Prodan et al. 2019; Li and Palanisamy 2019; Li et al. 2021). These often employ cryptocurrencies to incentivize users. Our work exists at the crossroads of these advancements, aiming to develop content moderation technologies for such platforms. We note our models would be equally applicable to non-decentralized social media platforms.

6.2 Web3 Moderation Recommendations

The objective of content moderation is to establish a healthy communication environment by mitigating aggression and anti-social speech (Langvardt 2017; De Gregorio 2020; Jhaver et al. 2018; Bo et al. 2020), while also ensuring compatibility between users and public values (Gorwa, Binns, and Katzenbach 2020; Liu and Bakici 2019). In recent times, the drive towards decentralization has been partially prompted by the content moderation policies of large social media platforms. However, implementing decentralized content moderation introduces new challenges (Hassan et al. 2021; Gillett and Suzor 2022). Enhanced efficiency in moderation is achieved through platforms' recommendation systems that redistribute moderated content (Himeur et al. 2022; Trienes, Cano, and Hiemstra 2018). Notably, Web3 social platforms like `memo.cash`, Steemit, and SocialX (Li and Palanisamy 2019; Walia and Raghwa 2022) eliminate the need (or capacity) for centralized management. Our study pioneers an exploration of `memo.cash`, shedding light on the recommendation system's role in Web3 moderation.

7 Conclusion

This study has focused on moderation in Web3. As an exemplar, we focused on a major decentralized microblogging platform, `memo.cash`.

We started by inspecting the important features that correlate with moderation mutes being employed (Section 3). We found that 91.3% of muted users on `memo.cash` have received at least one like for their post, highlighting the difficult of reaching distributed consensus on what content should be muted. Hateful speech emerged as a less significant factor, with users' action count ranking as the most important feature. Beyond individual user traits, we also looked at the impact of the follower network structure (Section 4). We found that 60.9% of mutes originate from users within the same follower community as those being muted. We then proposed ways to generate personalized mute lists for individual users in Section 5. We employed a series of recommendation models to identify the most appropriate tool to recommend the top 10 mute list for each user. The LightFM model exhibited good performance. Our findings contribute fresh insights for researchers engaged in Web3 moderation systems. We shed light on the key factors that impact user-controlled moderation and present a novel approach for empowering users through personalized mute lists.

Our investigation into mute localization and mute recommendation is based on numerical data related to user interactions, including following, liking, and muting counts, without explicit consideration of post topics or content. It is independent of the nature of published content. As a result, this study can be feasible and scalable for other user-controlled moderation systems. However, our study has limitations, primarily stemming from the use of a limited `memo.cash` dataset for mute recommendations. To address this, future research could expand upon our findings by conducting comparisons on other Web3 platforms.

References

Adam, A.; Delvenne, J.-C.; and Thomas, I. 2018. Detecting communities with the multi-scale Louvain method: robustness test on the metropolitan area of Brussels. *Journal of Geographical systems*, 20(4): 363–386.

Anaobi, I. H.; Raman, A.; Castro, I.; Zia, H. B.; Ibosiola, D.; and Tyson, G. 2023. Will Admins Cope? Decentralized Moderation in the Fediverse. In *Proceedings of the ACM Web Conference 2023*, 3109–3120.

Anelli, V. W.; Bellogín, A.; Di Noia, T.; Jannach, D.; and Pomo, C. 2022. Top-n recommendation algorithms: A quest for the state-of-the-art. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, 121–131.

Anwar, T.; and Uma, V. 2021. Comparative study of recommender system approaches and movie recommendation using collaborative filtering. *International Journal of System Assurance Engineering and Management*, 12: 426–436.

Bin Zia, H.; Raman, A.; Castro, I.; Hassan Anaobi, I.; De Cristofaro, E.; Sastry, N.; and Tyson, G. 2022. Toxicity in the decentralized web and the potential for model sharing. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(2): 1–25.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10): P10008.

Bo, H.; McConville, R.; Hong, J.; and Liu, W. 2020. Social network influence ranking via embedding network interactions for user recommendation. In *Companion Proceedings of the Web Conference 2020*, 379–384.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Cao, L. 2022. Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci. *IEEE Intelligent Systems*, 37(3): 6–19.

Cerisara, C.; Jafaritazehjani, S.; Oluokun, A.; and Le, H. 2018. Multi-task dialog act and sentiment recognition on mastodon. *arXiv preprint arXiv:1807.05013*.

