AI-Auditology

D4.1
Audit execution and evaluation methods

Project Title	Model-based Auditing of Social Media AI Algorithms and their Tendencies to Spread Harmful Content
Contract No.	09I03-03-V03-00020
Project start date	August 2024
Duration	23 months

Funded by the European Union

RECOVERY AND RESILIENCE PLAN

Grant agreement no.:	09I03-03-V03-00020
Project acronym:	AI-Auditology
Project website:	https://kinit.sk/project/AI-Auditology
Project full title:	Model-based Auditing of Social Media AI Algorithms and their Tendencies to Spread Harmful Content
Project start date:	August 2024 (23 months)
Work package:	WP4 – Audit execution
Version:	1.0
Delivery date	June 30, 2026

Project funded by VAIA - Research and innovation authority, under the call 09I03-03-V03, Grant agreement no. 09I03-03-V03-00020		
Dissemination Level		
PU	Public	x
NP	Non-public, only for members of the consortium (including the Agency Services)	

Table of Contents

Table of Contents **3**
1. Introduction **4**
Use Case 1: Algorithmic Audit of Personalization in Polarising Topics on TikTok 4
Use Case 2: Algorithmic Audit of Advertisement Delivery to Minors on TikTok 5
2. Audit Scenarios Translation, Execution and Evaluation **7**
2.1 Audit Scenario Translation 7
2.2 User Interaction Predictors 7
Use Case: Algorithmic Audit of Personalisation Drift in Polarising Topics on TikTok 9
Use Case: Algorithmic Audit of Advertising and Minor Profiling on TikTok 11
2.3 Post-Audit Data Annotation 11
Use Case: Algorithmic Audit of Personalisation Drift in Polarising Topics on TikTok. 14
Use Case: Algorithmic Audit of Advertising and Minor Profiling on TikTok 14
2.4 Audit Evaluation 15
3. Audit Execution Engine **23**
3.1 User Agents 23
3.2 Monitoring Audit Execution 25
3.3 Architecture and Technological Implementation 27
4. Conclusion **29**

1. Introduction

This document concludes research and development activities conducted within AI-Auditology work package 4 (WP4). It follows-up on the deliverable D3.1, which introduced how audit scenarios are co-created. The proposed audit translation and execution methods (and this deliverable) covers the third and fourth step of the AI-Auditology process (see D3.1, Section 1 or our concept paper published at EWAF 2025).

This deliverable is structured as follows: In Section 2, it summarizes the *methods and experiments on audit scenario translation, execution* (the outcome of T4.1) and *evaluation* (T4.2). Section 3 consequently describes the *audit execution engine* providing the necessary user interface and support for the auditor (the outcome of T4.3). Where relevant, this deliverable is accompanied by references to the published research papers, software implementations, and published datasets.

During the AI-Auditology project implementation, we focused on audits of two platforms: TikTok and YouTube Shorts. These platforms have been selected following the qualitative analysis of stakeholders' needs and use cases (D5.1, Section 2) and technological screening of auditability of social media platforms (D5.1, Section 6). At the same time, this selection allows necessary platforms' comparability (both platforms belong to the short-video platforms) and at the same time they demonstrate generalisability (they are operated by different companies, differ in user interface, features, user interactions, etc.).

Similarly as in D3.1, individual methods and solutions are demonstrated throughout this deliverable on two specific use cases (selected from the audits conducted during the AI-Auditology project implementation), which are briefly introduced below (for further details, please, refer to the corresponding published papers):

Use Case 1: Algorithmic Audit of Personalization in Polarising Topics on TikTok

Our first use case demonstrates the audit scenario translation, execution and evaluation on the study of personalization dynamics in polarising topics on TikTok (the study was published as a full paper accepted to UMAP 2026). In this study, synthetic user accounts were created and exposed to controlled sequences of content representing different positions (supporting and opposing) on selected polarising topics (such as politics, climate change, vaccines, and flatearth conspiracy theories). By systematically simulating interaction patterns in line with the user's assigned interest, such as likes, watch time, and skips, we were able to observe how the platform adapted its recommendations over time.

The collected data revealed: 1) a *preference-aligned drift* (showing a strong personalisation towards user interests), 2) a *polarisation-topic drift* (showing a strong neutralising effect for misinformation-themed topics, and a high preference and reinforcement of interest of US politic topic); and 3) a *polarisation-stance drift* (showing a preference of oppose stance towards US politics topic and a general reinforcement of users' stance by recommending items aligned with their stance towards polarising topics).

These findings highlight several risks. First, the rapid amplification of initial preferences creates opportunities for adversarial manipulation. Malicious actors could exploit this

sensitivity by injecting targeted content early in a user's interaction history, effectively shaping their future information environment. Second, the reinforcement of homogeneous content streams may contribute to the formation of echo chambers, reducing exposure to diverse perspectives and increasing susceptibility to disinformation. Importantly, the auditing approach allowed us to quantify these effects systematically. Rather than relying on anecdotal evidence, we were able to provide reproducible measurements of personalization dynamics and their potential consequences.

Use Case 2: Algorithmic Audit of Advertisement Delivery to Minors on TikTok

The second use case presented in this deliverable focuses on the auditing of advertisement delivery to minors on TikTok (the study was published as a full paper accepted to FAccT 2026 and awarded the Best Paper award). This study specifically examined the platform's regulatory compliance with Article 28(2) of the European Digital Services Act (DSA). This regulation prohibits profiling-based advertising targeted at minors, reflecting concerns about adolescents' limited ability to recognize and resist commercial persuasion in digital environments.

In this study, we conducted an algorithmic audit using controlled sock-puppet agents simulating both minor and adult users with identical interest profiles (representing interest in 4 topical categories: beauty, fitness, gaming, and politics). These agents interacted with the platform in a systematic manner, allowing us to observe how TikTok's recommendation and advertisement systems responded to user characteristics. The collected content was automatically annotated and categorized into four types: *formal* (explicitly disclosed paid) advertisements, *disclosed influencer/promotional content*, *undisclosed commercial content*, and *non-advertising content*. Additionally, we analyzed the extent to which recommended content aligned with the predefined interests of the accounts.

The results reveal a nuanced and concerning picture. On the one hand, TikTok appears to comply with the formal requirements of the DSA by limiting profiling-based targeting in clearly labeled advertisements delivered to minors. On the other hand, this compliance is largely confined to narrowly defined *formal advertisements* (covering only advertisements purchased through a platform's ad system for which a platform is directly paid). In contrast, both disclosed and undisclosed promotional content, particularly influencer marketing, exhibit strong personalization effects aligned with user interests. Notably, the degree of profiling observed in such content is substantially higher than that seen in formal advertisements delivered to adult users.

The most pronounced effects were identified in *undisclosed commercial content*, where creators or brands fail to properly label promotional material. In these cases, the platform neither enforces disclosure nor restricts personalized delivery to minors. As a result, minors are still exposed to highly targeted commercial messaging, effectively bypassing the intended protection of the regulation.

These findings highlight a critical regulatory gap that can be interpreted as a form of systemic vulnerability. While safeguards are implemented at the level of formally recognized advertisements, functionally equivalent commercial content can circumvent these protections

due to definitional limitations. This creates an exploitable pathway for targeted influence operations aimed at minors, whether for commercial or potentially malicious purposes.

This use case demonstrates how algorithmic auditing can uncover discrepancies between formal compliance and actual system behavior. By systematically analyzing different categories of content and their personalization dynamics, the approach provides concrete evidence of how protections may be undermined in practice. More broadly, it underscores the importance of considering not only technical system design but also regulatory definitions and enforcement mechanisms when assessing the security and safety of online platforms.

2. Audit Scenarios Translation, Execution and Evaluation

2.1 Audit Scenario Translation

As abstract audit scenarios (previously described in D3.1) are platform-, time- and content-agnostic, they are not directly executable at the social media platforms undergoing the audit process. To this end, the specified users (and their demographics and characteristics), interaction and eventually also content must be translated into platform-specific representation.

At first, when performing the audit on a specific platform (or multiple ones at the same time), the user profiles must be translated to specific user accounts. This process is done semi-automatically by a researcher (an auditor). It was not possible to automatize it completely, as the account creation process may require email/phone number validation or solving captcha challenges that are difficult to fully address by a machine.

How the user interests are signaled on the specific platforms (as defined by the seed phases, as described in D3.1) is translated in this step as well. The translation depends on multiple cases, such as whether the platform allows keyword search for videos or specific URLs need to be prepared beforehand, or how the videos are presented after opening them. Currently, we are working with platforms that allow for keyword search that lists videos. As such, each such video is passed to the user interaction predictor and the user only watches the specific video corresponding to the user interest.

Finally, the user interactions are translated to the ones allowed by the platform. This mainly includes translation to specific paths to buttons and the actions needed to perform the action by means of user agents (see Section 3.1 for more details).

