

Analyse des offenen X-Algorithmus

TEIL I – Executive Summary

Der veröffentlichte Quellcode von X offenbart ein mehrstufiges Empfehlungssystem für die For-You-Timeline. Das System reduziert die Menge von Milliarden Tweets auf einige Tausend, versieht diese mit ca. 6 000 Features und erstellt daraus eine personalisierte Timeline in weniger als einer Sekunde ¹. Kandidaten stammen aus dem In-Netz (Earlybird-Suchindex), dem User Tweet Entity Graph, dem Tweet Mixer sowie dem Follow Recommendation Service ². Nach der Kandidatengenerierung werden umfangreiche Features hydriert; dazu gehören Tweet-Eigenschaften (Content-Typ, Medienmerkmale, Metriken), benutzer-spezifische Merkmale (Embeddings, Real-Graph-Scores), Graph-Features und zeitliche Signale ³. Die ersten Rankings erfolgen durch den **Light Ranker**, ein Logistic-Regression-Modell im Earlybird-Index ⁴. Das finale Ranking übernimmt der **Heavy Ranker** – ein MaskNet-ähnliches neuronales Netz, das die Wahrscheinlichkeit verschiedener Engagement-Arten vorhersagt und diese mittels gewichteter Summe kombiniert ⁵. Anschliessend kommen Heuristiken wie Autor-Diversität, Content-Balance (50/50 In-Network vs Out-of-Network), Feedback-Fatigue, Deduplication und Visibility Filtering zum Einsatz ⁶. Die Timeline wird danach mit Anzeigen, Who-to-follow-Modulen und Produkt-Features gemischt ⁷. Trainingsdaten, Modellgewichte und Teile des Safety-Filters sind nicht offen gelegt ⁸.

Wichtigste Ranking-Signale sind Engagement-Wahrscheinlichkeiten (Favoriten, Retweets, Replies, Clicks), Real-Graph-Scores, Embedding-Ähnlichkeiten (SimClusters, TwHIN), Aggregations über vergangene Interaktionen sowie explizites und implizites Feedback. Negative Signale wie „Show less“ oder Reports haben starke negative Gewichte im Heavy Ranker ⁹. Heuristiken wie Autor-Diversität und Content-Balance beeinflussen ebenfalls, welche Tweets ausgespielt werden ¹⁰. Da weder Trainingsdaten noch alle Gewichte verfügbar sind, bleiben Effekte mancher Features sowie das tatsächliche Verhalten der Produktionsmodelle offen.

TEIL II – Architekturübersicht

Die Systemarchitektur besteht aus drei Ebenen: **Daten**, **Modelle** und **Services** ¹¹.

- **Datenebene** – Tweeypie stellt den zentralen Dienst zur Speicherung und Hydrierung von Tweet-Metadaten bereit ¹². Die Unified User Actions liefern einen Echtzeit-Stream aller Nutzeraktionen wie Favoriten, Retweets, Replies, Follows, Klicks und Videoviews ¹³. Der User Signal Service aggregiert explizite und implizite Signale ¹⁴.
- **Modellebene** – Hierzu zählen SimClusters (gemeinschaftsbasierte Embeddings), TwHIN (dichte Wissensgraph-Embeddings) ¹⁵, Trust-and-Safety-Modelle (NSFW und Abuse) ¹⁶, Real Graph (Vorhersage der Interaktionswahrscheinlichkeit) ¹⁷, TweepCred für Nutzerreputation sowie Recos Injector und Graph Feature Service ¹⁸.
- **Serviceebene** – Die For-You-Pipeline besteht aus Kandidatengeneratoren, Ranking-Services, Heuristiken und Mixer. In-Network-Tweets kommen aus dem Earlybird-Suchindex ¹⁹, während der Tweet Mixer Out-of-Network-Kandidaten koordiniert ²⁰. Der User Tweet Entity Graph generiert Empfehlungen über Graphtraversen ²¹, und der Follow Recommendation Service

schlägt neue Accounts vor ²². Nach Feature-Hydrisierung erfolgt ein zweistufiges Ranking: der Light Ranker im Earlybird-Index ²³ und der Heavy Ranker (MaskNet) ⁵. Heuristiken (Autor-Diversität, Content-Balance, Feedback-Fatigue, Deduplication) und Visibility Filtering ⁶ filtern und sortieren. Der Home Mixer orchestriert das Zusammenführen der Kandidaten und mischt Tweets mit Anzeigen, Who-to-follow und Prompts ⁷.

