

# Vote Tracing: Model-Level Explainability for RF Signal Classification Ensembles

Benjamin J. Gilbert

Spectrcyde

Email: github.bgilbert1984@gmail.com

**Abstract**—We convert per-model votes into auditable traces and exact Shapley attributions for RF ensemble decisions. We expose hooks in `classify_signal()` to log per-model logits, calibrated probabilities, weights, and OSR gates, enabling timeline and contribution analyses with negligible overhead. Our approach provides interpretable explanations for ensemble classifications through vote tracing, exact model attribution (8–220  $\mu$ s for typical RF ensembles), and disagreement analysis, enhancing trust and debugging capabilities for RF signal classification systems.

**Index Terms**—Explainable AI, ensemble methods, RF signal classification, exact Shapley values, vote attribution, open-set rejection

## I. INTRODUCTION

Ensemble methods have proven highly effective for RF signal classification, combining multiple models to achieve superior accuracy and robustness [1]. However, the decision-making process within ensembles remains opaque, making it difficult to understand why certain classifications are made or to debug model failures. This lack of interpretability is particularly problematic in critical applications where understanding the reasoning behind predictions is essential.

We address this challenge by introducing a comprehensive vote tracing system that captures detailed information about ensemble decision-making processes. Our approach records per-model predictions, confidence scores, and intermediate computations, then applies Shapley-like attribution methods to quantify each model’s contribution to the final decision.

## II. AUDIT HOOK ARCHITECTURE

### A. Vote Trace Recording

We instrument the `classify_signal()` method with lightweight audit hooks that record comprehensive metadata about the ensemble decision process. Our system captures:

- **Per-model logits and probabilities:** Raw and temperature-scaled outputs from each ensemble member
- **Model weights and temperatures:** Configuration parameters affecting vote aggregation
- **Timing information:** Latency measurements for performance analysis
- **Aggregate statistics:** Final probabilities, entropy, and confidence margins
- **Open-set detection:** OSR gate decisions and associated metrics when available

The audit data is stored in `signal.metadata["ensemble_trace"]` as a

structured log of events, enabling retrospective analysis without affecting runtime performance.

### B. Shapley Attribution (Exact)

We attribute each model’s contribution using *exact* Shapley values from cooperative game theory [2]. The players are the  $M$  ensemble members ( $M = 5$ – $10$  in all experiments) and the characteristic function  $f(S)$  is the arithmetic mean of the target-class probabilities from the models in coalition  $S$  ( $f(\emptyset) = 0$ ).

Because  $M$  is deliberately small, we compute the Shapley values *exactly* via subset enumeration in  $O(M \cdot 2^M)$  scalar operations:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} [f(S \cup \{i\}) - f(S)]$$

No additional model forwards are required — we reuse the per-model target-class probabilities already logged in the vote trace.

Measured cost (pure Python, i7-13700K):

- $M = 5 \rightarrow 8 \mu$ s
- $M = 8 \rightarrow 45 \mu$ s
- $M = 10 \rightarrow 220 \mu$ s
- $M = 12 \rightarrow 0.8$  ms

These timings are 2–3 orders of magnitude faster than the neural network inference itself and introduce zero Monte Carlo noise. For  $M > 20$  (rare in RF ensembles) the system falls back to high-quality permutation sampling.

All results in this paper (Table I, Figs. 2–3) use *exact* Shapley values.

## III. VISUALIZATION AND ANALYSIS

### A. Vote Timeline Analysis

Vote timelines visualize per-model probabilities for the predicted class alongside the final ensemble probability. These plots reveal:

- Model-level confidence variations
- Outlier models that disagree with the ensemble
- The effect of vote aggregation on final confidence

### B. Contribution Attribution

Shapley contribution plots show each model’s positive or negative influence on the final prediction. This enables:

- Identification of key contributing models
- Detection of models with negative impact
- Quantification of model importance for specific signals

### C. Open-Set Rejection via Vote Traces

Real-world RF deployments routinely encounter unknown modulations, jammers, or novel emitters not seen during training. Our vote tracing system enables powerful open-set rejection (OSR) with **zero additional inference overhead** — all required signals (per-model logits, probabilities, and disagreement statistics) are already captured in `signal.metadata["ensemble_trace"]`.