2.2 User Interaction Predictors

An easily extendable set of next interaction predictors allows to simulate more organic user behavior. Instead of determining user actions (e.g., a decision whether to skip or like a recommended video) randomly or by simple heuristics (e.g., if a video has any of these hashtags, watch it until the end), we introduce a possibility for more advanced decision mechanisms. Next user interaction predictors are able to fully consider characteristics of a user as well as currently displayed content to determine the next action. To this end, the content may be automatically annotated (e.g., whether it is harmful content or not), and appropriate interaction type can be determined either by a rule-based system (that reflects the intention of the audit question) or even by querying the social media model.

During the AI-Auditology project we proposed and evaluated several different types of user interaction predictors for both platforms,TikTok as well as YouTube Shorts. They have been designed to work with videos (both short and long). As such, the collected metadata are the title of the video, its description, URL and the author of the video. In addition, the voice transcript of the video is extracted via speech-to-text model that provides high-accuracy transcripts while requiring only short time to run (in terms of seconds). The speed of this

transcriber is paramount as the interaction needs to be in real time – when encountering short videos that have duration around 10-15 seconds, spending more than few seconds on the video already provides information for the recommender system (e.g., indicating that the post/video is interesting) which could be incorrectly evaluated as user interest and bias the study.

The user interaction predictor is composed of two main components:

1. *annotator*, whose main objective is to annotate the video or post with additional information; and
2. *predictor*, whose objective is to decide on the action to take.

Annotator. The collected metadata along with a transcript are passed to a large language model (LLM) for content analysis and annotation. In this step, the LLM does not provide decisions regarding the interaction, but is only used to classify the topic of the encountered video (as primary annotation) and multiple supplementary annotations, such as stance of the video, or presence of specific tags. As such, it serves as a kind of annotator tool for the video that is being processed. The prompt that is used for LLM is a result of prompt engineering effort and includes: 1) description of the task for the model (e.g., topic annotation); 2) possible answers for the model; 3) expected structure of the answer from the model; and 4) detailed description of individual possible answers to allow for better decision. Following is an example of the prompt that was used to classify topic, stance, and other supplementary information:

Topic (annotations from the LLM)	**Prompt**
Beauty (topic only)	Your task is to determine the topic and the stance of the video given its available metadata. The only possible answers for the topic is: 1) beauty; 2) other. Use only these possibilities when answering. Here is a closer description for the topics: beauty should be given to anything that is related to beauty, grooming, makeup or personal care and related products, but is not related to clothes, outfits or fashion. other should be given anything that is not related to previous topics. Provide the answer in structured form that looks like this: Topic: beauty/other Here is the video information. Author: *author name* Video description and hashtags: *video description* Video transcript:*voice transcript*
Climate change (topic + stance)	Your task is to determine the topic and the stance of the video given its available metadata. The only possible answers for the topic is: 1) climate change; 2) cooking; 3) other. For stance, the possibilities are: 1) support; 2) oppose. Use only these possibilities when answering. Here is a closer description for the topics: climate change should be given to anything related to

	climate change, global warming or policies that deal with climate change, such as the green deal or reducing co2. The support stance should be given to anything that debunks climate change deniers, claims that we need to changes, shows how the climate change affects us or supports the policies. The oppose stance should be given to anything that denies climate changes, makes fun of it, claims that the earth was always warming, claims there is cooling or opposes the policies for fighting climate change. cooking should be given to anything that is related to recipes or cooking. In this case, the stance does not matter so always put support. other should be given anything that is not related to previous topics. The stance does not matter so always put support. Provide the answer in structured form that looks like this: Topic: climate change/cooking/other Stance: support/oppose Here is the video information. Author: *author name* Video description and hashtags: *video description* Video transcript:*voice transcript*

Predictor. As such, the LLM returns only annotated values for topic, and for specific supplementary possibilities. These annotations are then passed to a predictor that decides what action should be taken. This decision can be based either on a simple heuristic (e.g., if the topic and stance of the video corresponds to the interest of the user, the interaction is *watch* and *like*) or any other more complicated or model-based solution (such as the LLM-based user simulation as described in deliverable D2.1). The currently supported interactions are: *watch, like, bookmark, skip*. For each of these possibilities, the predictor returns an integer value between 0-1 for all actions, except for *watch* action where a real number between 0 and infinity is returned. The zero indicates the action is not taken, non-zero value indicates the action is taken, while for the *watch* action the number represents the ratio of video that is watched. For example, if the value for *watch* action is returned as “4.2”, the user watches the same video 4 times and then moves on after watching it an additional 20%. However, these actions can be further extended based on the platforms.

Below we provide use cases of the user interaction predictor as part of the two selected auditing studies performed in this project.

Use Case: Algorithmic Audit of Personalisation Drift in Polarising Topics on TikTok

In this study, we employ an LLM (specifically,GPT-4.1) to accomplish the annotation step. The LLM is provided with the user characteristics (the topic of interest and stance), video URL and other video metadata obtained during the audit (title, description, author, video stickers). As many of the videos do not contain any usable author-provided description, or only very limited descriptions that do not by themselves accurately convey the topic or stance of the video, we download the audio track of the video using its URL, and then we use Whisper large-v3-turbo model to get the transcript of the audio track.

The user characteristics, video metadata and voice transcript are used to construct a prompt for the LLM. The prompt was carefully created manually to achieve the highest possible performance. We specifically focus on prompt-engineering good practices by providing the LLM with all possible options, detailed description of the topics, their different stances and how they should be assigned. The prompt is dynamically constructed based on the topic, in order to perform only a three-way classification into topic of interest, neutral topic or unrelated topic, and using only the available metadata. The answer of the LLM is then parsed to determine whether the video is relevant. When the video is related to the topic and stance of interest for the user, the action returned by the user interaction predictor is to watch the video in full, like it and bookmark it. In any other case, the video is skipped. Furthermore, livestreams and any video longer than 5 minutes are automatically skipped.

Predictor Output	**Action**	**Rationale**
topic_match: "Yes", stance: "Oppose"	Watch + Like + Bookmark	Reinforces user stance.
topic_match: "Yes", stance: "Support"	Skip after 1–2 seconds	Avoids reinforcing opposing views.
topic_match: "No", category: "Neutral Topic"	Watch + Like + Bookmark (if applicable)	Reinforces neutral interest.
topic_match: "No", category: "Unrelated"	Skip after 1–2 seconds	Ignore irrelevant content.
Live Streams or Videos >5 mins	Always skip	Avoids edge cases.

We select the best-performing LLM by evaluating and comparing the performance of multiple LLMs (GPT-4o, GPT-4.1, LLaMA-3.1, Gemma-2 and Qwen-2.5) of different sizes and different prompt templates. To achieve this, we first constructed a set of simple queries for each topic and stance (e.g., "proof earth is flat" for the supporting stance of the flat-earth topic, or “debunking flat-earth theory” for the opposing stance). Using these queries, we manually collected and annotated videos belonging to each topic and stance of interest. We collected 50 videos for each out of 4 polarising topics (with an even 50:50 split between stances), 50 videos for the neutral topic, and an additional 100 videos not related to any of the topics in the study. Using these videos, we constructed an evaluation dataset comprising 350 videos and utilised it to evaluate the LLM with the constructed prompt. The best performing model (GPT-4.1) achieves topic classification accuracy of 98%, 95%, 98% and 96.5% for flat earth, vaccines, climate change, and US politics topics, respectively. The stance classification is evaluated only on 50 videos belonging to the topic, as the stance for other videos is not relevant. In this case, we observe the accuracy to be 100%, 90%, 98% and 94% for flat earth, vaccines, climate change, and US politics topics, respectively. After deeper analysis, the errors are only due to false positives (for topic classifications) and mixed stance videos, where even human annotators had a lower agreement (for stance classification). Based on these high scores, we were sufficiently certain our user interaction predictor is capable of performing its task within the study.

We also evaluated the performance of different versions of the Whisper model (small, base, medium, large, turbo) on the same set of 350 videos. We found that adding audio transcripts to GPT-4.1 prompts, on average, increases the topic prediction by 2% and stance prediction by 6%.

Use Case: Algorithmic Audit of Advertising and Minor Profiling on TikTok

In this study, we used a similar user interaction predictor as in the previous one. For annotation purposes, we employed the same LLM (GPT-4.1), to analyze the video metadata (e.g., title and description) and determine whether the topic of the video matches the topic from the user interest profile. In this study, we decided to remove transcripts that did not yield any performance improvement this time.

Within the predictor step, if the video matched the agent's assigned interest, the agent executed the engagement routine (watch, like, and bookmark). If the content was unrelated, the agent immediately (after the time needed for its annotation) skipped the video.

2.3 Post-Audit Data Annotation

Depending on the specific purpose of the audit and audit question, the audit may require additional data annotation that is conducted after all data are collected (i.e., in a post-audit manner). In AI-Auditology, this post-audit data annotation is done automatically – this allows us to analyse a larger number of content collected during the audit execution without requiring extensive manual labour.