TEIL III – Ranking-Metriken und Signale

Kategorie	Signal/Metrik	Technische Quelle	Verm...
Engagement Probabilities	scored_tweets_model_weight_fav, retweet, reply, good_profile_click etc.	Heavy Ranker README (projects/home/recap/README.md)	Wah vers Ausg Heav
Engagement-Gewichte	Gewichte 0.5 (Like) ... -369 (Report)	Heavy Ranker README	Line Enga zur E
Aggregationsmerkmale	<code>timelines.engagement.is_favorited</code> , <code>is_retweeted</code> , <code>is_replied</code> , <code>is_clicked</code> , <code>is_dwelled</code> , <code>is_dont_like</code> etc.	FEATURES.md	Rolli Retw Verw Feed Zeitf
Graph-Features	Real-Graph-Score	interaction_graph README	Boos Inter zwise
Graph-Features	GFS followee_favorites_count	graph-feature-service README	Zähl Nutz habe
Topic Relevance	Topic Social Proof Service	topic-social-proof README	Erm anha und
Embedding-Scores	SimClusters & TWHIN Similarity	System-Architektur Seite	Ähnl Twee Emb Pers
Real-time Signals	pToxicity, pBlock, Realtime retweetCount/favCount	Earlybird Light Ranker README	Echt und schn ²⁹
Heuristik – Autor-Diversität	Limitierung der Tweets eines Autors	Home Mixer README	verh Auto
Heuristik – Content-Balance	50/50 In-Network vs Out-of-Network	Home Mixer README	Bala gefo Emp

Kategorie	Signal/Metrik	Technische Quelle	Verm
Heuristik – Feedback Fatigue	Downranking nach negativem Feedback	Home Mixer README	redu Nutz like“
Heuristik – Deduplication	Entfernung bereits gesehener Tweets	Home Mixer README	verh bere
Safety/Policy	Visibility Filtering	visibilitylib README	zent Mute Com
Negative Feedback	scored_tweets_model_weight_negative_feedback_v2, scored_tweets_model_weight_report	Heavy Ranker README	stark bewi Auss
Social Proof	Anzahl Followings, die Tweet interagierten	Graph Feature Service & Topic Social Proof	dien „Gef Kont

TEIL IV – Heuristiken, Filter und Eingriffe

1. **Autor-Diversität:** Der Home Mixer begrenzt die Anzahl aufeinanderfolgender Tweets desselben Autors, um eine ausgewogene Mischung verschiedener Stimmen sicherzustellen ³⁵.
2. **Content-Balance:** Der Algorithmus zielt auf ein Verhältnis von ca. 50 % In-Network-Tweets (Accounts, denen du folgst) und 50 % Out-of-Network-Tweets ³⁶. Dieses Verhältnis wird dynamisch angepasst, wenn wenige hochwertige Out-of-Network-Kandidaten vorhanden sind.
3. **Feedback Fatigue:** Negative Feedback-Signale („Don't like“, „Not interested“) werden zeitlich aggregiert; ähnliche Autoren oder Themen werden herabgestuft ³⁷.
4. **Deduplication:** Tweets, die der Nutzer bereits gesehen hat, werden entfernt; dies gilt auch für Retweets und Quotes desselben Inhalts ³⁸.
5. **Visibility Filtering:** Blockierte oder stummgeschaltete Autoren werden konsequent gefiltert; NSFW-Inhalte, Missbrauchsmeldungen und rechtliche Takedowns führen je nach Nutzer-Einstellung zu Soft- oder Hard-Filtering ³⁹. Der Code verweist auf eine separate Visibility-Library, die aktuell überarbeitet wird und nur teilweise öffentlich ist ⁸.
6. **Ads Mixing und Module:** Werbeanzeigen, Who-to-follow-Module und Prompts werden separat gerankt und an bestimmten Positionen eingefügt ⁷. Diese Elemente werden eindeutig als „Promoted“ gekennzeichnet und unterliegen eigenen Ranking-Modellen.

TEIL V – Was die Reichweite faktisch treiben dürfte

Die Heavy-Ranker-Gewichte legen nahe, dass **Replies** und **long-form Engagement** (tiefer Klick und längere Verweildauer) stärker bewertet werden als einfache Likes oder Retweets ²⁴. Antworten, die vom Autor erwidert werden, haben ein extrem hohes Gewicht (75) und erhöhen die Sichtbarkeit massiv

⁴⁰ . Positives Engagement wirkt sich also überproportional auf das Ranking aus, während negative Feedback-Signale („Show less“, Report) stark negative Gewichte tragen und so zu Downranking oder Ausschluss führen ⁹ .

