USENIX Security '26 Artifact Appendix: Chameleon Channels: Measuring YouTube Accounts Repurposed for Deception and Profit

Alejandro Cuevas
Carnegie Mellon University

Manoel Horta Ribeiro
Princeton University

Nicolas Christin
Carnegie Mellon University

A Artifact Appendix

A.1 Abstract

This artifact is a reproducibility package for the paper *"Chameleon Channels: Measuring YouTube Accounts Repurposed for Deception and Profit."* It includes (1) an R script to reproduce the paper's DSL-based modeling results and (2) a Python analysis pipeline to reproduce the paper's main figures (pre/post change, Sankey flow visualization, and survivability / Kaplan–Meier analysis), as well as the combined descriptive statistics table. The artifact is designed to run end-to-end on a standard Linux/macOS environment with R and Python installed, using the curated data files distributed via a restricted Zenodo repository.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

The artifact uses datasets derived from YouTube channels and Fameswap listings. Distribution is governed by YouTube's Data API policy and Fameswap's terms of service, which may inhibit public release. As a result, the replication data are distributed via a **restricted-access** Zenodo repository.

Evaluators should: (i) treat the provided data as confidential and store it locally on an access-controlled machine, (ii) avoid redistribution of the datasets and any access tokens, (iii) use the data only for artifact evaluation / academic reproduction purposes as permitted, and (iv) follow any additional conditions specified in the Zenodo record and/or accompanying documentation.

The dataset includes channel metadata such as handles and handle changes (public identifiers); evaluators should avoid re-publication of raw identifiers beyond what is required for reproducing the paper's results.

A.2.2 How to access

The data and reproduction code are available in a restricted Zenodo repository: https://doi.org/10.5281/zenodo.17955052. When requesting access, please indicate what your intended usage will be.

A.2.3 Hardware dependencies

None. A commodity laptop/desktop is sufficient.

A.2.4 Software dependencies

OS: Linux or macOS (the artifact uses Bash scripts).
R: version 4.0+ recommended.
R packages: `dsl` v.0.1.0 (see installation instructions from Egami et al. for the R package.).
Python: version 3.9+ recommended.
Python packages: installed via `requirements.txt`: `pandas`, `numpy`, `matplotlib`, `seaborn`, `plotly`, `lifelines`, `scipy`, `pyarrow`.
Bash (for running the setup script).

A.2.5 Benchmarks

The artifact reproduces results using curated datasets distributed in the restricted Zenodo repository. The main samples considered in the study are described in Appendix B.

To replicate the main findings, the artifact requires the labeled/categorized repurposed channels from Fameswap (n=1,024), labeled/categorized repurposed channels from Social Blade (n=1,047), and the Social Blade baseline sample (n=2,972), all restricted to channels with >1,000 subscribers. These data include handle changes per channel and are sufficient to reproduce: Figure 6, Figure 7, Figure 8, and Table 1.

Concretely, the repository expects the following files under `./data/`:

- `socialblade_v_baseline.csv`
- `fameswap_v_baseline.csv`
- `control_event_study.parquet`
- `treatment_event_study.parquet`
- `kaplan_data.parquet`
- `sankey_data.parquet`

A.3 Set-up

A.3.1 Installation

Obtain the artifact bundle from the restricted Zenodo repository (links above) and unpack it.

From the artifact root, create required directories:

```
mkdir -p data output
```

Place the required CSV/parquet files (listed in § Description & Requirements) into `./data/`.

Install the R dependency. Refer to the R package documentation for the latest installation instructions. At this time, the recommended installation method for (`dsl`) is:

```
if(!require(devtools))
 ↪  install.packages("devtools")
library(devtools)
install_github("naoki-egami/dsl",
 ↪  dependencies = TRUE)
library(dsl)
```

(Optional but recommended) Ensure Python can create a virtual environment and that `pip` is up to date.

After these steps, the evaluator should be able to run the basic tests below.

A.3.2 Basic Test

This basic test checks: (i) required input files exist, (ii) R can load `dsl`, and (iii) Python dependencies can be imported.

(1) Verify required data files exist From the artifact root:

```
ls -lh data/
```

Expected outcome: the directory listing includes all six required files (`*.csv` and `*.parquet`) named in § Benchmarks.

(2) Verify R dependency loads

```
Rscript -e 'library(dsl); cat("dsl: OK\n")'
```

Expected outcome: prints `dsl: OK`.

(3) Verify Python dependencies import

```
python3 -c "import pandas, numpy, matplotlib,
 ↪  plotly, lifelines, scipy, pyarrow;
 ↪  print('python: OK')"
```

Expected outcome: prints `python: OK`.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): The artifact reproduces the paper's pre/post repurposing analysis results shown in Figure 6 via Experiment (E1).

