

DisinfoBotWatch

Analysis and visualization of Russian Internet Research Agency (IRA) troll bot networks using Apache Spark and Flask.

Key Features

- **Large-scale Data Processing:** Apache Spark for batch processing of hundreds of thousands of tweets
- **Network Analysis:** Graph-based analysis of bot mention patterns and coordinated behavior
- **Interactive Dashboard:** Web-based visualization of analysis results with real-time data exploration
- **Account Classification:** Distribution analysis across bot types, categories, regions, and languages
- **Coordinated Behavior Detection:** Identification of synchronized posting patterns and network clusters

Requirements

System Requirements

- **Java:** JDK 17 (Spark 3.3.2 has compatibility issues with Java 25+)
- **Python:** 3.12 or higher

Software Dependencies

The project automatically installs all Python dependencies through uv. Key packages include:

- **pyspark:** 3.3.2 - Distributed data processing
- **flask:** Web framework for dashboard API
- **pandas:** Data manipulation and analysis
- **networkx:** Graph analysis and network construction
- **pyvis:** Interactive network visualization
- **numpy:** Numerical computations
- **matplotlib:** Statistical visualization

Installation

Step 1: Install Java 17

DisinfoBotWatch requires Java 17. Spark 3.3.2 has compatibility issues with later versions.

On macOS:

```
brew install openjdk@17
```

On Ubuntu/Debian:

```
# Choose appropriate version here: https://www.oracle.com/java/technologies/javase/jdk17-arm.html
# For example, with .deb
sudo apt install ./jdk-17.0.12_linux-x64_bin.deb

Verify installation:

java -version
```

Step 2: Install uv

uv is a fast Python package manager written in Rust. It automatically manages virtual environments and dependencies.

On macOS:

```
brew install uv
```

On Linux:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

With pip:

```
pip install uv
```

Or download from: <https://github.com/astral-sh/uv/releases>

Verify installation:

```
uv --version
```

Data Setup

Obtaining the Dataset

The project uses Twitter bot data from the Russian Internet Research Agency (IRA).

To download the dataset:

```
cd data
# Edit dl_data.sh and uncomment the files you want to download

# Make the script executable and run it
chmod +x dl_data.sh
./dl_data.sh
# Then go back to previous path
cd ..
```

The script will download CSV files containing tweet data and metadata from <https://github.com/fivethirtyeight/russian-troll-tweets>

Running the Project

Option 1: Complete Pipeline

This will run the analysis and start the dashboard:

```
./run.sh
```

This script automatically: 1. Checks for existing analysis output 2. If missing, runs the complete pipeline with `uv run python main.py` 3. Starts the Flask web server

The dashboard will be available at: `http://localhost:5000`

Option 2: Run Analysis Only

To run just the data analysis without starting the server:

```
uv run python main.py
```

This generates: - `outputs/top_active_accounts.csv` - Most active bot accounts - `outputs/account_distribution.csv` - Bot classification statistics - `outputs/network.html` - Interactive network graph visualization

Option 3: Start Dashboard with Existing Data

If analysis has already been run, you can start just the web server with:

```
uv run python api.py
```

Dashboard URL: `http://localhost:5000`

Dashboard Features

Overview Tab

- Total bot accounts analyzed
- Total tweets processed
- Unique account count
- Average tweets per bot

Top Bots Tab

- Ranked list of most active bot accounts
- Tweet count per account
- Account classification (type and category)
- Horizontal bar chart visualization

Distribution Tab

- Account type distribution (pie chart)
- Account category distribution (pie chart)

- Visual breakdown of bot classification patterns

Bot Network Graph Tab

- Interactive network visualization
- Node size proportional to network degree
- Edge weights representing interaction strength
- Two interaction types: mentions and coordinated posting
- Drag to pan, scroll to zoom
- Hover over nodes for account information

Analysis Details

Metrics Calculated

- 1. Basic Statistics**
 - Total tweet count
 - Unique authors and accounts
 - Distribution by type, category, region, and language
- 2. Account Activity**
 - Tweet count per account
 - Retweet vs original tweet ratios
 - Content length analysis
- 3. Coordinated Behavior**
 - Detection of identical content posted by multiple accounts
 - Identification of posting patterns indicating coordination
- 4. Network Analysis**
 - Mention network construction
 - Degree centrality ranking
 - Betweenness centrality analysis
 - Network clustering coefficient
 - Network density measurement

Troubleshooting

Issue: “Spark didn’t work with my java version”

Solution: Install Java 17 as specified in the installation section. Verify with:

```
java -version
```

Issue: Network graph not displaying

Solution: 1. Ensure analysis has completed (check outputs/network.html exists) 2. The file should be ~1MB in size 3. Clear browser cache and reload

Issue: Dashboard is slow or unresponsive

Solution: - Large network graphs are loaded lazily (only when “Bot Network Graph” tab is clicked) - Dashboard limits data display to top 100 accounts and 50 distribution items - Close unnecessary browser tabs - Ensure sufficient system RAM

References

The data is sourced from FiveThirtyEight’s publicly released IRA troll tweets dataset.

- Apache Spark Documentation
- Flask Documentation
- NetworkX Documentation
- FiveThirtyEight IRA Data

Valentin RAPP Cédric MARTZ