BEHAVIORAL SIGNALS

WHITE PAPER · BEHAVIORAL SIGNALS

The Behavioral Signals API

Behavioral, emotional and speaker signals from human voice, per utterance and with probabilities

Prepared by	Behavioral Signals
Date	Monday, 21 September 2026
Contact	Theodoros Giannakopoulos, Director of Machine Learning thodoris@behavioralsignals.com
Contents	Core outputs and models, integration and support, limitations and intended use, access and documentation
Access	behavioralsignals.com/portal

WHITE PAPER

Contents

1 Core Outputs and Underlying Models 3

1. Executive Summary ... 3
2. Output Description ... 3
3. Continuous Data and Annotation ... 4
4. Model Architecture and Performance ... 4
5. Robustness — RobuSER ... 5
6. Higher-Level Applications ... 5

Sources ... 5

2 Integration and Support 6

1. Speed ... 6
2. Support ... 6
3. Upgrades and Backwards Compatibility ... 6

3 Limitations and Intended Use 7

1. What the Outputs Are ... 7
2. What Was Evaluated, and How ... 7
3. Known Limits ... 7
4. What Is Built Above the API, Not In It ... 8

4 About This Paper, Access and Documentation 9

1. Scope and Audience ... 9
2. Get Access ... 9
3. Developer Documentation ... 9

TECHNICAL

Core Outputs and Underlying Models

Behavioral Signals · 21 September 2026

In brief. The Behavioral Signals API extracts behavioral, emotional and speaker signals from human voice, per utterance and with class probabilities. On our internal, multi-domain evaluation it reaches 77.9 % Macro-F1 on emotion, 74.6 % on valence and 74.3 % on arousal, varying by approximately ±10 % with the domain of the input. This paper summarizes those outputs, the models behind them, their measured performance and known limits, and how integrators access and support the service.

1. Executive Summary

The Behavioral Signals API takes audio recordings as input and returns a rich set of behavioral, emotional and speaker predictions per utterance. Every prediction is accompanied by class posterior probabilities, so integrators can apply their own thresholds, uncertainty measures or temporal smoothing on top of the labels. The models operate directly on raw audio, run in production today as a cloud API, on customer-hosted deployments, and in edge implementations, in both batch and real-time streaming modes, and have been deployed across applications from national-security and forensic-audio contexts to call-center analytics, customer-experience monitoring and marketing research.

2. Output Description

The following predictions are available through the production API. **Batch processing returns every output listed below. The real-time streaming path covers the paralinguistic outputs and does not include transcription or speaker diarization.** Measured speed for both modes is given under Integration and Support.

Category	Output	Classes
Structural	Speaker Diarization	Speaker IDs + utterance timestamps (start / end)
	ASR	Text transcription of each utterance
	Language	Detected language + winning probability
Speaker-level	Age	18–22 · 23–30 · 31–45 · 46–65
	Gender	Male · Female
Emotional	Emotion	Happy · Angry · Neutral · Sad
	Strength (arousal)	Weak · Neutral · Strong
	Positivity (valence)	Negative · Neutral · Positive
Behavioral	Speaking rate	Very slow · Slow · Normal · Fast · Very fast
	Hesitation	No · Yes (probability of "yes" is a continuous hesitation score)
Authenticity	Voice Deepfake Detection	Bonafide · Spoofed (speaker-agnostic)

All outputs are probabilistic. Every prediction is returned as a soft distribution over its classes — each class carries a posterior probability at inference time, not just a hard label. Integrators can use these probabilities

directly for temporal smoothing, uncertainty-aware routing, or to derive meta-metrics such as a **continuous –1 to +1 score** for arousal and valence via a weighted combination of the class posteriors.

3. Continuous Data and Annotation

To train and evaluate the core models, we have compiled a proprietary speech corpus of **over 1,000 hours of audio**, containing **more than 500,000 annotated utterances** that span discrete emotions, valence, arousal, deepfake and cloner detection, and secondary behaviors such as hesitation, engagement, speaking rate and speaker attributes. Data is drawn from diverse domains — interviews, phone calls, television and podcasts — to ensure broad contextual coverage.

Annotation is performed by a **dedicated internal team of expert annotators — not crowd-sourced** — following an iterative, multi-round protocol. Each utterance is reviewed by a minimum of five annotators, with up to nine independent annotations on selected examples. A statistical and manual QA pipeline resolves inter-annotator disagreements, with senior annotators and members of the machine-learning team performing hands-on review. This in-house, expert-led approach is a deliberate choice: it produces the label quality and consistency our downstream models depend on, in a way crowd-sourced pipelines routinely fail to. Class imbalance is addressed through deep-learning-based augmentation techniques [1]. Public benchmarks complement the proprietary set for external evaluation.

4. Model Architecture and Performance

The current generation of models is based on **large-scale self-supervised Transformer architectures operating directly on raw audio**, replacing an earlier convolutional stack that used MFCC / spectrogram features. This shift materially improves accuracy by capturing longer-range contextual structure in speech.