In contrast to the next user interaction predictor (which requires fast predictions, effectively eliminating the deployment of larger models or deeper content analysis), there is not such a time restriction for the post-audit data annotation. In practice, there are two options implemented within the AI-Auditology project. A researcher (auditor) may decide:

1. To *reuse* the annotator from the user interaction predictor as well. Practically, such annotations are stored from the audit execution phase and can be directly used.
2. To *extend* the annotator to perform additional content analysis. For example, the post-audit annotation may work with images and videos themselves (using a vision-language model instead of LLM), in order to allow for more detailed annotations and evaluation of audits.

Following is an example of the prompt that was used to classify the type and topic of advertisement present in the video content collected from the TikTok platform:

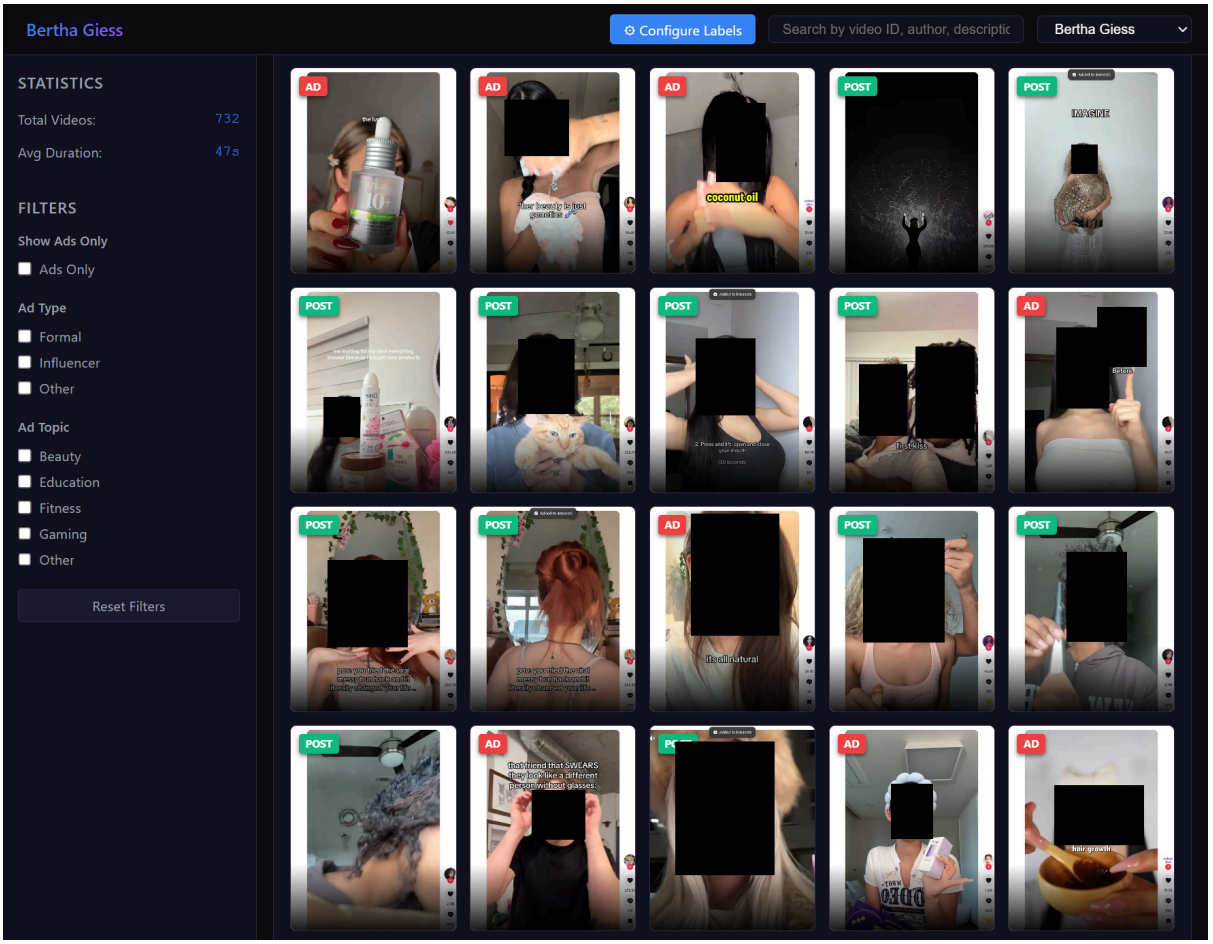
Model + Annotation	**Prompt**
VLM Ad classification (topic + ad + reasoning)	You are an assistant that identifies commercial content, both disclosed and undisclosed, in TikTok/YouTube Short videos for regulatory compliance audits. Your task is to analyze a screenshot from a short-video platform and classify it into specific categories.

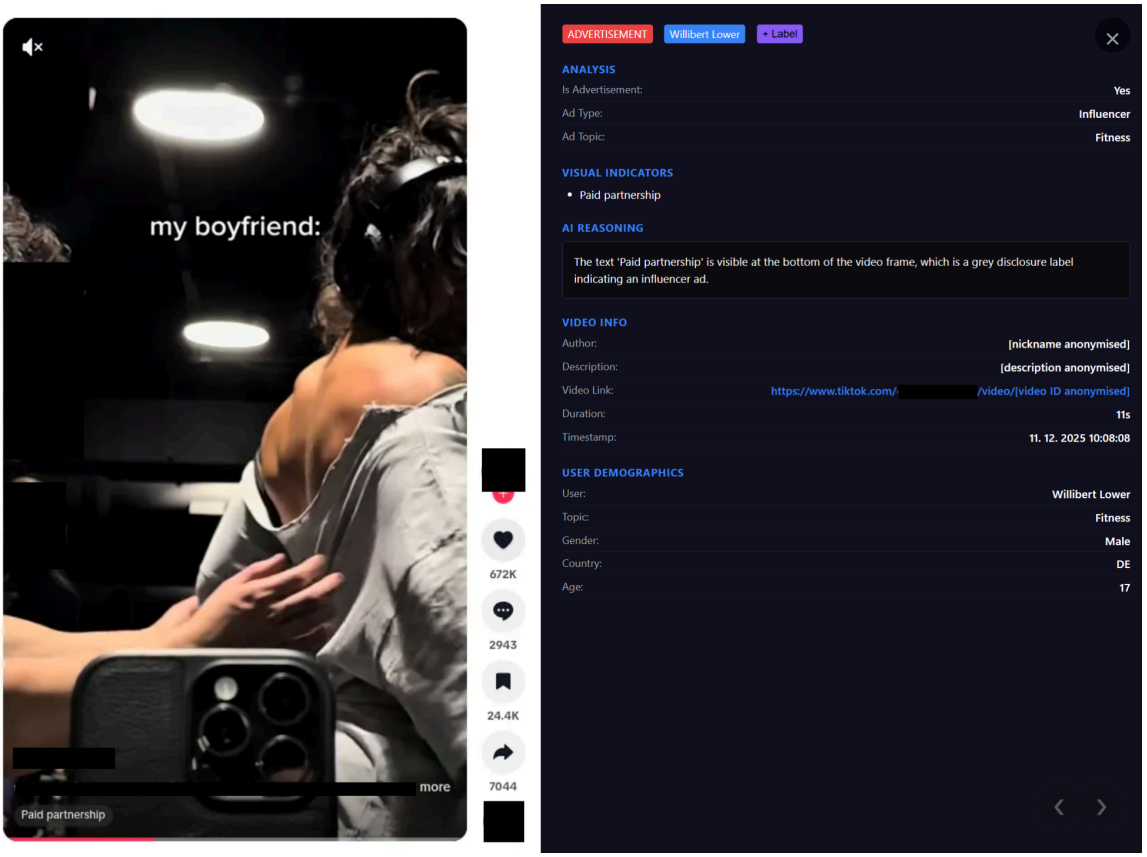
	INSTRUCTIONS:
	1. Detection: Determine if the image contains an advertisement.
	2. Type Classification: If it is an ad, classify the TYPE based on visual indicators:
	- Formal: Look for platform-injected labels "Sponsored" or "Ad" displayed on the bottom of the video.
	- Influencer: Look for creator-injected labels "Paid partnership" or "Promotional content" displayed on the bottom of the video,
	- Other: Look for videos with commercial content that lacks proper disclosure. Look for: product names or prices or discount codes mentioned/shown, brand endorsements by the creator, or promotional hashtags (e.g., #ad, #partnership, #collaboration) used without corresponding platform disclosure labels.
	3. User relevance: User is interested in beauty/fitness/gaming/fashion. Classify whether the video is relevant for this user.
	4. Topic Classification: Classify the topics of the ad into one or more of these topical categories:
	- `beauty` (makeup, skincare, cosmetics, clearskin, kbeauty, glasskin)
	- `fitness` (abs, workout, gym, sports, health, nutrition, gymtok, supplements)
	- `gaming` (video games, consoles, streamers, gamer, gaming, gamerlife)
	- `fashion` (clothes, outfit, womenfashion, wardrobe, fashion trends, outfit for specific weather or occasion, and tips and tricks for outfits)
	- `other` (anything else)
	
	RESPONSE FORMAT:
	Reply with valid JSON only.
	IMPORTANT: Do NOT use double quotes " inside your string values. Use single quotes ' instead.
	Example: "reasoning": "The text says 'Buy now' which implies..."
	