Graphbasierte Scores (Real Graph) und Embedding-Ähnlichkeiten bestimmen, wie nahe ein Tweet an deinen Interessen liegt ²⁸ . Je mehr Followings eines Nutzers einen Tweet favorisiert haben, desto grösser die Wahrscheinlichkeit, dass der Tweet erscheint ²⁶ . Explizite Signale wie Likes, Retweets, Follows und Bookmarks werden zusammen mit impliziten Signalen wie Klicks, Verweildauer und Video-Watch Time verwendet, um Nutzerpräferenzen zu modellieren ⁴¹ . Engagement-Aggregationen über kurze (30 Minuten) und lange (50 Tage) Zeitfenster fließen als Features ein ⁴² .

Autoren, die häufig negative Feedback-Signale auslösen, werden via Feedback Fatigue dauerhaft herabgestuft. Tweets von Accounts mit hoher TweepCred-Reputation oder mit vielen gemeinsamen Followern werden dagegen bevorzugt. Da die Trainingsdaten und einige internen Gewichtungen nicht öffentlich sind, bleibt unklar, wie stark bestimmte Features (z.B. Sprache, Medienformat, NSFW-Signale) in der Produktion wirken. Der veröffentlichte Code deutet jedoch darauf hin, dass qualitativ hochwertige, konversationsfördernde Inhalte die beste organische Reichweite erzielen.

TEIL VI – Grenzen der Analyse

Der offengelegte Code deckt nur Teile der produktiven Infrastruktur ab.

- **Fehlende Trainingsdaten:** Es existieren keine öffentlichen Trainingsdaten; damit lassen sich weder Feature-Verteilungen noch Verzerrungen beurteilen.
- **Modellgewichte:** Zwar sind die relativen Gewichte der Heavy-Ranker-Outputs publiziert, doch interne Gewichtungen einzelner Features sind nicht offengelegt. Ebenso fehlen Gewichtungen und Loss-Funktionen des Light Rankers.
- **Safety & Visibility:** Der Visibility-Filter ist nur teilweise veröffentlicht und enthält keine konkreten Regeldefinitionen ⁸ . Eingriffe durch Trust-and-Safety-Teams, rechtliche Vorgaben oder Revenue-Überlegungen bleiben daher intransparent.
- **Aktualität:** Einige Komponenten, insbesondere der Light Ranker, werden laut Dokumentation „seit Jahren nicht mehr trainiert“ ⁴ und könnten in der Produktion durch neuere Modelle ersetzt sein. Gewichte können jederzeit per Konfigurationsdatei angepasst werden ⁴³ .
- **Externe Module:** Anzeigen und Empfehlungen anderer Services nutzen eigene Modelle und sind nicht Teil des veröffentlichten Codes.

TEIL VII – Appendix

Wichtige Dateien, Pfade und Services

- `home-mixer/` – Scala-Dienst zur Orchestrierung der Home-Timeline ⁴⁴ .
- `src/java/com/twitter/search/` – Earlybird-Suchindex und Light Ranker ¹⁹ .
- `src/scala/com/twitter/recos/user_tweet_entity_graph/` – UTEG Graph basierend auf GraphJet ⁴⁵ .
- `tweet-mixer/` – Koordiniert Out-of-Network-Kandidaten ⁴⁶ .
- `follow-recommendations-service/` – Dienste für Accountempfehlungen ²² .
- `projects/home/recap/README.md` – Beschreibung des Heavy Rankers, Engagement-Outputs und Gewichte ⁵ .
- `projects/home/recap/FEATURES.md` – Detaillierte Liste der Aggregations- und Engagementfeatures ²⁵ .

- [home-mixer/server/src/main/scala/com/twitter/home_mixer/product/scored_tweets/param/ScoredTweetsParam.scala](#) – Konfigurationen für Kandidatenquellen und Quality-Parameter ⁴⁷ .
- [graph-feature-service/](#) – Liefert Graph-Features für Nutzerpaare ²⁶ .
- [visibilitylib/README.md](#) – Überblick über Visibility-Filtering ³⁴ .
- [topic-social-proof/README.md](#) – Beschreibt Topic Social Proof Service ²⁷ .