Cheng, X.; Dale, C.; and Liu, J. 2008. Statistics and social network of youtube videos. In *2008 16th Interntional Workshop on Quality of Service*, 229–238. IEEE.

Cutillo, L. A.; Molva, R.; and Strufe, T. 2009. Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network. In *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops*, 1–6. IEEE.

Dai, M.; Guo, W.; and Feng, X. 2020. Over-smoothing algorithm and its application to GCN semi-supervised classification. In *Data Science: 6th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2020, Taiyuan, China, September 18-21, 2020, Proceedings, Part II 6*, 197–215. Springer.

Daneshvar, H.; and Ravanmehr, R. 2022. A social hybrid recommendation system using LSTM and CNN. *Concurrency and Computation: Practice and Experience*, 34(18): e7015.

De Gregorio, G. 2020. Democratising online content moderation: A constitutional framework. *Computer Law & Security Review*, 36: 105374.

De Meo, P.; Ferrara, E.; Fiumara, G.; and Provetti, A. 2011. Generalized louvain method for community detection in large networks. In *2011 11th international conference on intelligent systems design and applications*, 88–93. IEEE.

Dong, J.; Song, C.; Liu, S.; Yin, H.; Zheng, H.; and Li, Y. 2022. Decentralized peer-to-peer energy trading strategy in energy blockchain environment: A game-theoretic approach. *Applied Energy*, 325: 119852.

Gao, Y.; Zhu, Z.; Kali, R.; and Riccaboni, M. 2018. Community evolution in patent networks: technological change and network dynamics. *Applied network science*, 3(1): 1–23.

Gillett, R.; and Suzor, N. 2022. Incels on Reddit: A study in social norms and decentralised moderation. *First Monday*, 27(6): Article–number.

Gorwa, R.; Binns, R.; and Katzenbach, C. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1): 2053951719897945.

Guidi, B.; Michienzi, A.; and Ricci, L. 2021. Data persistence in decentralized social applications: The ipfs approach. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 1–4. IEEE.

Hassan, A. I.; Raman, A.; Castro, I.; Zia, H. B.; De Cristofaro, E.; Sastry, N.; and Tyson, G. 2021. Exploring content moderation in the decentralised web: The pleroma case. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 328–335.

Hearst, M. A.; Dumais, S. T.; Osuna, E.; Platt, J.; and Scholkopf, B. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4): 18–28.

Himeur, Y.; Sayed, A.; Alsalemi, A.; Bensaali, F.; Amira, A.; Varlamis, I.; Eirinaki, M.; Sardianos, C.; and Dimitrakopoulos, G. 2022. Blockchain-based recommender systems: Applications, challenges and future opportunities. *Computer Science Review*, 43: 100439.

Hutto, C.; and Gilbert, E. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, 216–225.

Jhaver, S.; Ghoshal, S.; Bruckman, A.; and Gilbert, E. 2018. Online harassment and content moderation: The case of blocklists. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(2): 1–33.

Jung, W.; Jung, D.; Kim, B.; Lee, S.; Rhee, W.; and Ahn, J. H. 2019. Restructuring batch normalization to accelerate CNN training. *Proceedings of Machine Learning and Systems*, 1: 14–26.

Lam, I.-F.; Chen, K.-T.; and Chen, L.-J. 2008. Involuntary information leakage in social network services. In *International Workshop on Security*, 167–183. Springer.

Langvardt, K. 2017. Regulating online content moderation. *Geo. LJ*, 106: 1353.

Lata, M.; and Gupta, A. 2020. Role of social media in Environmental Democracy. In *Examining the Roles of IT and Social Media in Democratic Development and Social Change*, 275–293. IGI Global.

Li, C.; and Palanisamy, B. 2019. Incentivized blockchain-based social media platforms: A case study of steemit. In *Proceedings of the 10th ACM conference on web science*, 145–154.

Li, C.; Palanisamy, B.; Xu, R.; Xu, J.; and Wang, J. 2021. Steemops: Extracting and analyzing key operations in steemit blockchain-based social media platform. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, 113–118.

Liu, Y.; and Bakici, T. 2019. Enterprise social media usage: The motives and the moderating role of public social media experience. *Computers in Human Behavior*, 101: 163–172.

