

# From Scraping to Bot Detection: Navigating Mastodon and the Fediverse with APIs



SAPIENZA  
UNIVERSITÀ DI ROMA

Tutti i diritti relativi al presente materiale didattico ed al suo contenuto sono riservati a Sapienza e ai suoi autori (o docenti che lo hanno prodotto). E consentito l'uso personale dello stesso da parte dello studente a fini di studio. Ne è vietata nel modo più assoluto la diffusione, duplicazione, cessione, trasmissione, distribuzione a terzi o al pubblico pena le sanzioni applicabili per legge.

- ❖ **Introduzione**
- ❖ **Metodologia**
- ❖ **Risultati**
- ❖ **Conclusioni**

# Introduzione

# Introduzione

OSN

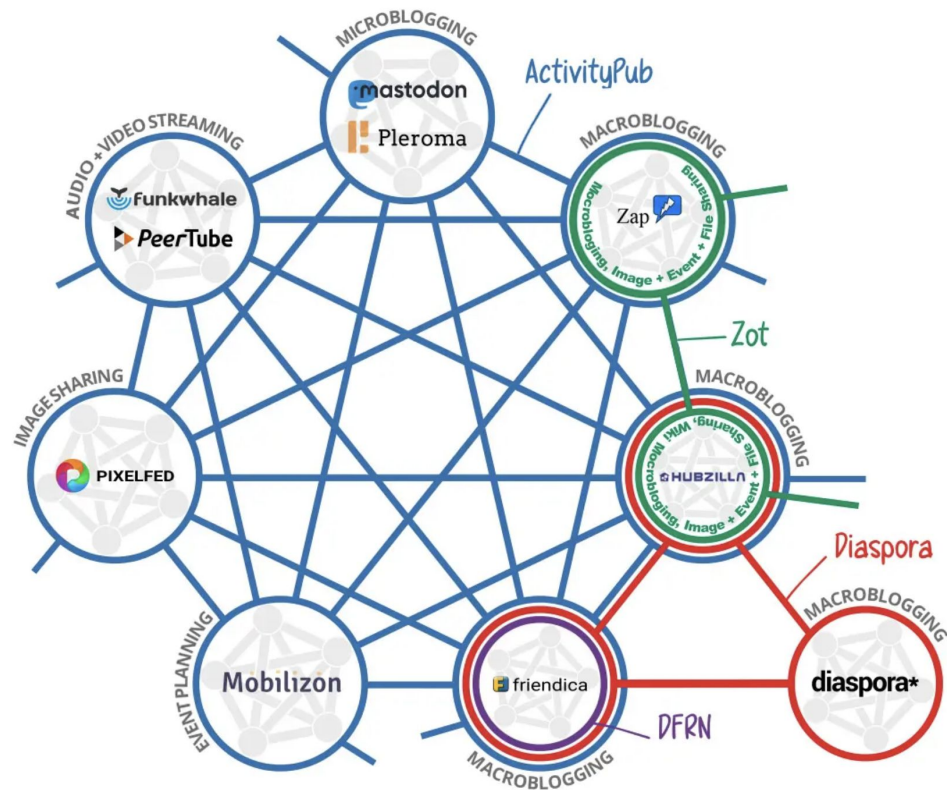


# Introduzione

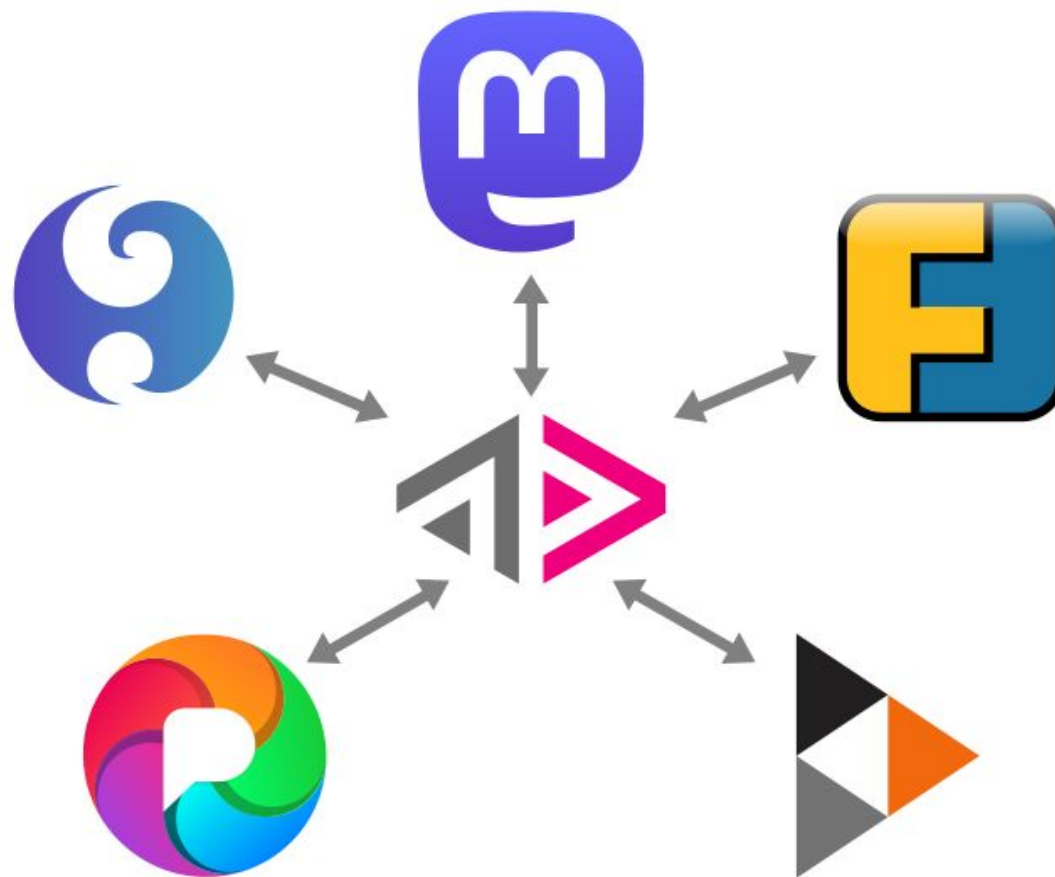
DOSN



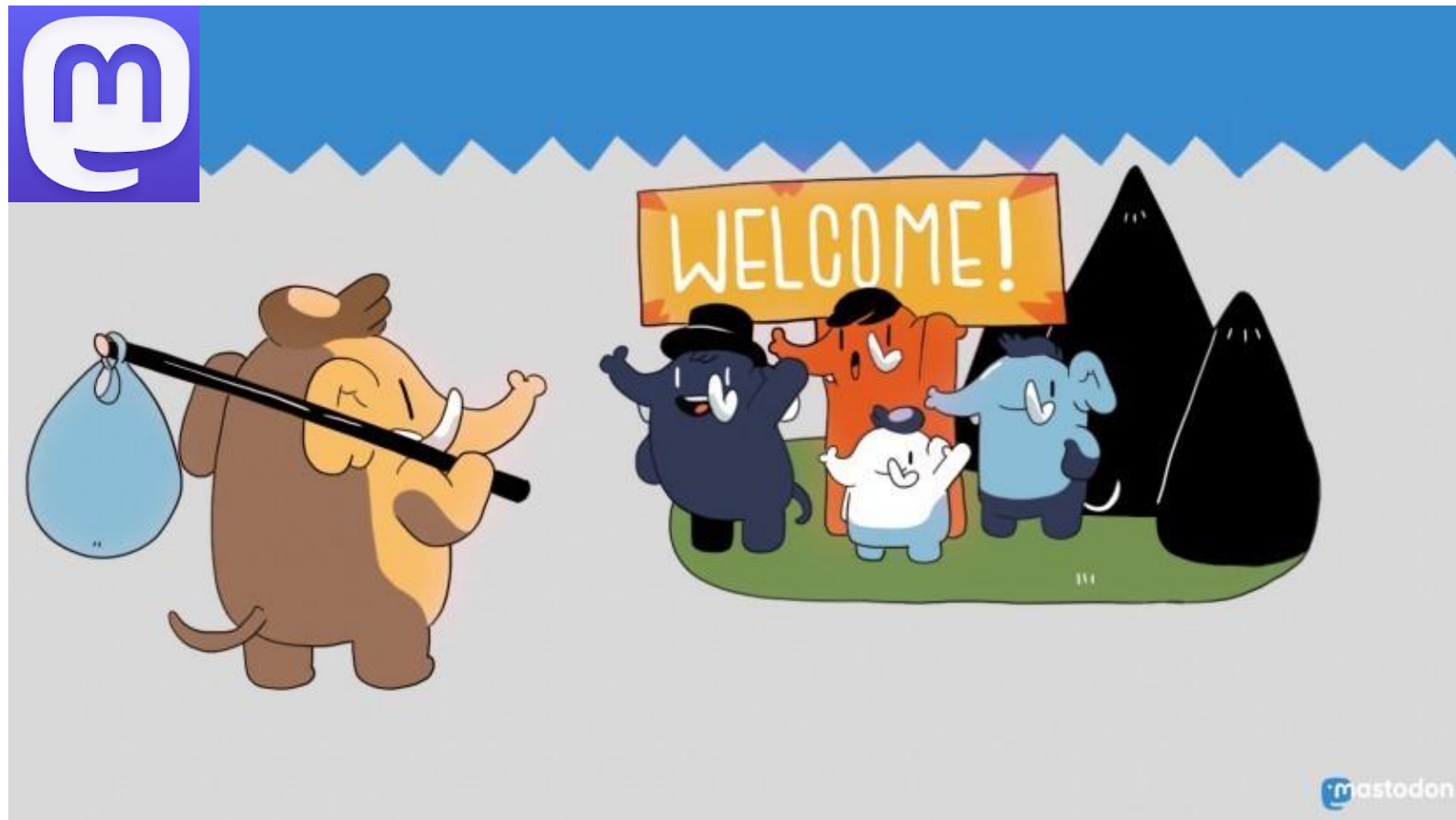
# Introduzione - Fediverso



# Introduzione - Activity Pub



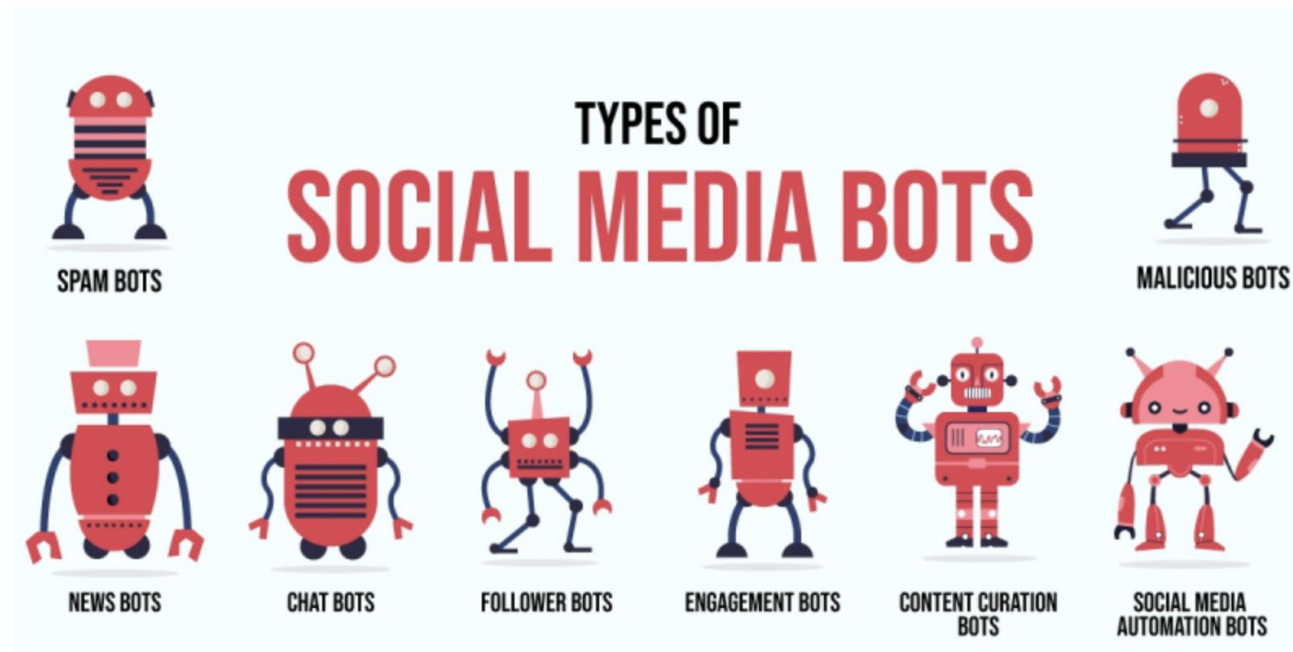
# Introduzione - Mastodon





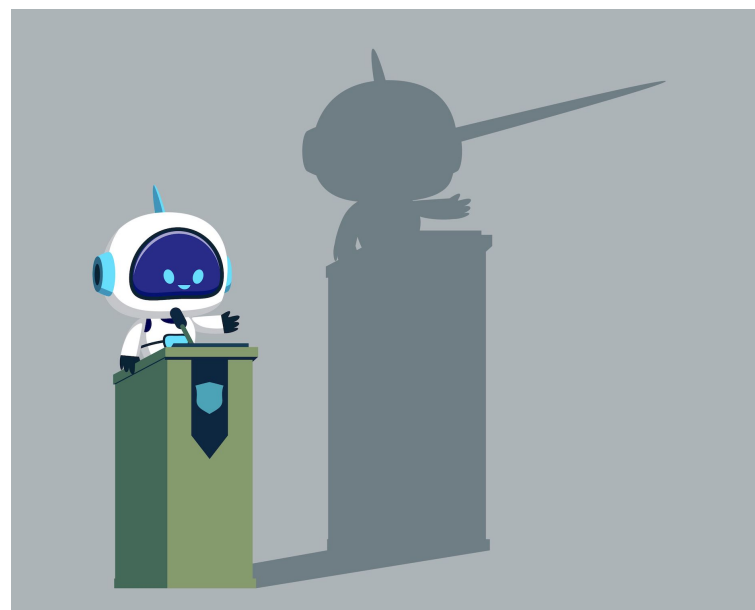
# Why Mastodon?