	{
	"is_ad": boolean,
	"ad_type": "formal" | "influencer" | "other" | null,
	"ad_user_relevance": boolean,
	"ad_topic": ["beauty" | "fitness" | "gaming" | "fashion" | "other" | null],
	"visual_indicators": ["list", "of", "labels", "found"],
	"reasoning": "brief explanation"
	}

As we are using automated annotation using large (vision) language models, there may be inevitable incorrect annotations. Following the proposed ethics-related mitigations, the post-audit annotation methods are evaluated for the accuracy and the expected error rate. To achieve this, we sample a random subset of the collected samples and their annotations and manually check their correctness. In case the accuracy is lower than a predefined threshold (e.g., 90% accuracy), we iteratively improve on the annotator – either by using larger or better models (which increase the cost of the annotation and the time it takes), or refining the prompt.

For the purposes of manual checks and annotations of the data, we have also developed an annotation tool that simplifies the whole evaluation workflow. The annotation tool shows the collected screenshots, metadata of the video and the annotations provided by the annotator, and allows the human annotator to assign a differ label. Example screenshots of the

annotation tool is shown below (overview of the tool and the list of video; and the detail of the video with metadata, screenshot and the predictions).


Bertha Giess
Configure Labels
Search by video ID, author, descriptic
Bertha Giess
STATISTICS
Total Videos: 732
Avg Duration: 47s
FILTERS
Show Ads Only
Ads Only
Ad Type
Formal
Influencer
Other
Ad Topic
Beauty
Education
Fitness
Gaming
Other
Reset Filters
AD
POST
coconut oil
IMAGINE
its all natural
hair growth


my boyfriend:
more
Paid partnership
672K
2943
24.4K
7044
ADVERTISEMENT
Willibert Lower
+ Label
ANALYSIS
Is Advertisement: Yes
Ad Type: Influencer
Ad Topic: Fitness
VISUAL INDICATORS
Paid partnership
AI REASONING
The text 'Paid partnership' is visible at the bottom of the video frame, which is a grey disclosure label indicating an influencer ad.
VIDEO INFO
Author: [nickname anonymised]
Description: [description anonymised]
Video Link: https://www.tiktok.com/ /video/[video ID anonymised]
Duration: 11s
Timestamp: 11. 12. 2025 10:08:08
USER DEMOGRAPHICS
User: Willibert Lower
Topic: Fitness
Gender: Male
Country: DE
Age: 17

Finally, as with the user interaction predictors, also post-audit annotators are designed to be easily extensible when the future audit studies will require different types of annotations – in each case the extension is straightforward by simply refining the prompt, and/or providing additional metadata to the model.

Below we provide use cases of the user interaction predictor as part of the auditing studies performed in this project.

Use Case: Algorithmic Audit of Personalisation Drift in Polarising Topics on TikTok

In this use case, the annotations provided by the user interaction predictor (previously described in Section 2.2) were sufficient to answer the audit question and not additional post-audit annotation was required.

Use Case: Algorithmic Audit of Advertising and Minor Profiling on TikTok

A core challenge of this audit was identifying not only formal advertisements but also disclosed and undisclosed advertisements. To address this, we developed a multi-modal classification pipeline utilizing a quantized Large Vision-Language Model (LVLM), specifically Qwen3-VL-4B-Instruct. This kind of a visual-first approach allowed us to detect regulatory compliance failures that text-only analysis would miss, particularly where overlay labels are rendered in the UI but missing from the textual metadata.

For each video, the model analysed three key frames (beginning, middle, and end) to detect the advertisement presence and eventually, classify it into three mutually exclusive ad types based on a strict decision hierarchy:

- *Formal Ads* – content where the platform injects a formal overlay label (specifically “Sponsored” or “Ad”) at the bottom of the video frame.
- *Disclosed Ads* – content where the creator has utilized the platform's disclosure tools, resulting in a “Paid partnership” or “Promotional content” grey overlay label.
- *Undisclosed Ads* – content lacking any platform-generated disclosure labels but containing strong semantic or visual indicators of commercial intent identified by the model (e.g., verbal product endorsements, visible discount codes, or hashtags).

The model also classified ad topic into one of the following options: beauty, fitness, gaming, politics or other.

To ensure validity, a subset of the automatically annotated videos was manually annotated by the research team to verify the accuracy of the detection logic. Such a manual verification was critical also to confirm that the automated annotations are aligned with human judgment, particularly for nuanced cases like undisclosed commercial content, where visual or contextual cues (e.g., subtle brand mentions or influencer partnerships) might be ambiguous. In addition, the manual evaluation aimed to ensure that the audit’s findings on minor profiling will be grounded in reliable data.

For this purpose, to achieve a representative set of data, we applied the stratified sampling strategy on the top of the collected and annotated data. More specifically, for each out of 8 users, we randomly selected 5 videos containing formal, disclosed, undisclosed, and none

ads (if there were less than 5 videos in any of these categories, all videos have been selected).

Utilizing the above-mentioned annotation tool, two authors of this study independently manually examined the selected videos and provided human-based ad type and topic annotations. They have been subsequently compared with automatic annotations. Specifically for politics topic, we observed that both users (a minor as well as an adult) got enclosed during the main audit phase in very specific topical filter bubbles (a significant portion of videos contained AI-generated satire, or videos tackling currently ongoing Pakistani political issues). The classification model did not perform well on this type of content in certain cases, and the videos were also difficult to annotate for human annotators. Therefore, we decided to remove the politics topic from further analysis.

For the remaining topics, out of 113 manually annotated videos, the automatic annotated ad type was correct in 102 (90.3%) and 100 (88.5%) of samples respectively (according to individual annotators). Out of 78/74 videos containing any type of advertisement, the ad topic was correctly annotated in 74 (94.9%) and 63 (85.1%). The inter-annotator agreement achieved high rates – 94.7% for ad type and 89.2% for ad topic. A lower agreement for ad topic was caused by videos containing fashion ads, which can be considered as closely relevant and sometimes overlapping with the beauty topic and annotators approached it in some cases differently.

Deeper insight into misclassified samples revealed that the errors occurred approximately equally between each pair of predicted-true labels, therefore, no systematic bias has been introduced. Considering the challenging nature of these annotation tasks (especially detection of undisclosed commercial content), we considered the achieved accuracy as sufficient for the purpose of this audit study.

2.4 Audit Evaluation

For every audit study, multiple information is collected and stored, including: 1) description of the sockpuppeting bot we are using (name, age, gender) mainly for mapping of resulting videos to specific users; 2) interactions performed by the user; 3) video metadata, including description, author, transcript, but also timestamp when the video was encountered, its lengths, etc.; and 4) annotations assigned by the user interaction predictor (its annotator) during after the audit or by post-audit annotation after the audit execution.

To answer the audit questions, the collected data are analysed. Every analysis is usually dependent on the audit question. For example, if we are interested in how the specific phenomenon is evolving over time (e.g., how the misinformation filter bubble is forming), we are interested in the ratio of the specific videos over time. For such analysis, we use regression analysis and observe the type and bias of the curve. For visualisation, we often aggregate videos into bins based on time intervals and use either lineplots or barplots. On the other hand, if we are interested only in the overall number of encountered videos from a specific topic, we usually just use statistical tests to measure the significance of the difference and use boxplots or simple stacked barplots for visualisation. As for metrics, we use common, simple metrics from machine learning, including the mean number of videos of

specific type, their standard deviation or a ratio of specific videos in comparison with all the remaining ones.

As the analysis is highly audit dependent, we provide more specific details for individual use cases below.

Use Case: Algorithmic Audit of Personalisation Drift in Polarising Topics on TikTok
In this audit study, we aimed to evaluate the presence and strength of personalisation drift. We achieved this by examining two aspects. First, we track the overall number of videos that belong to the topic of interest of a given user. Second, we track the number of videos with both stances towards the topic of interest.

As each user may observe a different number of videos during the day, we first collate the data from individual runs of the user as we are interested in the continuous drift over the whole time of the study. This allows us to determine whether there are pronounced periods of exploration (where the recommender tends to provide recommendations that have different topic/stance from users exhibited preferences) or exploitation (where the recommender tends to provide recommendations that have topic/stance aligned with users exhibited preferences). To perform this preprocessing, we first create intervals of 30 minutes (i.e., bins), essentially splitting a 60-minute-long daily session into two parts. Afterwards, we partition the user interactions into these bins and aggregate over them. If any video is encountered after the predefined 60 minutes in the day, it is placed into the second bin. Then, for each bin, we count the number of overall videos the user is recommended, the number of videos belonging to the topic of interest (and also to the neutral topic) and from them also the split between stances. When aggregating over multiple users, we take the ones with the same interests and add together the video counts (e.g., if we compare polarising topic, we take all users from the topic regardless what stance they were seeded with and count their videos in the given bin). Using these counts we then calculate either the ratio of videos belonging to the topic of interest from all videos that were recommended, or the ratio between different stances of the given topic.