Massey, D. S.; and Denton, N. A. 1988. The dimensions of residential segregation. *Social forces*, 67(2): 281–315.

Mozafari, M.; Farahbakhsh, R.; and Crespi, N. 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. *PloS one*, 15(8): e0237861.

Neelakandan, S.; and Paulraj, D. 2020. A gradient boosted decision tree-based sentiment classification of twitter data. *International Journal of Wavelets, Multiresolution and Information Processing*, 18(04): 2050027.

Nezami, Z.; Zamanifar, K.; Djemame, K.; and Pournaras, E. 2021. Decentralized edge-to-cloud load balancing: Service placement for the Internet of Things. *IEEE Access*, 9: 64983–65000.

Prodan, R.; Saurabh, N.; Zhao, Z.; Orton-Johnson, K.; Chakravorty, A.; Karadimce, A.; and Ulisses, A. 2019. ARTICONF: towards a smart social media ecosystem in a blockchain federated environment. In *European Conference on Parallel Processing*, 417–428. Springer.

Rajarajeswari, S.; Naik, S.; Srikant, S.; Sai Prakash, M.; and Uday, P. 2019. Movie recommendation system. In *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018, Volume 1*, 329–340. Springer.

Raman, A.; Joglekar, S.; Cristofaro, E. D.; Sastry, N.; and Tyson, G. 2019. Challenges in the decentralised web: The mastodon case. In *Proceedings of the internet measurement conference*, 217–229.

Rendle, S.; Krichene, W.; Zhang, L.; and Anderson, J. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 240–248.

Tang, W.; Zhao, X.; Rafique, W.; Qi, L.; Dou, W.; and Ni, Q. 2019. An offloading method using decentralized P2P-enabled mobile edge servers in edge computing. *Journal of Systems Architecture*, 94: 1–13.

Trienes, J.; Cano, A. T.; and Hiemstra, D. 2018. Recommending users: whom to follow on federated social networks. *arXiv preprint arXiv:1811.09292*.

Walia, K.; and Raghwa, N. 2022. Social Networking in an Information-Centric System with Blockchain. In *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, 175–182. IEEE.

Wei, Y.; and Tyson, G. 2024. Exploring the Nostr Ecosystem: A Study of Decentralization and Resilience. *arXiv preprint arXiv:2402.05709*.

Wright, R. E. 1995. Logistic regression.

Yang, D.; Qu, B.; and Cudré-Mauroux, P. 2018. Privacy-preserving social media data publishing for personalized ranking-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(3): 507–520.

Zhang, N.; and Izquierdo, E. 2023. A four-point camera calibration method for sport videos. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhang, S.; Tong, H.; Xu, J.; and Maciejewski, R. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1): 1–23.

Zuo, W.; Raman, A.; MondragÓN, R. J.; and Tyson, G. 2023a. A First Look at User-Controlled Moderation on Web3 Social Media: The Case of Memo. cash. In *3rd International Workshop on Open Challenges in Online Social Networks*, 29–37.

Zuo, W.; Raman, A.; Mondragón, R. J.; and Tyson, G. 2023b. Set in Stone: Analysis of an Immutable Web3 Social Media Platform. In *Proceedings of the ACM Web Conference 2023*, 1865–1874.

8 Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes. It adheres to ethical principles, as it advances science without violating social contracts.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes. We claim contributions and scope in section 1.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes. We briefly summarize this in Section 1.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? No, because our dataset is public and unchangeable stored on the blockchain.
   (e) Did you describe the limitations of your work? Yes. We mention the cold starting issue in Section 5.
   (f) Did you discuss any potential negative societal impacts of your work? No, because we do not mention any contentious social issues.
   (g) Did you discuss any potential misuse of your work? No, because our work is the public accessible dataset.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? No, because the analysis code is proprietary.

(i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes. I read them and ensure that our paper fully conforms to them.

2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? Yes. we clearly state the assumptions underlying all theoretical results in Section 1.
   (b) Have you provided justifications for all theoretical results? Yes. We provide justifications for all theoretical results in Section 3,4,5.
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? Yes. We discuss competing hypotheses that challenge or complement our theoretical results in Section 3.
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? Yes, we considered alternative explanations in Section 5.
   (e) Did you address potential biases or limitations in your theoretical framework? No, because we state potential biases of our work in Section 7 as future work.
   (f) Have you related your theoretical results to the existing literature in social science? Yes. We list existing literature related to our results in Section 6.
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? Yes. We discuss the implications in Section 1, 7.