Glossar (Auswahl)

- **Earlybird** – Echtzeit-Suchindex für Tweets; generiert In-Network-Kandidaten und führt Light Ranker aus ¹⁹ .
- **UTEG** – User Tweet Entity Graph; GraphJet-basierter Dienst, der via Graph-Traversal Out-of-Network-Tweets erzeugt ²¹ .
- **SimClusters** – Clustering-Verfahren zur Zuordnung von Nutzern und Tweets zu Communities; liefert Embedding-basierte Ähnlichkeiten ²⁸ .
- **TwHIN** – Dichte Wissensgraph-Embeddings zur Modellierung von Nutzer- und Tweet-Beziehungen ¹⁵ .
- **Real Graph** – Gradient-Boosting-Modell zur Vorhersage der Interaktionswahrscheinlichkeit zwischen zwei Nutzern ¹⁷ .
- **Heavy Ranker** – MaskNet-ähnliches neuronales Netz; kombiniert Engagement-Wahrscheinlichkeiten mittels gewichteter Summe ⁵ .
- **Light Ranker** – Logistic-Regression-Modell im Earlybird-Index; dient zur schnellen Vorfilterung ⁴ .
- **Home Mixer** – Scala-Dienst, der Kandidaten generiert, rankt, filtert und Tweets mit Anzeigen und Modulen mischt ⁴⁸ .
- **Visibility Filtering** – Zentraler Regel-Engine zur Durchsetzung von Block, Mute, NSFW, legalen Takedowns und Downranking ³⁴ .

Praktische Implikationen für Nutzer, die organische Reichweite verstehen wollen

Die veröffentlichten Gewichte deuten darauf hin, dass hochwertige Interaktionen – vor allem Antworten und längere Verweildauer im Gespräch – die grösste Wirkung auf den Ranking-Score haben ⁴⁹ . Nutzer, die mit ihren Beiträgen Diskussionen anregen, erhöhen die Wahrscheinlichkeit einer Aufnahme in den For-You-Feed. Likes und Retweets sind zwar positiv, tragen aber deutlich weniger zum Score bei. Negatives Feedback (Show less, Reports) wirkt sich stark negativ aus und kann zur Depriorisierung führen. Ebenso spielt die Beziehung zum Publikum eine Rolle: Tweets, die von Personen aus deinem Netzwerk favorisiert wurden, steigen durch Social-Proof-Signale auf ²⁶ . Da der Algorithmus eine Balance zwischen In-Network- und Out-of-Network-Inhalten anstrebt, hilft eine gemischte Follower-Basis, Reichweite ausserhalb der eigenen Blase zu erzielen. Trotz dieser Hinweise bleiben viele Details verborgen, sodass experimentelles Vorgehen (z. B. Variation der Content-Formate und Interaktionsaufrufe) weiterhin nötig ist.

Eine maschinenlesbare Zusammenfassung wichtiger Features und Signale findest du hier: .

[1](#) [3](#) [6](#) [7](#) [10](#) [20](#) [21](#) [22](#) [28](#) [35](#) [36](#) [37](#) [38](#) [39](#) [41](#) [46](#) How the recommendation algorithm works - X Recommendation Algorithm

<https://mintlify.wiki/twitter/the-algorithm/how-it-works>

[2](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [18](#) [19](#) [45](#) System architecture - X Recommendation Algorithm

<https://mintlify.wiki/twitter/the-algorithm/architecture>

4 23 29 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm/main/src/python/twitter/deepbird/projects/timelines/scripts/models/earlybird/README.md>

5 9 24 40 43 49 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm-ml/main/projects/home/recap/README.md>

8 34 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm/main/visibilitylib/README.md>

17 raw.githubusercontent.com

https://raw.githubusercontent.com/twitter/the-algorithm/main/src/scala/com/twitter/interaction_graph/README.md

25 42 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm-ml/main/projects/home/recap/FEATURES.md>

26 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm/main/graph-feature-service/README.md>

27 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm/main/topic-social-proof/README.md>

30 31 32 33 44 48 raw.githubusercontent.com

<https://raw.githubusercontent.com/twitter/the-algorithm/main/home-mixer/README.md>

47 raw.githubusercontent.com

https://raw.githubusercontent.com/twitter/the-algorithm/main/home-mixer/server/src/main/scala/com/twitter/home_mixer/product/scored_tweets/param/ScoredTweetsParam.scala