(C2): The artifact reproduces the paper's channel-handle/topic transition visualization shown in Figure 8 via Experiment (E2).

(C3): The artifact reproduces the paper's survivability analysis of channels for sale (Kaplan–Meier style) shown in Figure 7 via Experiment (E3).

(C4): The artifact reproduces the paper's combined descriptive statistics reported in Table 1 via Experiments (E4).

A.4.2 Experiments

There are four main experiments that this artifact supports. E1-3 are Python-based, and E4 is R-based. We provide E0 as a convenient script to execute E1-3.

(E0): *End-to-end Python pipeline (convenience runner) [10–20 human-minutes + 10–30 compute-minutes + ∼1–5GB disk; approximate]*
How to: Use the provided Bash script to create an isolated Python environment and run all Python analyses (E1 to E3).
Preparation: Ensure `requirements.txt` is present and all required `./data/` files exist.
Execution: `chmod +x run_python_exps.sh` and then `./run_python_exps.sh`
Results: The script should (i) create a virtual environment, (ii) install dependencies, (iii) verify required data files exist, and (iv) run `kaplan.py`, `pre_post_change.py`, and `sankey.py`.
Expected outcome: all scripts complete without error and populate `./output/` with the files needed for Figure 6, Figure 8, and Figure 7.

(E1): *Pre/Post Change (Figure 6) [15–30 human-minutes + 5–20 compute-minutes + ∼1–5GB disk; approximate]*
How to: Run the Python pre/post analysis script over the provided event-study parquet files.
Preparation: Ensure `control_event_study.parquet` and `treatment_event_study.parquet` exist and are in the `data` folder. Ensure Python dependencies are installed.
Execution: `python3 pre_post_change.py`
Results: The script should write one figure file into `./output/` corresponding to Figure 6.
Expected outcome: output files are created and are non-empty; the plot matches the paper qualitatively (same direction and shape of the effect, with the same axes/structure).

(E2): *Sankey Visualization (Figure 8) [10–20 human-minutes + 2–10 compute-minutes + ∼1GB disk; approximate]*
How to: Generate the Sankey diagram from the provided `sankey_data.parquet`.
Preparation: Ensure `sankey_data.parquet` exists, is placed in the `data` folder, and Python dependencies are installed.
Execution: `python3 sankey.py`
Results: The script should write a Sankey visualization into `./output/` (commonly `.html` for Plotly, or an exported static image if configured).

Expected outcome: the file opens correctly and visually matches Figure 8 in structure (same node groups and dominant flows). Ensure that n=374 for ideological content and n=286 for financial content.

(E3): *Survivability / Kaplan–Meier (Figure 7) [10–20 human-minutes + 2–10 compute-minutes + ∼1GB disk; approximate]*
How to: Reproduce survivability curves and comparisons using `lifelines`.
Preparation: Ensure `kaplan_data.parquet` exists, are placed in the `data` folder, and Python dependencies are installed.
Execution: `python3 kaplan.py`
Results: The script should write one or more survivability plots into `./output/`.
Expected outcome: generated curves match the figure provided in the paper (Figure 7).

(E4): *DSL Modeling (supports Table 1 and related modeling outputs) [15–30 human-minutes + 10–30 compute-minutes + ∼1GB disk; approximate]*
How to: Run the R script that executes DSL experiments over the provided CSV inputs.
Preparation: Ensure `socialblade_v_baseline.csv` and `/fameswap_v_baseline.csv` exist, are placed in the `data` folder, and R can load `dsl`.
Execution: `Rscript run_R_exps.R`
Results: The script should print numeric results in the console output.
Expected outcome: the file exists and is non-empty; console output includes experiment summaries and variables with significant effects match those reported in Table 1).

A.5 Notes on Reusability

Beyond reproducing the analyses in this paper, the provided channel-level datasets can be reused to support longitudinal monitoring of repurposed channels. In particular, the channel identifiers and handle-change metadata can be used to (i) check whether the channels in our repurposed samples have been repurposed again since the observation window of this study, (ii) assess whether their content or topical focus has shifted over time, and (iii) track additional platform outcomes such as channel suspension/termination status and monetization status (where observable via public signals or permitted interfaces).

More broadly, the artifact includes both a set of repurposed channels and a randomly selected baseline sample of non-repurposed channels (subject to the same subscriber threshold). This enables future work that compares downstream outcomes for repurposed channels against "typical" channels, including extended follow-up windows, alternative outcome definitions, and additional covariates derived from subsequent observations.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2026/.