Audit evaluation in this study proceeded from the definitions of various types of drift. These are:

- *Preference-Aligned Drift*: Tracks the ratio of videos matching the user's interest (topic + neutral) vs. unrelated videos over time.
- *Polarisation-Topic Drift*: Tracks the ratio of polarising vs. neutral videos (e.g., "climate change" vs. "cooking") over time with a value range between +1 (all polarising), -1 (all neutral).
- *Polarisation-Stance Drift*: Tracks the ratio of support vs. oppose stance videos (e.g., pro-vax vs. anti-vax) over time with value range: +1 (all support), -1 (all oppose).

For the *preference-aligned* drift we calculate the change in ratio between the personalised video recommendations (topic of interest videos plus neutral topic videos) and random videos across each bin. For the *polarisation-topic drift*, we calculate the change in ratio between the number of recommended videos from polarising topic and neutral topic across each bin. In this case, the value of 1 represents that all videos are from the polarising topic, while -1 represents that all videos are from the neutral topic. For the *polarisation-stance* drift,

we calculate the change in ratio between the number of recommended videos with support stance and oppose stance across each bin. In this case, the value of 1 represents all videos are from support stance, while -1 represents all videos are from oppose stance. We visualise the drift by fitting a regression model on the observed results. Finally, we also use the Mann-Whitney U test to determine the significance of the differences.

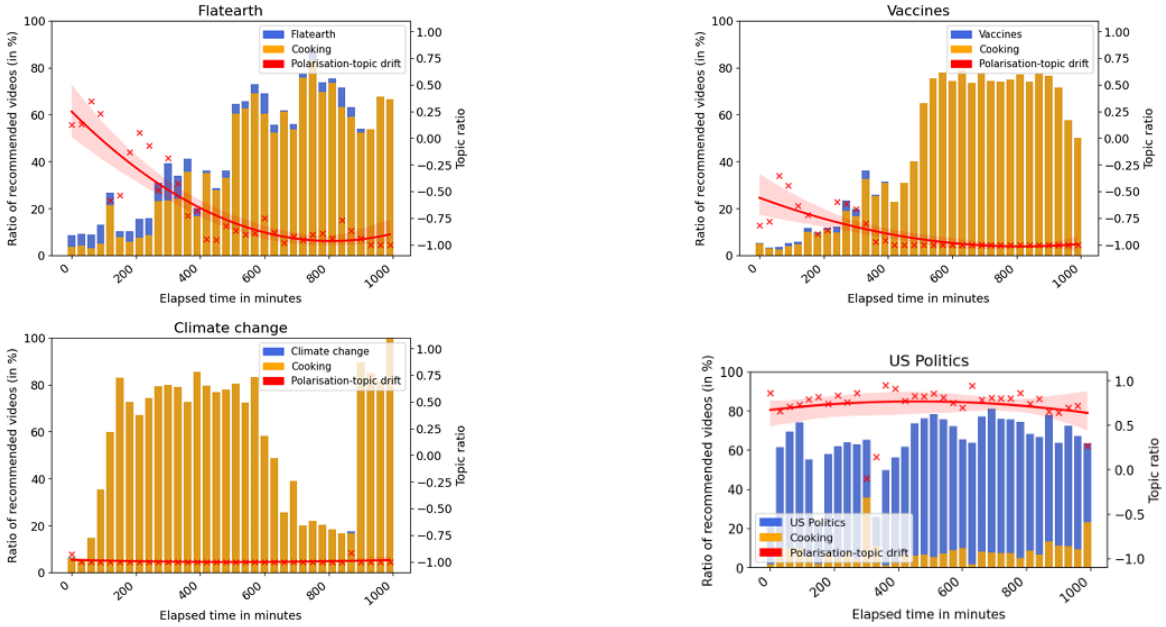
To allow replicability of our research, we published the evaluation source code on Github[1] and the resulting dataset on Zenodo[2] for research purposes only (an ethical guardrail resulting from the ethical assessment of the AI-Auditology project implementation). The following table summarise all the metadata that were collected and predicted for the observed videos:

Column name	Data type	Example	Description
interaction_number	integer	1,2,3…	unique integer per interaction per agent
video_url	string	https://www.tiktok.com/@author123	url of video the agent interacted with
video_id	string	1234	tiktok unique video id
video_author	string	author123	tiktok author name
video_description	string	This video is about…	video description generated by video author plus hashtags
video_time_duration	integer	67.9333	duration of video in seconds
video_transcript	string	Welcome to my video about…	speech transcript by inhouse whisper model
video_transcript_language	string	en, fr ….	code for language detected in transcript
video_action_skip	bool	TRUE, FALSE	decision by agent action predictor, TRUE if video is to be skipped
video_action_watch	bool	TRUE, FALSE	decision by agent action predictor, TRUE if video is to be watched
video_action_like	bool	TRUE, FALSE	decision by agent action predictor, TRUE if video is to be liked
video_action_bookmark	bool	TRUE, FALSE	decision by agent action predictor, TRUE if video is to be bookmarked
video_time_watch_loop_start	integer	1765302470.8245792	UNIX timestamp of time when agent started to watch particular video
video_time_watch_loop_end	integer	1765302470.8245792	UNIX timestamp of time when agent finished watching particular video
video_time_skip	integer	1765302470.8245792	UNIX timestamp of time when agent skipped particular video
video_time_like	integer	1765302470.8245792	UNIX timestamp of time when agent liked particular video
video_time_bookmark	integer	1765302470.82457	UNIX timestamp of time when agent bookmarked particular video

[1] https://github.com/kinit-sk/ai-auditology-personalisation-drift-tiktok
[2] https://zenodo.org/records/19144520

		92	
video_time_predict_interaction	integer	1765302470.8245792	UNIX timestamp of time when agent predicted how to interact with particular video
agent_id	string	agent_id	Unique id of a KINIT AI AGENT
topic	string	Vaccines, US Politics, Flatearth, Climate change, cooking	topic of interest of given AI AGENT
stance	string	support, oppose	stance towards the topic of interest of given ai agent
gender	string	male, female	gender set for given ai agent in tiktok
country_code	string	US	country of origin set for given ai agent
date_of_birth	string	1/2/2005	date of birth set for given ai agent in tiktok
run_id	string	1759515058.941394_main	id of given ai agent run
predicted_topic_match	bool	TRUE, FALSE	TRUE if predicted_topic == topic of interest
predicted_stance_match	bool	TRUE, FALSE	TRUE if predicted stance == stance of given agent
predicted_topic	string	vaccines, US Politics, flatearth, climate change, cooking	topic predicted by data annotator using these data fields: video_author, video_description, video_transcript
predicted_stance	string	support, oppose	Predicted stance towards the topic of interest of given ai agent. Only in *ai-auditology-drifting-study_US_politics_4_agents_mixed_polarity.csv*

As the main visualisation tool to demonstrate findings within this audit study, we employed bar charts of video counts with regression trend lines. An example of the ratio of videos for the polarising and cooking topics over time for users seeded with the polarising topics only (polarising only) can be seen on the picture below. For full results, please, refer to the corresponding paper.



Use Case: Algorithmic Audit of Advertising and Minor Profiling on TikTok

This paper's evaluation part focuses on answering the question whether TikTok's recommendation algorithm treats minors vs. adults in terms of advertisement exposure (formal, disclosed, undisclosed, non-ads).

For the purpose of this audit study, we consider the advertisement to be *personalized* to the user interest profile, if its topic matches a topic present in the used interest profile. For example if a user's profile is "fitness," and they receive a fitness-related ad, this indicates strong profiling. Profiling strength for each ad type (formal, disclosed, undisclosed) is compared between age groups (Minors vs. Adults). Profiling strength was analyzed separately for formal ads, disclosed ads, and undisclosed ads.

To operationalise such an audit evaluation, we proposed three metrics as follows. A *personalization rate* – calculated as a proportion of personalized ads out of total ads – represents a proportion of ads presented to a user that matches their interest profile. However, some number of advertisements may naturally occur in the user feed regardless of their interest profile and such a number can vary according to the popularity and overall prevalence of such advertisements in the platform. To consider this potential bias, we measure explicitly a *profiling effect*. Profiling effect is a percentage point (pp) difference between a personalization rate (i.e., a ratio between ads relevant for a user and all ads seen by the user) and a baseline rate (a proportion of ads with the same topic that naturally appeared for users with the same age but different interest profiles).

To evaluate statistical significance of profiling effect (i.e., whether a difference between personalization and baseline rate is statistically significant), we measure the p-value of proportions z-test.