# Introduzione - Social bots



# Introduzione - Casi di Studio

1. Manipolazione politica
2. Campagna di disinformazione
3. Frodi finanziarie



# Introduzione - Bot Detection

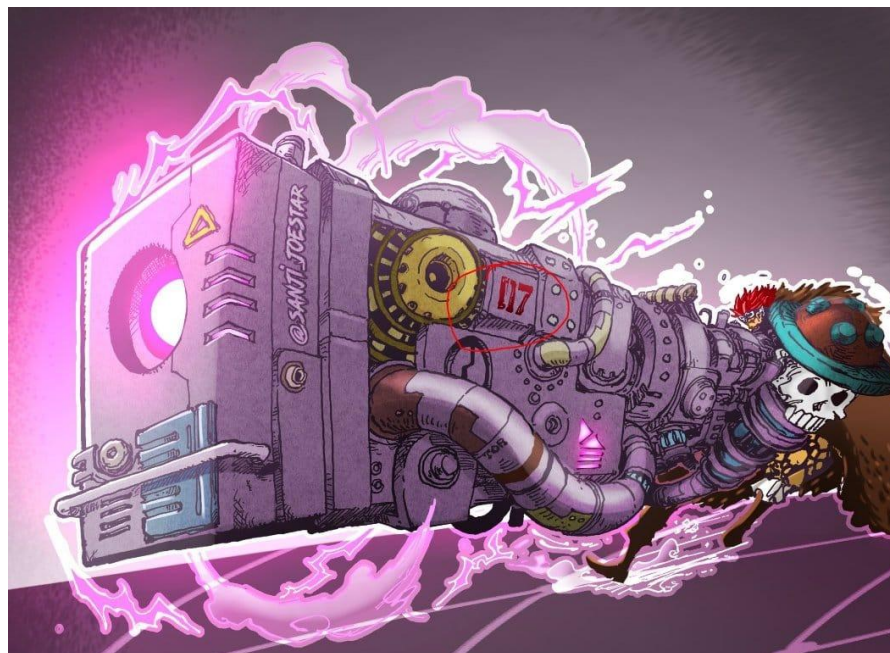
1. **Approccio euristico**
2. **Approccio multimodale**
3. **Modelli NLP**
4. **Crowdsourcing**



# Metodologia

# Metodologia - Mastroanalyzer

**Mastroanalyzer** è uno strumento di analisi che facilita la raccolta dei dati su Mastodon e supporta il rilevamento dei bot.



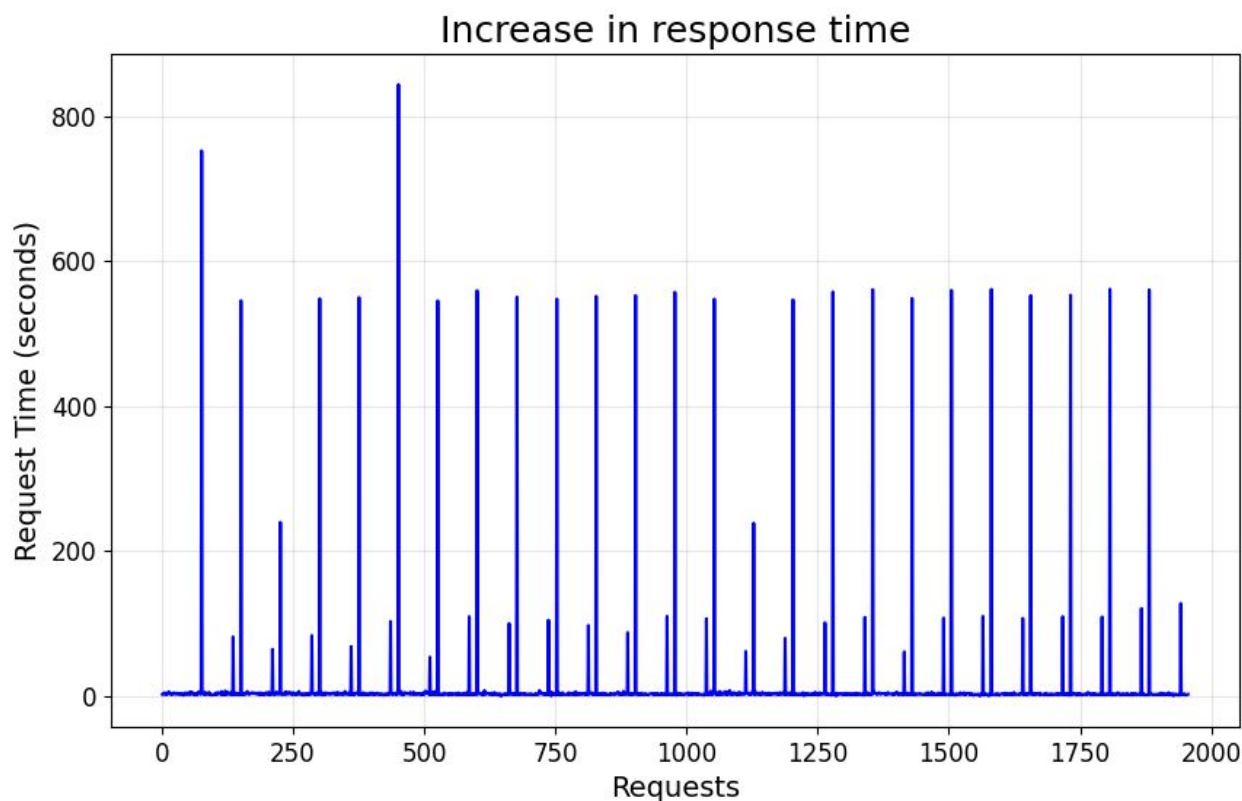
# Metodologia - Mastroanalyzer

- **Ricerca Utenti:** vengono raccolti sfruttando la timeline di Mastodon;
- **Raccolta Post:** vengono raccolti i post di ogni utente;
- **Analisi Utenti Reali:** sfruttando i dati raccolti e principi euristici alcuni utenti vengono classificati 'sospetti'.