3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA
   (b) Did you include complete proofs of all theoretical results? NA

4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes. We include the instructions to reproduce the main experimental results in Section 5
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? No. because we focus on the model results rather technique details.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider) No.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes. We justify in Section 3, 5.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? No.

5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes.
   (b) Did you mention the license of the assets? NA
   (c) Did you include any new assets in the supplemental material or as a URL? NA
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? No. There is no identifiable information in our dataset.
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR? NA
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset? NA

6. Additionally, if you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots? NA
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA
   (d) Did you discuss how data is stored, shared, and deidentified? NA

A Appendix

A.1 Ethics Statement

We utilize publicly accessible post records from `memo.cash`. `memo.cash` anonymizes users using identification IDs, thereby ensuring their real-world identities remain protected. Following this, we aggregate our data and conduct analyses for their platform activities. We further note that, in the USA and UK, web crawling is protected under the law for non-commercial research purposes. Our work has key societal benefits. Work exploring content moderation offers valuable insights that contribute to fostering a more inclusive, fair, and transparent digital environment for user-level moderation on Web3.

A.2 Overview of Recommendation Models

Models	Description	Parameter	Value
Random	Generates random recommendations for users without considering their preferences.	None	
Most Popular	Recommends the globally most popular mutes to all users irrespective of their individual preferences.	None	
SVD	Constructs a matrix based on user labels, breaking down user-mute interaction matrix to acquire diagonal matrix that signifies the significance of latent features (Anwar and Uma 2021).	”learning rate”	[0.001, 0.01, 0.1]
		”epochs”	[100, 1000]
		”reg all”	[0.01, 0.1, 1]
UserKNN	An early user-based nearest neighbor approach introduced by (Resnick et al. 1994). We incorporate nearest-neighbor techniques to assess their evolution with small datasets over time, as they have proven effective in recent studies (Ferrari Dacrema et al. 2021).	”K”	[5, 10, 20, 50]
		”similarity metric”	[cosine, Pearson]
LightFM	LightFM merges collaborative filtering and content-based methodologies, leveraging user-mute interactions and mute features concurrently to generate tailored recommendations. The model employs multi-layer neural networks to capture intricate interactions between user and mute embeddings. LightFM addresses the cold start issue by integrating mute features, even for novel users (Kula 2015).	”n_components”	[10, 50, 100]
		”learning rate”	[0.01, 0.05, 0.1]
		”epochs”	[10, 100, 300]
		”loss”	[logistic, bpr, warp]
		”item_alpha”	[0.01, 0.05, 0.1]
		”user_alpha”	[0.01, 0.05, 0.1]
CNN-baseline	In CNN-based recommendation systems, the interaction between users and items is depicted as matrices, with CNN layers employed to extract pertinent features from these matrices. These learned features are utilized to formulate recommendations (Daneshvar and Ravanmehr 2022).	”batch size”	[32, 64, 128]
		”epochs”	[10, 20, 50]
		”kernel size”	[(3, 3), (5, 5)]
		”activation function”	[ReLU, Sigmoid]
		”stride”	[1, 2]
CNN-preprocess	One of CNN model variation, involves input data standardization or normalization to expedite training and enhance convergence speed.	”batch size”	[32, 64, 128]
		”epochs”	[10, 20, 50]
		”kernel size”	[(3, 3), (5, 5)]
		”activation function”	[ReLU, Sigmoid]
		”stride”	[1, 2]
CNN-normalised	One of CNN model variation. Utilizes convolutional layer normalization to accelerate training, mitigate gradient issues, and boost model stability.	”batch size”	[32, 64, 128]
		”epochs”	[10, 20, 50]
		”kernel size”	[(3, 3), (5, 5)]
		”activation function”	[ReLU, Sigmoid]
		”stride”	[1, 2]
CNN-preprocess+ normalised	One of CNN model variation. An approach that combines data preprocessing and convolutional layer normalization for improved performance and stability	”batch size”	[32, 64, 128]
		”epochs”	[10, 20, 50]
		”kernel size”	[(3, 3), (5, 5)]
		”activation function”	[ReLU, Sigmoid]
		”stride”	[1, 2]

Table 6: Overview of recommendation models.