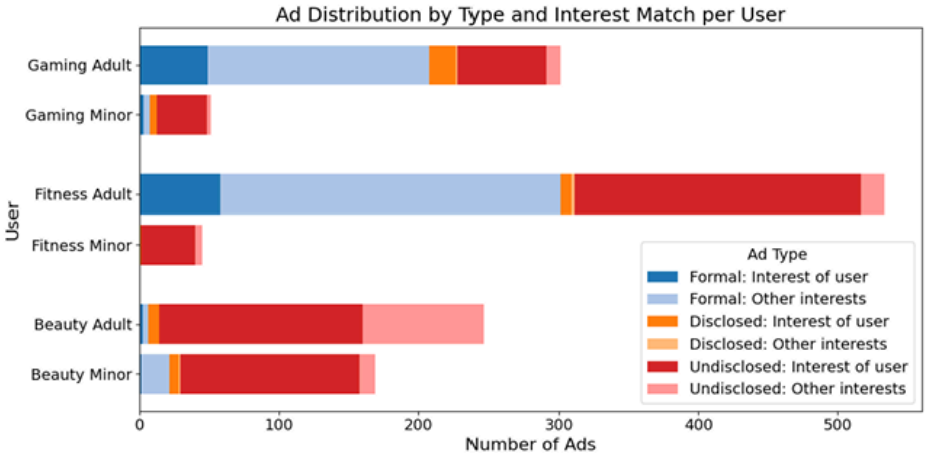
To allow replicability of our research, we published the evaluation source code on Github[3] and the resulting dataset on Zenodo[4] for research purposes only (an ethical guardrail resulting from the ethical assessment of the AI-Auditology project implementation). The following table summarise all the metadata that were collected and predicted for the observed videos:

Column Name	Data Type	Description	Example Value
session_id	string	Session identifier captured during browsing	1765302414.743265
video_id	string	Platform video identifier	[anonymized]
timestamp	datetime	Timestamp when the record was captured	2025-12-09T17:47:56.296448
is_ad	boolean	Whether the video was classified as an ad	false
ad_type	string (nullable)	Ad classification type when is_ad is true	other
ad_topic	string (nullable)	Detected topic for ad content	beauty
visual_indicators	array[string]	List of visual indicators used to classify ads	["hashtag #clearskin"]
reasoning	string	Model reasoning for the ad classification	No disclosure label visible.
interaction_number	integer	Sequential interaction count within the session	1
search_term	string	Search term used to find the content	clear skin
video_action_skip	boolean	Whether the user skipped the video	False
video_action_watch	boolean	Whether the user watched the video	True
video_action_like	boolean	Whether the user liked the video	True
video_action_bookmark	boolean	Whether the user bookmarked the video	True

[3] https://github.com/kinit-sk/ai-auditology-advertising-and-minor-profiling-tiktok
[4] https://zenodo.org/records/18879043

video_time_watch_loop_start	float (nullable)	Timestamp when watch loop started	1765302470.8245792
video_time_watch_loop_end	float (nullable)	Timestamp when watch loop ended	1765302477.842666
video_time_skip	float (nullable)	Timestamp when the video was skipped	nan
video_time_like	float (nullable)	Timestamp when the video was liked	1765302471.8269806
video_time_bookmark	float (nullable)	Timestamp when the video was bookmarked	1765302477.3054323
video_time_predict_interaction	float (nullable)	Timestamp for predicted interaction (if any)	nan
topic	string	User interest topic used for personalization	beauty
gender	string	User gender	female
country_code	string	User country code	DE
date_of_birth	date	User date of birth	2009-11-29
agent	string	Agent identifier added during processing	Beauty_minor
video_url	string	Full URL to the video	https://www.tiktok.com/[anonymized]
video_author	string	Account handle of the video author	[anonymized]
video_description	string	Video description text	little bonus - your waist? nonexistent #chiaseeds #guthealth
video_time_duration	float	Video duration in seconds	25.866667
video_transcript	string (nullable)	Auto-transcribed video text if available	nan
video_transcript_language	string (nullable)	Language of the transcript	nan

As the main visualisation tool to present the findings from this study, we opted for barcharts and confusion matrices. The following figure depicts ad distribution by ad type and ad topic match per user.

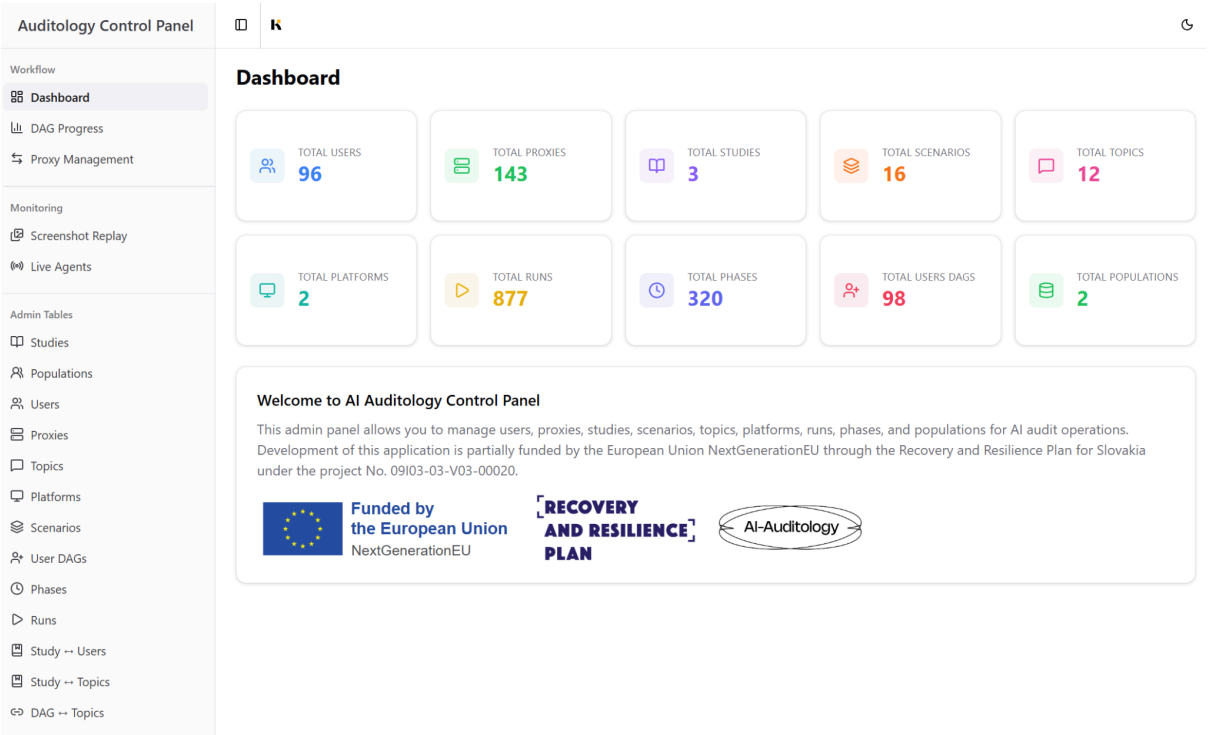


3. Audit Execution Engine

To streamline the auditing process, the AI-Auditology project designed and developed the complex software infrastructure. Within this infrastructure, the *Control Panel* represents the end-user-facing interface that ties all platform's features together and serves as a main frontend for any researcher and operator of an audit study. More specifically, it produces and maintains the audit configurations (audit scenarios), hands them over to the execution layer, and brings the resulting observations back into one place for review.

- The first part of the Control Panel supports audit scenario co-creation. It allows users to easily create, update and manage audit studies and persist them into the database.
- The second part of the Control Panel provides monitoring capabilities, exposing logs and screenshots for debugging and monitoring purposes. Live agent pods can be inspected via embedded VNC. Screenshot and log archives uploaded to AWS S3 can be browsed, extracted, and replayed frame-by-frame from inside the same UI.

The Control Panel is designed to be platform agnostic and easily extensible, currently supporting both TikTok and YouTube Shorts platforms.



The rest of this section describes the second part of the Control Panel supporting audit scenario translation, execution and evaluation. The first part was previously described in the preceding deliverable D3.1.

The Audit Execution Engine is responsible for executing the end-to-end steps of an audit within a simulated browser session. The core execution workflow consists of the following phases:

- **Implementation of User Agents and Agent Scheduling:** Based on these definitions, the engine schedules automated "sock puppet" agents to replicate human-like interactions in the browser.
- **Isolated Execution:** To ensure security, consistency, and environment reproducibility, each browser session and agent run is executed within an isolated Kubernetes pod container.

3.1 User Agents and Agent Scheduling

User agents (or sock puppets) are algorithms that control the browser session interaction. The steps performed during a session are controlled by a Scenario definition, which is stored in the pipeline configuration for the user agent to utilize. These definitions serve as interaction rules that the agent executes during each session. The pipeline configuration is expressed via an Action DSL (Domain-Specific Language), which is structurally similar to the configurations of other existing pipeline tools.

