Findings on Twitter Scraping

I am starting the research process by first focusing on the core structure and initial data points for X (Twitter) data scraping.

Scraping is an automated process of collecting data from twitter such as tweets , user Id's, hashtags etc..

There are automated tools to integrate this process , to gain insights, metrics validation, new trends, user's behaviour and market sentiments.

File format→ CSV or JSON, we can do further analysis from these formatted files.

How scraping works?

→ Using automated tools (to retrieve required data from the platform) they are bots or maybe software that sends requests to retrieve the required information.

→ Extract data (tweets→ the content of the posts, including replies, retweets, etc.

→ User profiles: bio, follower counts, usernames, etc.

→ Hashtags : data and content under specific hashtag for trend analysis

→ Engagement metric: likes, views and comments on the particular posts

we can structure the output in spreadsheets(CSV or JSON)- easy to organize and analyse

Scraping methods:

1. Twitter APIs : these APIs provides access to Twitter's data
2. Web scraping tools: software/browser extensions to simplify the process without code
3. Python libraries: snscrape and Playwright (data extraction using python libraries)

Scraping Twitter(X)

→Latest & recommended version: X API v2

→ has legal risks, privacy policy changes, paid API restrictions (due to request limits)

→ API turns into error if the limits are exceeded

How the platform detects scraping?

1. unusual requests rate limit responses

2. IP blocking :Multiple requests from one IP cause per-IP blocklists
3. Account fingerprinting : repeated automated request from same account
4. CAPTCHA'S and bot detection: detects non-human usage by giving JS challenges and captchas.

Reasons for IP/ ID blocks:

Running large scale automated scraping using logged in accounts (sudden spikes in activity).

Excessive page loads, parallel connections, or scraping pages that require JS.

Exceeding rate limits, using data center IPs, no random delays, or bot-like pattern.

An IP block happens when too many automated requests are sent from the same address, while an ID block occurs if an account or API key shows suspicious, bot-like activity. These blocks occur mainly due to exceeding rate limits, repetitive request patterns, or using flagged datacenter IPs.

To avoid them, scraping must be designed carefully by limiting request speed, introducing random delays, and using proxy rotation so the traffic appears more natural. This ensures smoother, uninterrupted data collection, which is essential for powering the matching engine.

Alternatives:

→Official X APIs

→Data resellers (choose who provide legal and legitimate data, might be costly)

→Managed scraping(pay per use) pay per 1k-10k requests

→Open Source tools, Snscrape → widely used for research purposes. Snscrape can scrape twitter data using the python library "Snscrape " to easily pull millions of historic tweets and save them off on the computer.

A cost effective alternative to Twitter's paid APIs is using open source tools. **Snscrape** can efficiently collect large volumes of tweets, profiles, hashtags, and threads without API access, while headless browsers like Playwright or Selenium can be used to capture replies, hidden comments, and JavaScript rendered content that snscrape may miss. By combining these methods, storing results incrementally, and applying techniques such as proxy rotation, user-agent switching, and random delays, data can be scraped reliably and efficiently while minimizing the risk of detection or blocking.

X for SCOUTING

Scouting on Twitter for a venture capital firm means identifying investors by analyzing bios with keywords like VC,angel investor, or partner,tracking tweets and engagements about funding or startups, and mapping networks of who follows or interacts with founders and other VCs. By combining these signals like bio keywords, activity level, and network connections a VC can filter and rank investors who best align with their stage, sector, and geography focus, making Twitter a powerful tool for discovering the right co investors. Analyzing follows and interactions helps identify groups of active investors.

Scrape tweets, post, comments

X data such as posts, replies, and hashtags is typically collected either through API or using scraping tools. The official API provides structured tweet objects containing text, author details, engagement metrics, and entities like hashtags, along with identifiers .

To scrape tweets, posts, and comments efficiently from Twitter, the most practical approach is to use tools like snscrape or headless browsers (Playwright) that can collect public data without costly APIs. Efficiency comes from batching queries, targeting specific keywords, hashtags, or user handles and storing results in a structured format (CSV, JSON, or database) to avoid re-scraping.

For replies and comment threads, fetching the conversation ID and pulling all tweets linked to it ensures complete context. Finally, efficiency requires rate control and proxy rotation so requests don’t trigger blocks, while deduplication and incremental updates keep the dataset clean and up to date.

End solution for scraping

Define Query → Scrape Tweets and Posts → Scrape Comments and Replies → Store Data → Clean & Deduplicate → Apply Rate Control & Proxy Rotation → Ready Dataset for Analysis or Investor Matching.