On our internal, multi-domain evaluation, the models achieve approximately **75 % overall classification accuracy** across the core emotional outputs (emotion, valence, arousal). Per-task Macro-F1 figures, measured on that same internal evaluation set, are:

Task	Average Macro-F1
Emotion	77.9 %
Valence (positivity)	74.6 %
Arousal (strength)	74.3 %
Speaking rate	77.0 %
Hesitation	79.9 %
Language	98.5 %
Gender	94.8 %
Age	95.2 %
Deepfake detection (speaker-agnostic)	97.9 %

Speaker diarization achieves a **17.8 % Diarization Error Rate**, with most errors localized at speaker-turn boundaries. Performance can vary approximately ±10 % depending on the domain of the input (calls, interviews, movies, and so on).

5. Robustness — RobuSER

Model robustness against noise is continuously monitored through **RobuSER**, an internal robustness benchmarking framework that we have released publicly on GitHub as a community benchmark for Speech Emotion Recognition [2]. It measures performance across varied acoustic environments, background-noise conditions and signal corruptions, and tracks it from release to release.

Beyond public benchmarking, the models have been benchmarked and iteratively enhanced against **extreme noise regimes, degraded and non-cooperative channels, heavy compression, and adversarial signal corruptions** — conditions materially more demanding than most commercial deployments encounter. That work feeds back into the general-purpose API: the robustness a mainstream customer experiences today has been stress-tested under those conditions.

6. Higher-Level Applications

The same core models power several downstream applications built on top of the API:

- **Cognitive impairment detection from speech**, including cross-lingual Alzheimer's dementia detection [4] and a 4th-of-80 placement in the regression task of the PROCESS Challenge at ICASSP 2025 [5].
- **Depression detection in the wild**, built directly on the core emotional outputs (emotion, valence, arousal, hesitation and speaking-rate posteriors) fed into a downstream classifier. Trained and evaluated on an internal in-the-wild corpus compiled from social-media clips and vlogger content, this application shows that the API's per-utterance behavioral signals transfer to a challenging real-world mental-health screening task without any change to the underlying models.
- **Emotional-content analysis of movies**, using the core emotional outputs to characterize scene and film-level affect [6].
- **Popularity prediction of oral presentations from speaking style**, recognized with the Runners-Up Best Paper Award at ACII 2024 [7].

Related work on filled-pause detection combining audio and ASR signal supports the hesitation output described above [3].

Sources

1. Chatziagapi, A. et al. (2019). Data Augmentation Using GANs for Speech Emotion Recognition. *Interspeech*, pp. 171–175.
2. Petrogianni, A. et al. (2024). RobuSER: A Robustness Benchmark for Speech Emotion Recognition. *ACII 2024*, IEEE.
3. Chatziagapi, A. et al. (2022). Audio and ASR-based filled pause detection. *ACII 2022*, IEEE.
4. Melistas, T. et al. (2023). Cross-Lingual Features for Alzheimer's Dementia Detection from Speech. *Interspeech*.
5. Placed 4th out of 80 submissions in the regression task of the PROCESS Challenge at ICASSP 2025 (Prediction and Recognition of Cognitive Decline through Spontaneous Speech). Leaderboard: processchallenge.github.io/finalresults
6. Giannakopoulos, T. et al. (2019). Using Oliver API for emotion-aware movie content characterization. *CBMI 2019*, IEEE.
7. Sgouropoulos, D. et al. (2024). Emotion-Aware Speech Popularity Prediction: A Use-Case on Ted Talks. *ACII 2024*, IEEE — Runners-Up Best Paper Award.

TECHNICAL

Integration and Support

Behavioral Signals · 21 September 2026

1. Speed

The production API delivers behavioral, emotional and speaker outputs at approximately **20 × real-time** on a mainstream cloud deployment — a one-hour audio file is typically processed in around three minutes end-to-end, including upload and polling overhead. Individual endpoint measurements on modern GPU hardware exceed this figure; the twenty-times number is a defensible aggregate that customers can rely on for capacity planning.

Measured batch and streaming performance on representative hardware:

Hardware	Batch real-time ratio	Streaming p95 latency
NVIDIA A100 SXM4 (production cloud)	29.0 ×	105 – 150 ms
NVIDIA A10 (mid-range cloud GPU)	20.0 ×	105 – 150 ms
NVIDIA RTX 3070 (consumer laptop GPU)	10.0 ×	105 – 150 ms
Intel Core i7 (laptop CPU-only)	1.2 ×	≈ 320 ms

Streaming figures are for the paralinguistic outputs, which is what the streaming path runs; transcription and diarization are batch. Figures are normalized for network and gateway overhead so that GPU classes can be compared directly. The consumer laptop-GPU row is the number to rely on when scoping an edge or on-premises appliance — the same production stack runs comfortably on hardware a customer already has.