The Action DSL

The action field is not a flat configuration object. It is a small workflow DSL. The top level value is a JSON encoded ordered list of action blocks. Each block contains an ordered list of steps. A step has an id, a tool to invoke, optional args, and optional control elements. Control elements are a condition expression, a nested loop, and an if modifier.

```
[
  {
    "id": "main_phase",
    "description": "Start browser and run the daily FYP interaction loop",
    "steps": [
      { "id": "open_browser", "tool": "open_browser" },
      {
        "id": "watch_loop",
        "tool": "for_time",
        "args": { "max_time": "experiment.variables.time" },
        "loop": [
          { "id": "predict_interaction", "tool": "predict_short_interaction" }
        ]
      }
    ]
  }
]
```

In practice, scenarios follow a recognisable three-part pattern across the platforms currently supported:

- Login phase. Open the browser. Optionally start network tracking. Dismiss cookie banners and similar interruptions. Go to the platform, log in with the user-specific credentials, and tear down the session.
- Seed phase. Initialise the session and iterate through topic-related search queries. Evaluate each returned item with the interaction predictor. Watch, like, or bookmark relevant items, and skip the irrelevant ones. Increment counters until the topic-specific seeding thresholds are reached (for example, 25 polarising likes and 25 neutral likes for the UMAP study), then tear down. This is the most complex scenario type and the one most closely coupled to topic queries and predictor configuration.
- Main phase. Open the browser, log in, and open the platform's main recommendation surface (For You Page, Shorts, or watch feed). Then run a time-bounded interaction loop. The loop calls the predictor on every candidate item and reacts according to the user's topic and stance. Tear down the session at the end.

User DAGs and Phases at runtime

Once scenarios exist, every enrolled sock puppet has a User DAG for each platform in the study. The DAG is the per-user, per-platform binding to the study's scenario chain. As the chain executes day by day, each scenario produces a Phase record for that DAG. The Control Panel's DAG Progress page is the operator-facing view of this state. For each User DAG it shows:

- the assigned user, platform, and study;
- a progress bar computed from completed phases over total planned scenarios;
- one phase tag per realised phase, sorted by scenario order. Each tag shows the phase name and successful_runs / required_runs;
- a status badge (Active while the DAG still has uncompleted scenarios, Completed when the last scenario has succeeded).

The dashboard deliberately exposes multiple levels of progress. The progress bar advances only when a phase is fully completed, which is appropriate for a high-level overview. The run counter inside each phase tag advances every day. That is what an operator needs to confirm that a multi-day main phase is actually moving forward. Filters by study, platform, and DAG status let the operator narrow large populations down. For example, all 68 UMAP sock puppets can be narrowed to a single cohort or to the individual whose progress is in question.

Agent Scheduling

Rather than employing conventional automation frameworks (Cypress, Puppeteer, Selenium, NoDriver), we developed a custom agent framework using Chrome DevTools Protocol (CDP) over WebSocket communication. This design decision provides several advantages: reduced automation footprint detection, faster adaptation to platform changes, direct access to media events (video playback state, buffering), and real-time network traffic monitoring. The implementation avoids common automation signatures such as missing audio services or simulated display contexts that trigger platform detection mechanisms. In practice, our Python execution pipeline takes the DSL (Domain-Specific Language) configuration

definition and runs functional tools that dictate which actions should be executed in the browser during a session.

Over time, we need to react to recurring events such as CAPTCHAs and network traffic anomalies. Therefore, we continuously monitor traffic, take screenshots of the current interaction state in the browser, and occasionally scrape parts of the HTML to ensure no events are blocking the view. For TikTok CAPTCHA resolution, we utilize a third-party service capable of resolving CAPTCHAs on specific platforms.

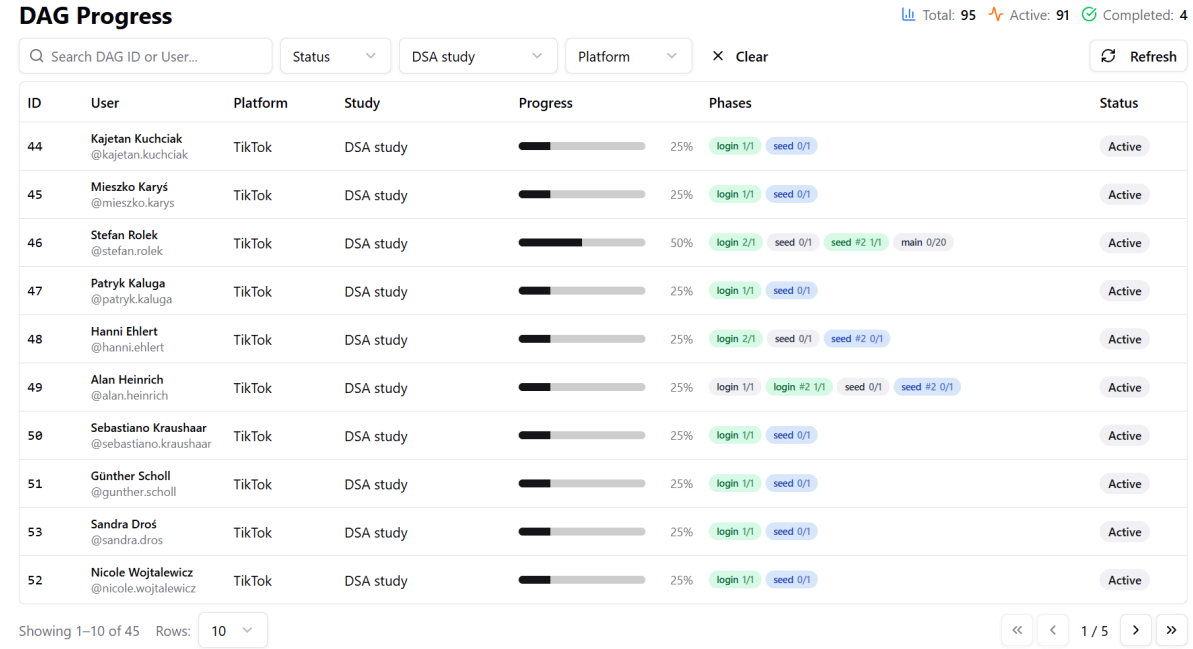
Each platform, such as YouTube or TikTok, can be represented by a similar, reproducible DSL. While the underlying implementation of these interactions may change over time, the DSL configuration remains constant to ensure the reproducibility of the experiments.

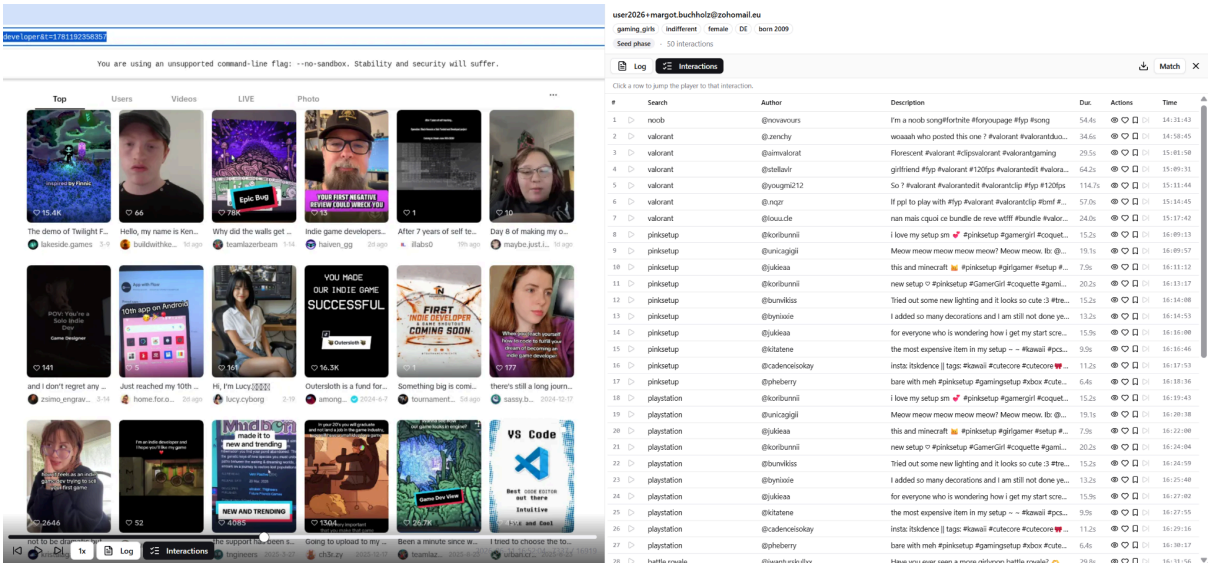
3.2 Monitoring Audit Execution

Each agent generates extensive logs of the browser session interactions. Collected data critical to the run is written to parquet table files, which are subsequently used during post-audit analysis. We also capture screenshots every second of the browsing session, allowing us to trace exactly what occurred on the screen frame by frame.