# Mastoanalyzer 1.0

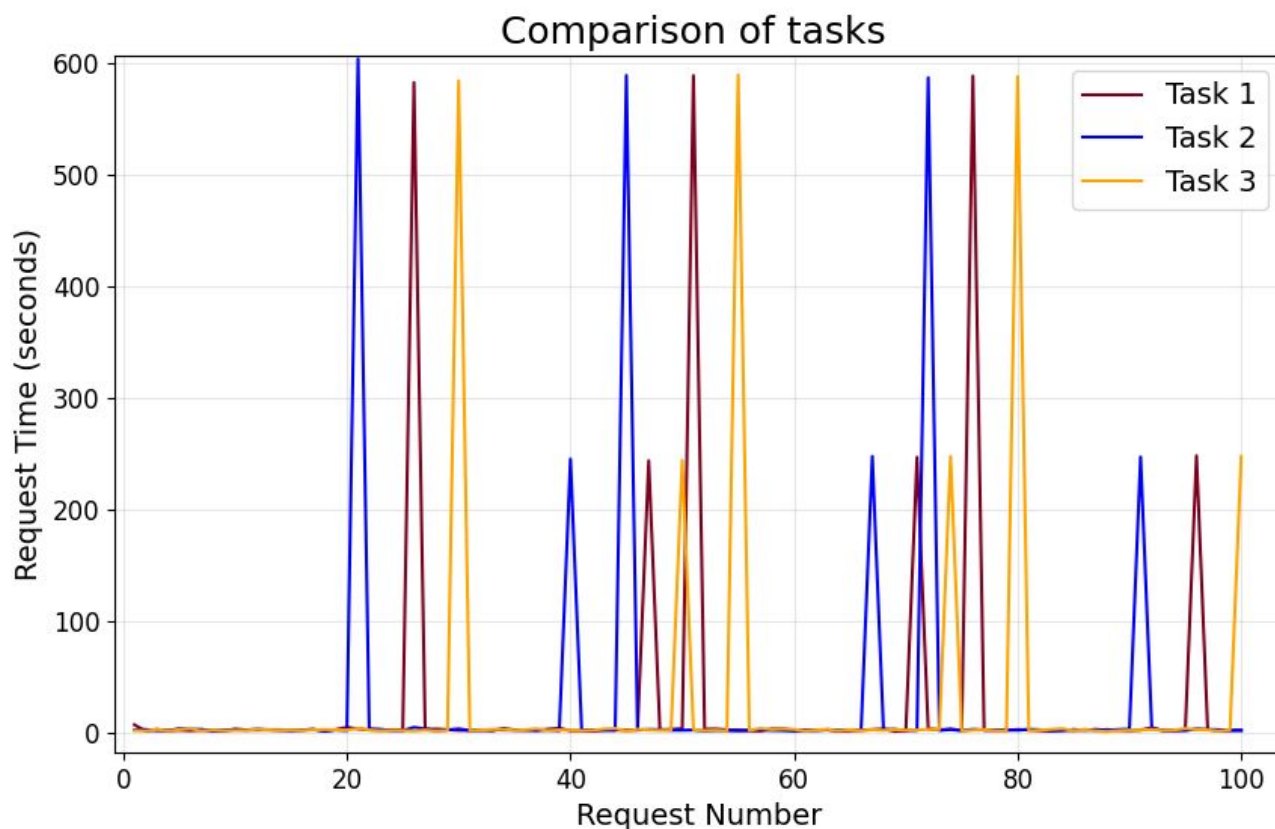
Richieste sfruttando *time.sleep()* per automatizzare l'esecuzione del programma:





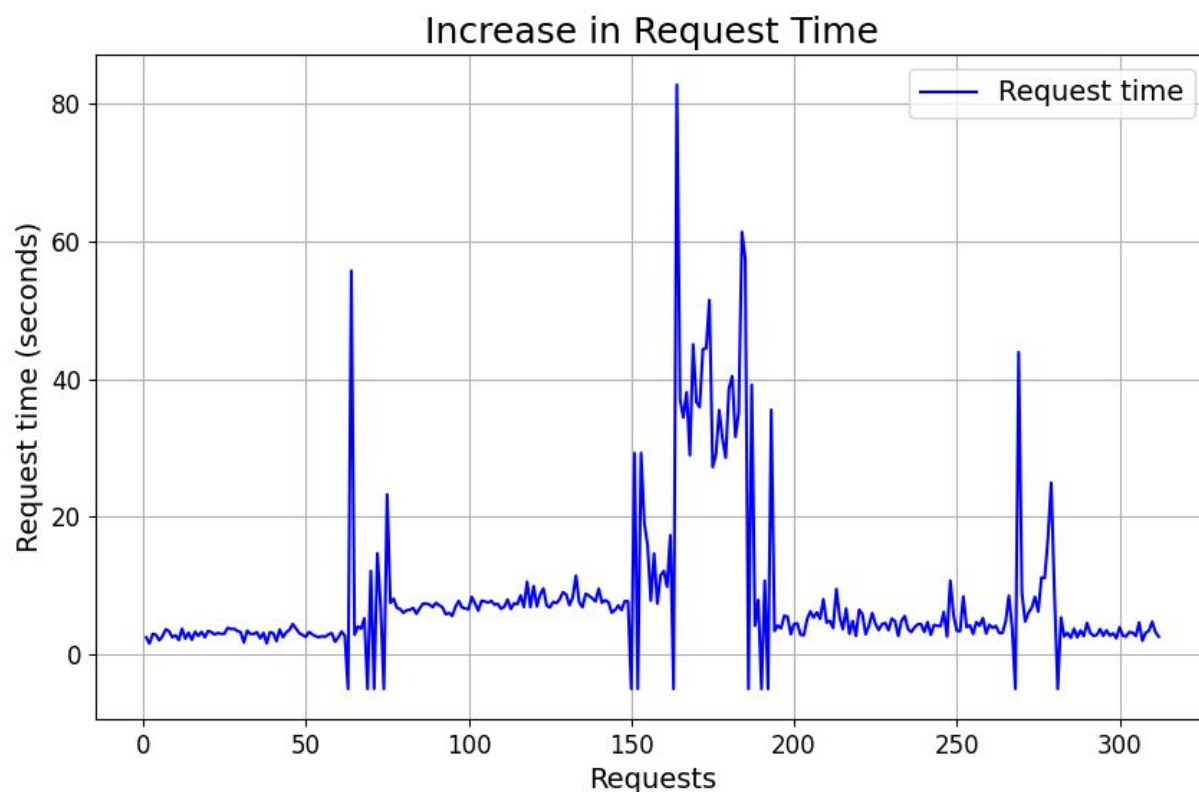
# Mastoanalyzer 2.0

Richieste *asincrone* usando solo il nostro indirizzo IP:



# Mastoanalyzer 3.0

Richieste sfruttando la rotazione di 8 *proxy* in aggiunta all'indirizzo IP:



# Mastoanalyzer 4.0

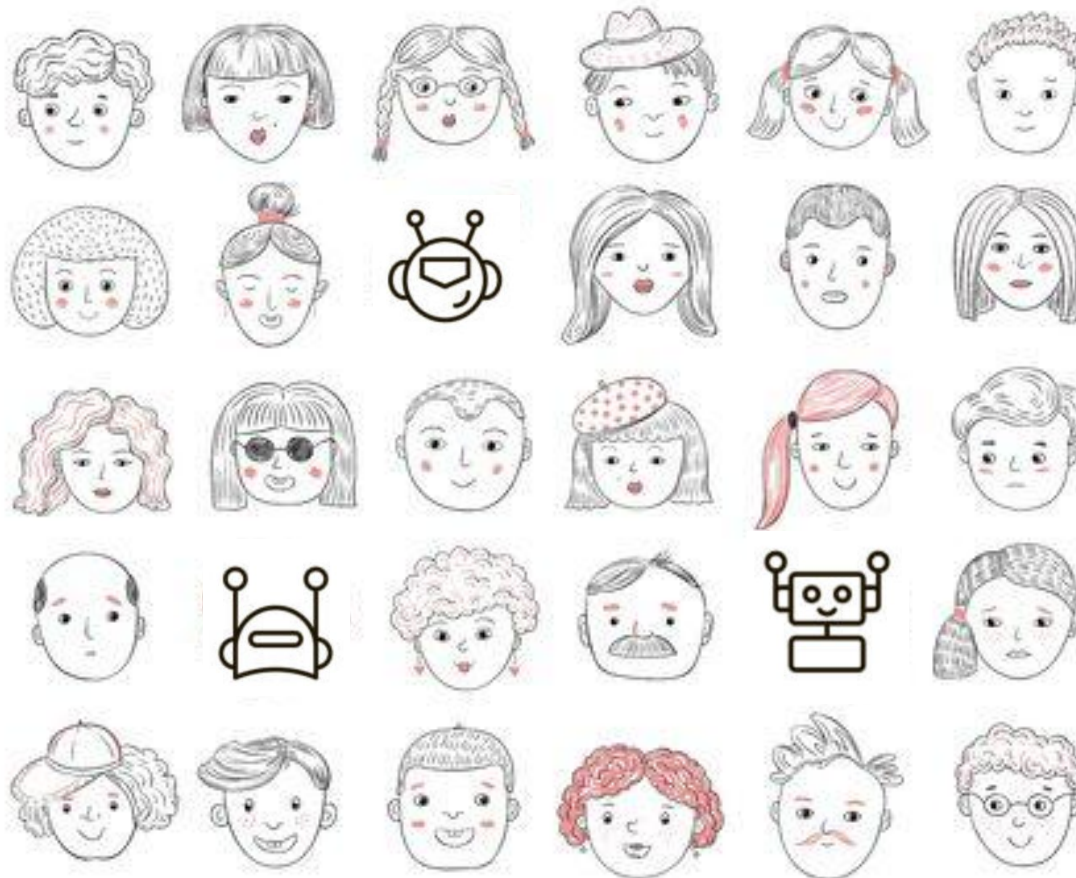
Questa versione sfrutta sia la rotazione dei proxy che le richieste asincrone.

Vengono avviate 3 *task* e ognuna sfrutta un numero variabile di proxy.

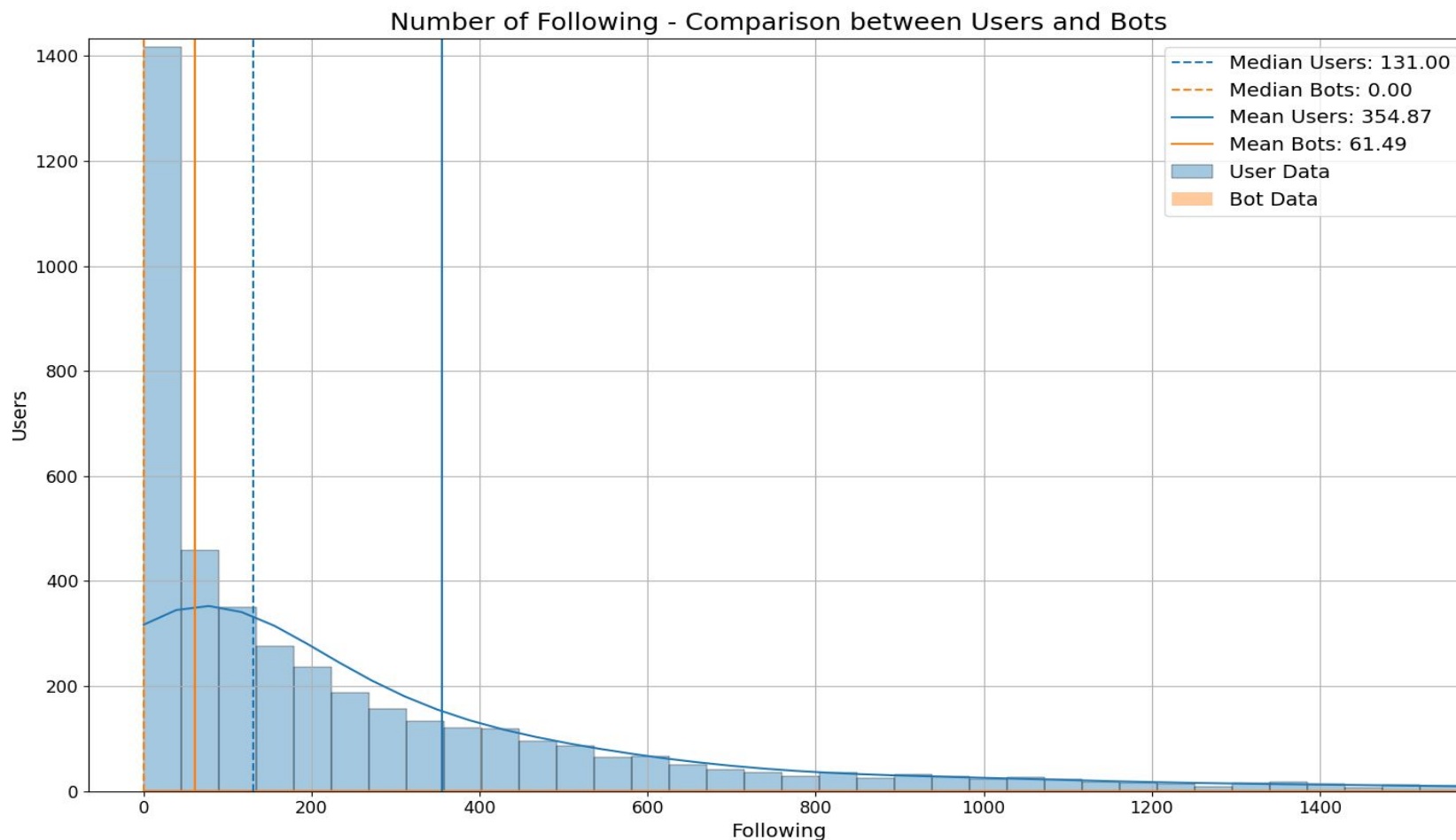
Queste task effettuano contemporaneamente richieste al server così da impiegare  $\frac{1}{3}$  del tempo necessario del grafico precedente.