2. Support

Support is anchored on a dedicated **Discord channel** for API integrators — join at https://discord.gg/fxjRrbMH3Q. The channel is monitored during business hours with a 24-hour target response, and is the primary surface for questions, incident reports and change announcements. For teams with more demanding operational requirements, an **optional on-demand dedicated support service** is available — direct engineering contact, prioritized response and named ownership on the customer side. Terms of that service are agreed per engagement.

3. Upgrades and Backwards Compatibility

The API is under active development. **Minor updates are transparent to users** — no client change is required and no interface shifts. **Major updates are announced in advance** on the Discord channel, with migration notes and a timeline for the transition.

Backwards compatibility is **one major version, starting from v5**. Clients running on v5 continue to work when v6 ships, and are given adequate lead time to migrate before v5 is retired. Any client that pins a specific major version is protected from surprise interface changes within the compatibility window.

TECHNICAL

Limitations and Intended Use

Behavioral Signals · 21 September 2026

1. What the Outputs Are

The API returns a probabilistic estimate of an expressed signal, per utterance. It is not a determination, not an inner state, not a fact about a person, and not a decision. An output describes what is expressed in the speech signal — the emotional tone of an utterance, whether a speaker expresses engagement in their tone, signs of hesitation — and an integrator who reads it as a measurement of what a speaker feels, intends or believes is reading it wrongly.

Every prediction is returned as a distribution over its classes, so the integrator sets the operating threshold rather than inheriting ours.

Voice deepfake detection returns a detection result — Bonafide or Spoofed, speaker-agnostic. It is not a verdict about a person's honesty, and it is independent of the emotional and behavioral outputs; the two are not merged into a single confidence.

2. What Was Evaluated, and How

Every figure in this paper was measured on our **internal, multi-domain evaluation set**, drawn from the proprietary corpus described in §3 — over 1,000 hours of audio and more than 500,000 annotated utterances, spanning interviews, phone calls, television and podcasts.

The metric is Average Macro-F1 per task, except speaker diarization, which is reported as Diarization Error Rate. Macro-F1 is not accuracy, and the two produce different numbers for the same capability.

Performance varies by approximately ±10 % with the domain of the input — calls, interviews, movies and so on. A figure quoted without that range is quoted incompletely.

3. Known Limits

- **Domain sensitivity.** The ±10 % range above is the expected variation, not a worst case. A deployment whose audio differs materially from the evaluated domains should be validated on its own data before launch.
- **Speaker diarization.** 17.8 % Diarization Error Rate, with errors concentrated at speaker-turn boundaries. Overlapping speech is where the error is.
- **Age.** Four bands spanning 18 to 65. **Speakers outside that range are not represented in the output space**, and the model does not signal that a speaker falls outside it.
- **Gender.** Two classes. Speakers who fall outside that binary are not represented.
- **Emotion taxonomy.** Four classes — happy, angry, neutral, sad. A product needing a different or finer taxonomy builds it as a layer above the API, from the class posteriors, rather than expecting it from the API directly.
- **Utterance length.** **The behavioral outputs do not share one minimum.** Arousal holds down to about a second of voiced speech; the others require longer. A single global floor quoted across all outputs would be inaccurate.

4. What Is Built Above the API, Not In It

Intent, churn risk, fraud, customer satisfaction and escalation are not API outputs. They are what an integration builds on top of the outputs, and the distinction is load-bearing: the test is whether a reader can find the value in the results payload. The higher-level applications described in §6 are themselves built this way — downstream models over the per-utterance signals, not additional endpoints.

Features the API marks experimental are documented as experimental, with their stated limits, in the developer reference.

WHITE PAPER

About This Paper

Behavioral Signals · 21 September 2026

1. Scope and Audience

This paper is a technical summary of what the Behavioral Signals API produces, how the underlying models were trained and measured, and how integrators access and support the service.

The audience is the developer or product owner evaluating the API for a real deployment. Everything described here is available today through the production API; nothing in this document is a roadmap item.

2. Get Access

The API is self-serve. Create an account at behavioralsignals.com/portal, verify your email, and issue a key from the portal. Both a REST interface and a Python SDK are provided. The SDK is on PyPI:

```
pip install behavioralsignals
```

A working key, a sample audio file and a first successful call are the intended first-hour experience. The REST interface is language-agnostic and covers every capability the SDK exposes.

3. Developer Documentation

Full endpoint reference, request and response schemas, code samples and error semantics are at behavioralsignals.readme.io. The reference includes a live console on every endpoint, so integrators can call the API from the documentation without leaving the page. The same site links the Python SDK, the changelog and the release notes.

BEHAVIORAL SIGNALS

Get started.

Sign up at behavioralsignals.com/portal and be running your first call within minutes. Developer documentation, code samples and live-console access are at behavioralsignals.readme.io; support is on our Discord channel.

CONTACT

Theodoros Giannakopoulos
Director of Machine Learning
thodoris@behavioralsignals.com

Behavioral Signal Technologies, Inc.
4470 W Sunset Blvd, Suite 94260 · Los Angeles, CA 90027, US
behavioralsignals.com