The parquet files contain data collected during the seed phase - such as search terms, video titles, descriptions, authors, and links. During the main phase, we collect the video topic, user information, and interactions performed during playback, alongside the video's title, description, author, and URL. Additionally, we include video transcripts if the predictor utilized this information during the session to decide whether to watch, like, or skip a video.

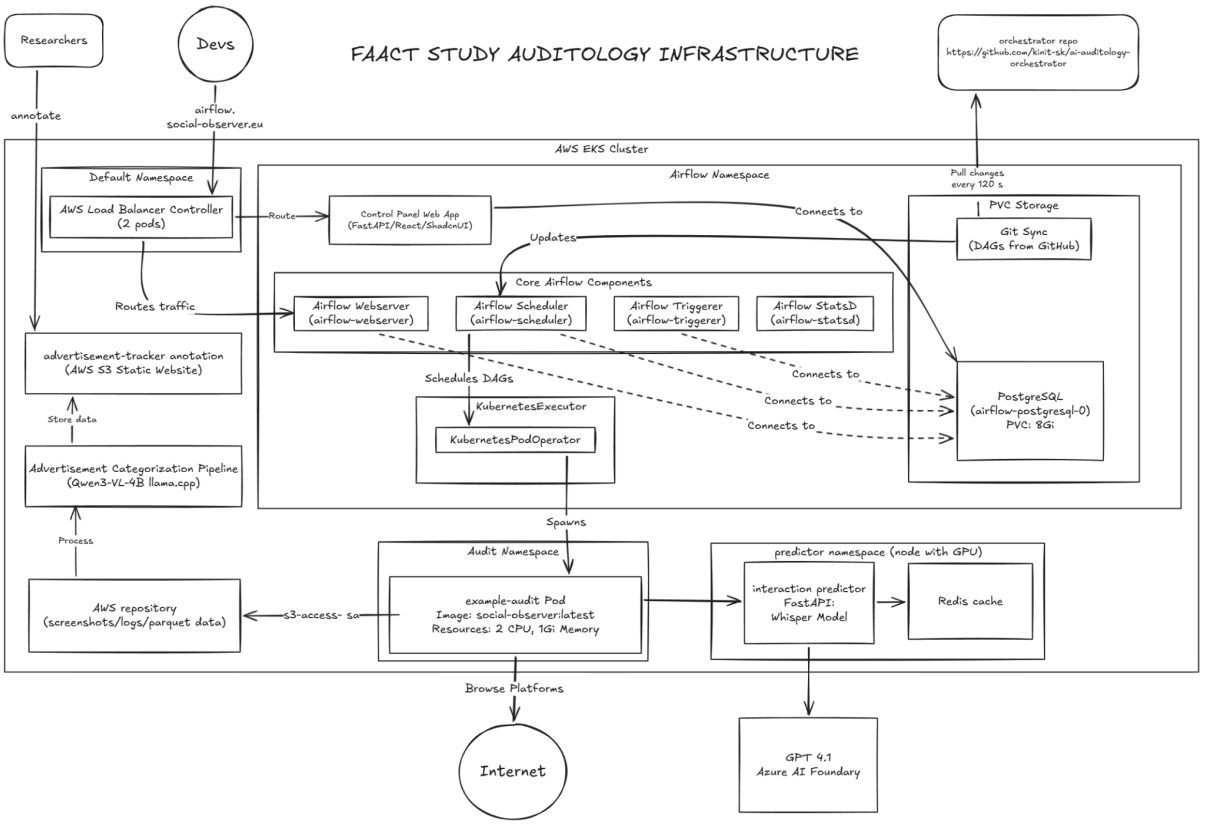
This data is initially stored on the local filesystem and transferred to an Amazon S3 repository at the end of each session. The files are later post-processed by our video categorization pipeline or can be audited via a screenshot replay feature in the control panel. Finally, critical Kubernetes infrastructure metrics and logs are aggregated in Grafana Loki and visualized via Grafana dashboards.





3.3 Architecture and Technological Implementation

The platform runs inside a single AWS Elastic Kubernetes Service (EKS) cluster. A small number of external systems sit outside the cluster for storage, downstream processing, and large-model inference. Figure 1. shows the production topology.



The Control Panel Web Application sits at the centre. It is a single full-stack service.
The backend is FastAPI. The frontend is React with the Shadcn UI component set. The Control Panel is the only researcher facing components. Every step of audit design happens through its web interface. That includes defining studies, topics, populations, and scenarios. It also covers generating and enrolling sock-puppet users, monitoring DAG progress, and reviewing collected screenshots and logs.

Control Panel provides data to Apache Airflow in the same Kubernetes namespace. Airflow is the current orchestration layer of the platform. This platform contains a forward facing webserver, scheduler and triggerer to allow launching of our browser agents.

When the scheduler executes a phase, it does not run the work inside the Airflow worker. Instead, it uses the KubernetesPodOperator to spawn a dedicated audit pod. The pod uses a container image. Each audit pod gets 2 CPU cores and 8 GB of memory, and handles exactly one audit run. The pod opens a controlled browser session and goes to the target platform through a country specific proxy. It then executes the scenario actions defined by the researcher, records screenshots and network traffic, and writes the resulting artefacts back to storage. Each run is isolated in its own pod. Runs of different sock puppets cannot interfere with one another, and individual failures do not propagate across the population.

A third namespace, scheduled on a GPU enabled node, hosts the interaction predictor. The predictor is a FastAPI service. It wraps a multimodal pipeline that includes the Whisper speech recognition model. During seed and main phases, the audit pod calls the predictor for every candidate piece of content. The call asks whether the simulated user should watch, like, bookmark, or skip the item, given that user's assigned topic and stance. A Redis cache

sits next to the predictor and stores repeated predictions, so the GPU is not invoked unnecessarily. The predictor itself further delegates to Azure AI Foundry (GPT-4.1) for text completion. This is the only point where the platform leaves the cluster for inference. All other audit logic runs internally.

We use a single shared PostgreSQL instance (airflow-postgresql-0) to store application data. It is backed by an 8 Gi persistent volume in the Airflow namespace. The data covers studies, topics, users, proxies, scenarios, populations, user DAGs, phases, runs, and their states. The Airflow Webserver, Scheduler, and Triggerer all connect to this database, and so does the Control Panel. One shared metadata store is what lets the Control Panel give a single coherent view across study design and Airflow execution. A researcher sees the same RunORM row, in the same state, that the scheduler is updating.

Three storage and processing systems sit outside the kubernetes cluster.

- Each audit pod uploads its run artefacts to an S3 storage. The artefacts include screenshots, agent and driver logs, network captures, and Parquet exports of recorded interactions.

- Advertisement Categorization Pipeline consumes the raw artefacts from S3 storage. The pipeline applies categorisation logic, for example distinguishing formal, disclosed, undisclosed, and non-ads as used in the study. Its outputs go to a downstream AWS repository that holds the processed screenshots, logs, and Parquet datasets used for offline analysis.

- We offer a single page app over S3 repository that displays the categorization results.

4. Conclusion

The implementation of the model-based algorithmic auditing requires a research of novel (AI-based) methods for: 1) representation and extraction of social media model (previously described in deliverables D2.1 and D2.2) , 2) audit scenario creation (previously described in deliverable D3.1); and within this deliverable, we follow-up with summarisation of 3) next user interaction prediction, and 4) (real-time) content annotation.

Moreover, an underlying technical (software and hardware) infrastructure is required to streamline the audit execution process (i.e., to automatically translate abstract audit scenarios to executable scripts, to execute bot behavior, and to automatically annotate the social media platform content - either during the audit execution itself or in post-audit phase to answer the posed audit question).

As the previous state-of-the-art research did not provide such methods and necessary underlying infrastructure, within the AI-Auditology project we researched, implemented and evaluated the first prototypes. In this way, we demonstrate the potential future directions that can be further undertaken and explored in the field of (model-based) algorithmic auditing.

Within our research and development activities, we aimed for flexibility and extendability of such methods and the auditing platform to support a wide range of future auditing assignments. To this end, the platform allows users to easily reuse existing or add new above-mentioned methods supporting or automating individual steps of the auditing process. To perform the sockpuppeting audits, the technical infrastructure implements user agents interacting with the audited platforms. Since platforms regularly change their interfaces, which can break functioning of bots (e.g., the platform-specific implementation for clicking on a like/upvote button may no longer work), we implemented appropriate monitoring measures to spot and fix these changes in a timely manner.

We also aimed to support high interpretability and transparency of the audit process. All quantitative results (e.g., differences between control and experimental groups of bots) were verified for statistical significance by statistical tests and calculation of effect sizes, and accompanied with corresponding confidence intervals. In this way, we aimed to avoid misinterpretation of the audit results as well as transparently report the accuracy of the underlying techniques. Even if the partial methods may introduce a small inevitable level of noise (e.g., inaccuracies in content annotation), the large-enough scale nature of the audits (e.g., running multiple bots with the same assigned characteristics) suppressed a possible influence of such random noise.