# Risultati

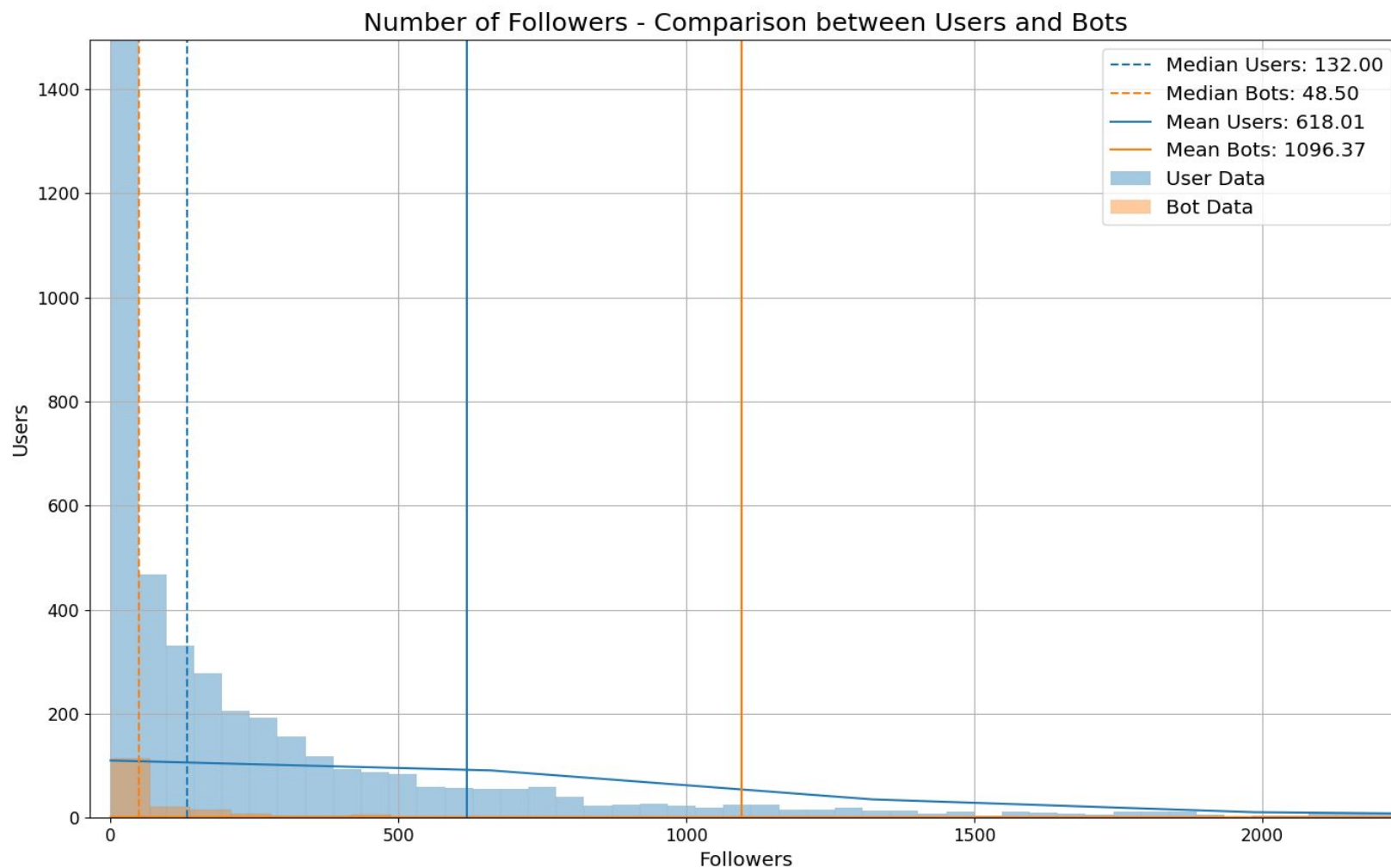
# Risultati



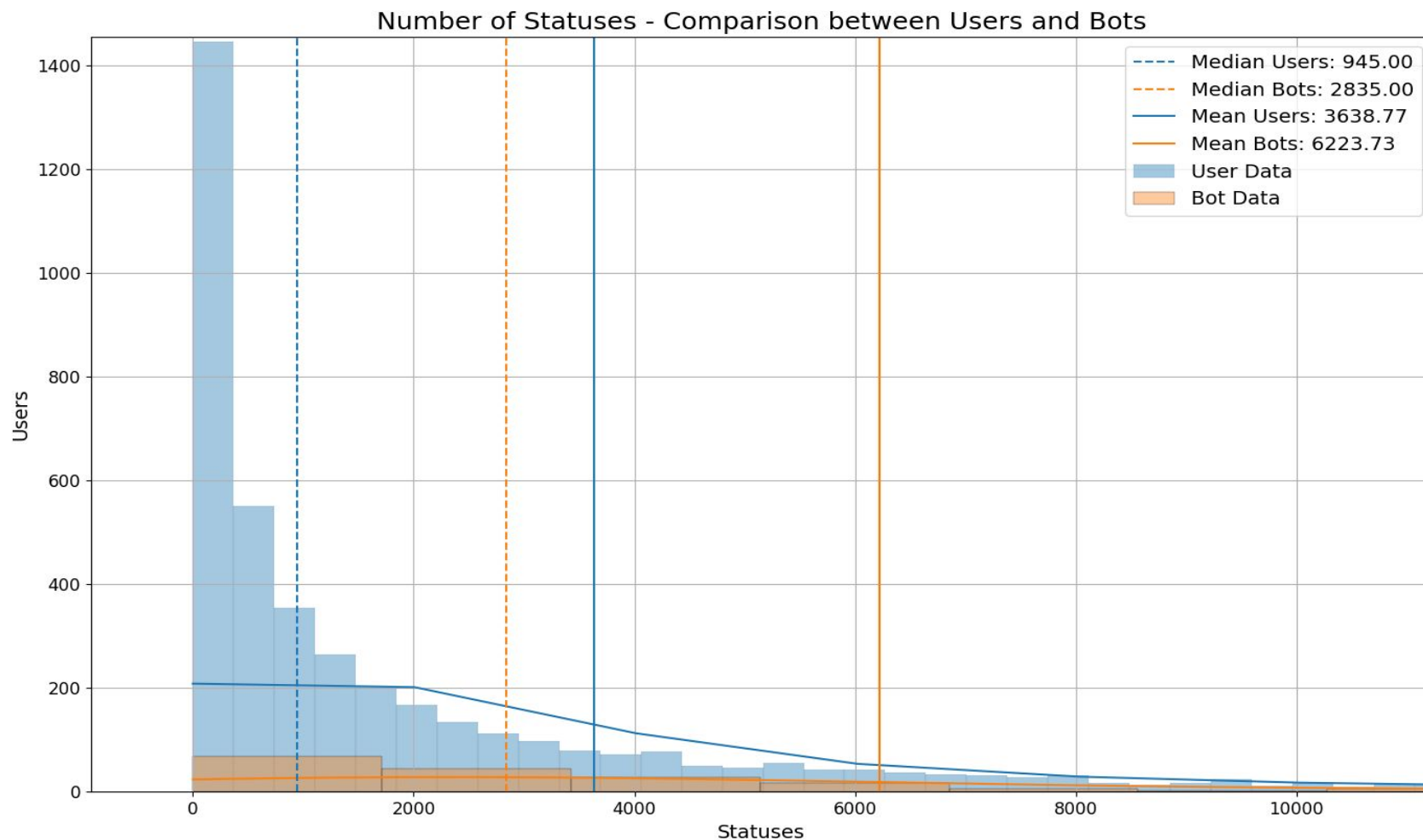
# Risultati



# Risultati

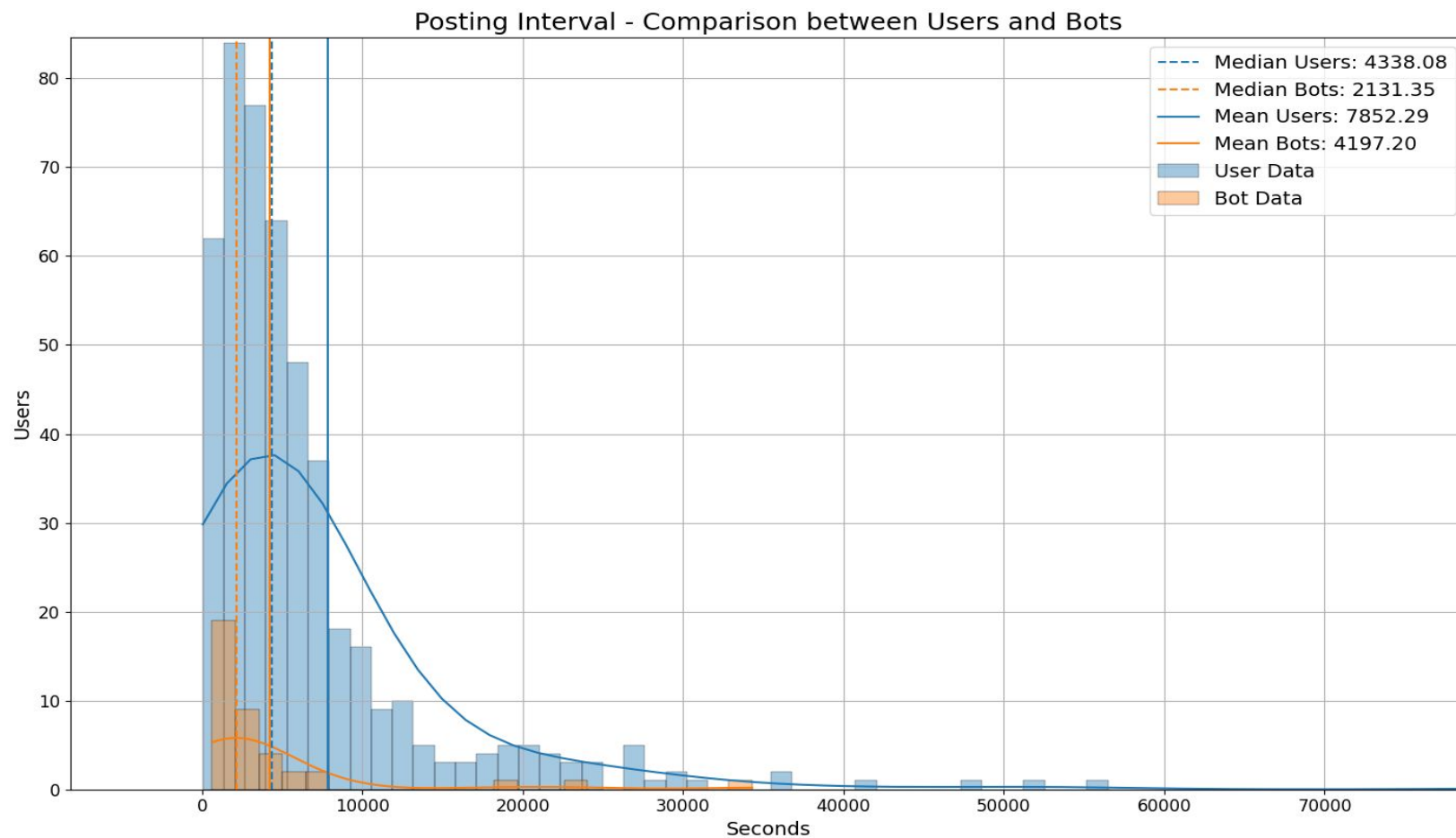


# Risultati

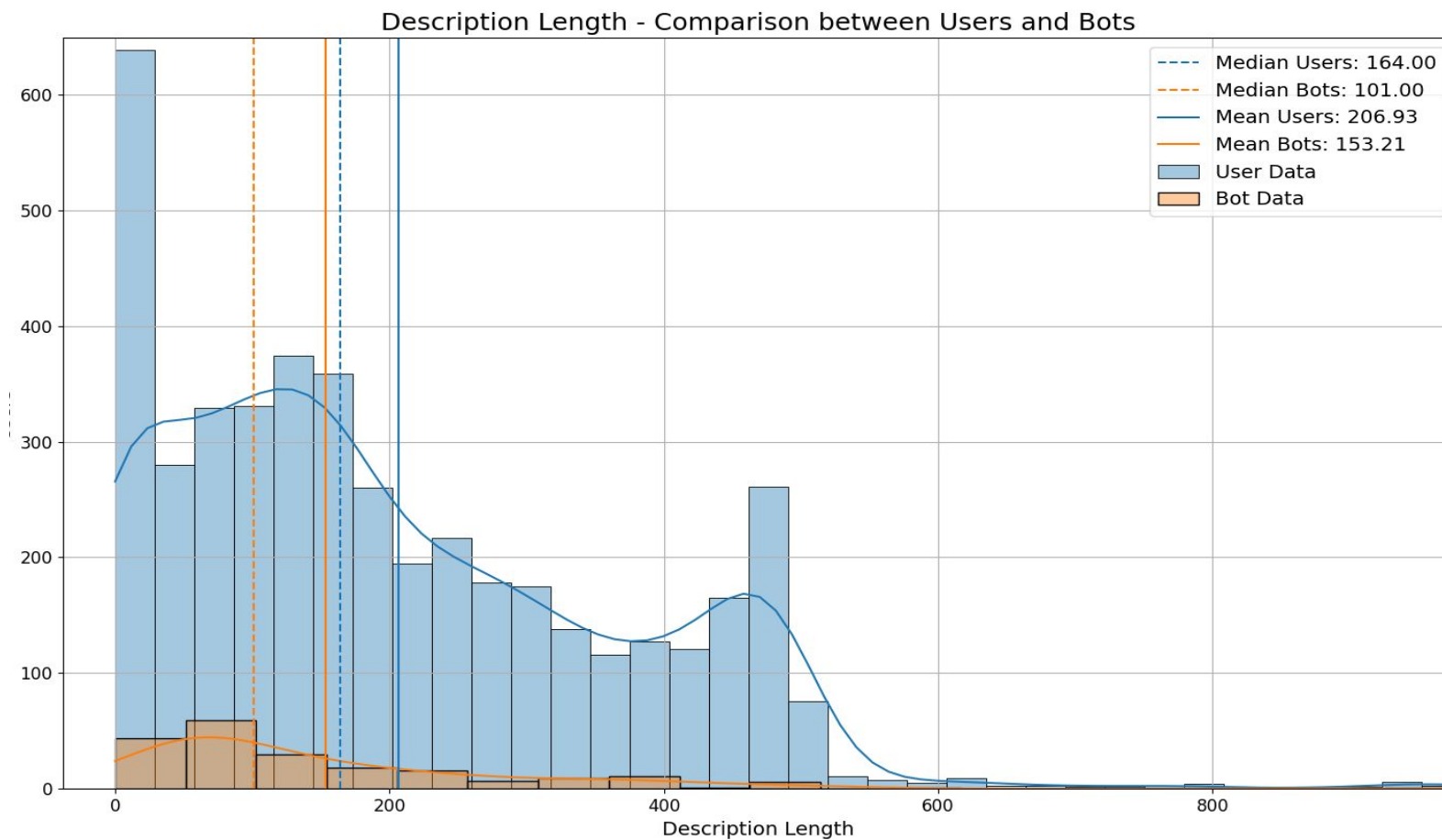




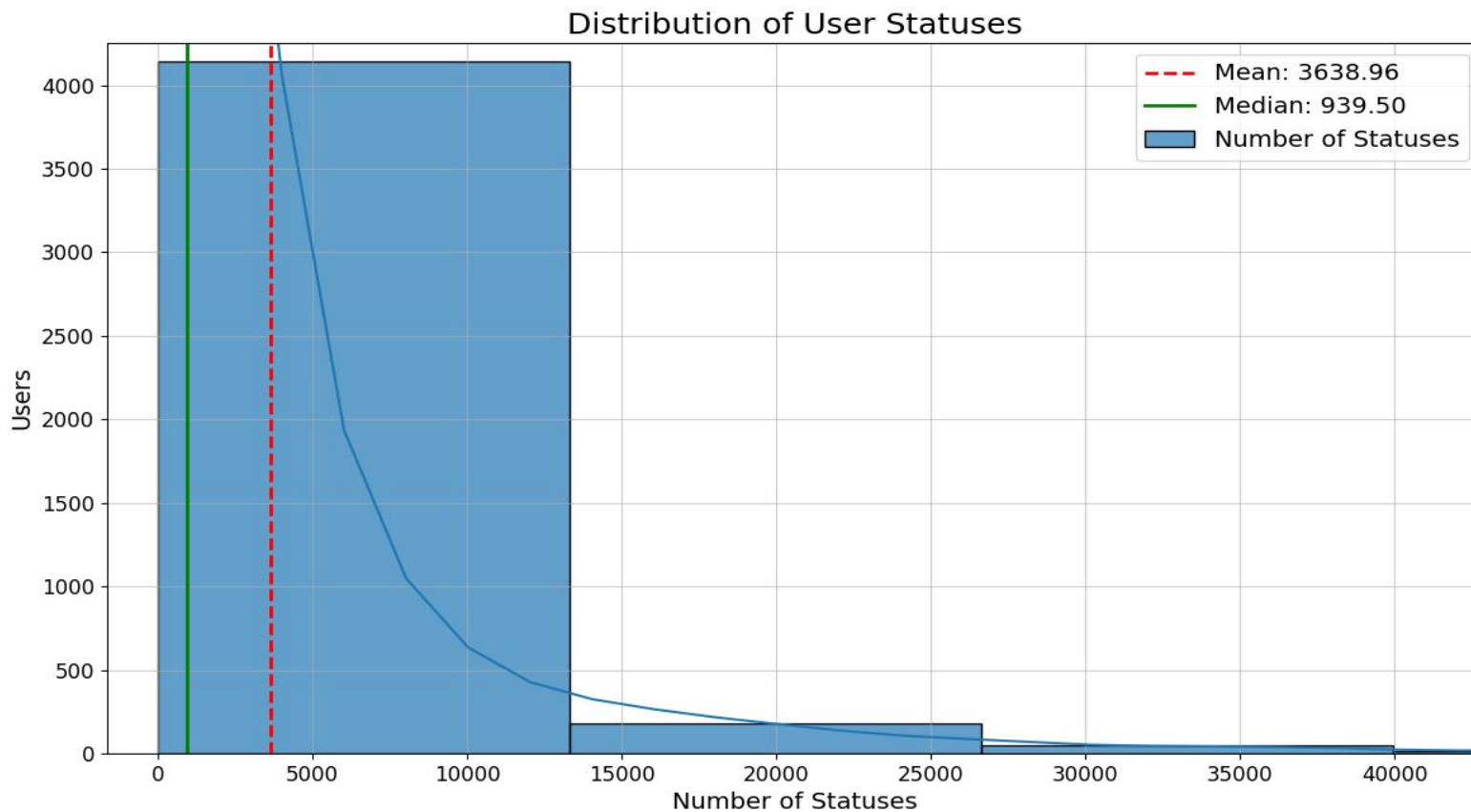
# Risultati



# Risultati

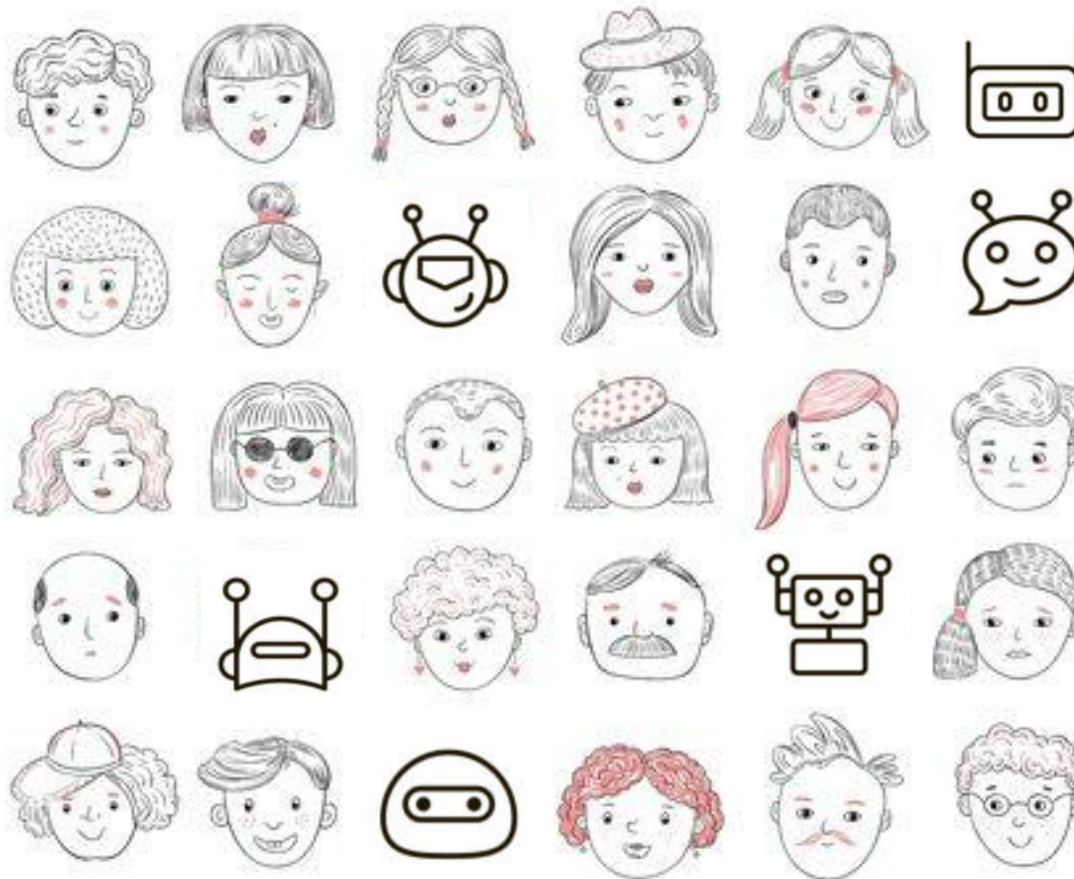


# Risultati



# Conclusioni e Lavori Futuri

# Conclusioni



# Lavori Futuri

- 1. Modelli di intelligenza artificiale**
- 2. Analisi cross-platform**
- 3. Rilevamento di attacchi coordinati**
- 4. Grafici temporali avanzati**

# Grazie per l'attenzione