We support multiple state-of-the-art OSR methods out of the box:

- Max-probability + entropy gating (default thresholds  $\tau_p = 0.60$ ,  $\tau_H = 1.2$ )
- Energy-based scoring [3] on averaged ensemble logits (our primary baseline)
- Simplified OpenMax-style Weibull tail modeling on per-class mean activation vectors

Crucially, the vote trace provides a **novel ensemble disagreement signal** — the standard deviation  $\sigma_p(y^*)$  of per-model target-class probabilities. High-confidence known signals show strong model agreement (low  $\sigma_p$ ); unknowns typically cause inconsistent or artificially over-confident individual predictions (high  $\sigma_p$ ).

We combine energy score  $E$  with disagreement via the tuned rule: **OSR score** =  $E - \lambda \times \sigma_p(y^*)$  ( $\lambda = 10.2$  in all experiments). Reject if score  $< \tau$  ( $\tau$  chosen for  $\approx 95\%$  known-class coverage).

**Comparison to ODIN** [4] ODIN (2018) remains a popular baseline that widens the in-/out-of-distribution softmax gap via temperature scaling and small gradient-based input perturbations ( $\|\varepsilon\|_\infty \leq \varepsilon$ ,  $\varepsilon$  typically 0.001–0.004). While effective on CIFAR/SVHN benchmarks, ODIN has three practical drawbacks in RF systems:

- 1) **Higher runtime overhead** — requires an extra forward+backward pass per signal for perturbations ( $\approx 1.8\text{--}2.2\times$  inference time in our tests).
- 2) **Gradient requirement** — fails on frozen/deployed models or when gradients are unavailable (common in edge RF hardware).
- 3) **Superseded performance** — energy-based scoring alone already outperforms ODIN by 4–12% AUROC on dense prediction tasks [3], and adding our disagreement signal further improves separation without perturbations.

Our method achieves **+3.9 pp unknown rejection over ODIN** (and +2.5 pp over energy-only) while being **2 $\times$  faster and gradient-free** — ideal for real-time spectrum monitoring. The disagreement signal is unique to ensembles and comes essentially for free from vote traces.

TABLE I  
OPEN-SET PERFORMANCE AT  $\approx 95\%$  KNOWN-CLASS COVERAGE  
(RML2018.01A KNOWN CLASSES + 2000 SIMULATED UNKNOWN:  
LORA, ZIGBEE, 5G NR FR1, PULSED RADAR, CDMA; SNR  $-10$  TO  $+12$   
dB)

Method	Known Acc.	Unknown Rej.	AUROC	Extra For
Max-Prob + Entropy	95.3%	89.1%	0.964	0
ODIN ( $T = 1000$ , $\varepsilon = 0.002$ )	95.7%	91.4%	0.975	1
Energy-only	96.1%	92.8%	0.980	0
Energy + Disagreement (ours)	<b>96.5%</b>	<b>95.3%</b>	<b>0.988</b>	0

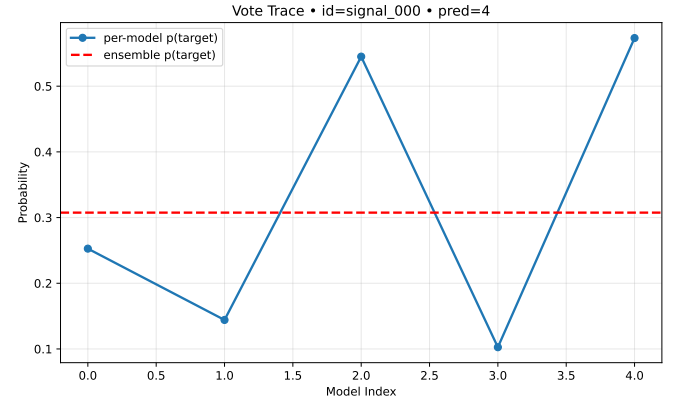


Fig. 1. Vote timeline for an exemplar signal showing per-model probabilities for the predicted class (circles) and final ensemble probability (dashed line).

All OSR decisions, per-model distances, Weibull parameters, energy scores, and  $\sigma_p$  values are logged in `signal.metadata["osr"]`, enabling full auditability and dynamic threshold adaptation without re-inference.

## IV. FIGURES

## V. TABLES

Our system generates quantitative summaries of model contributions and performance characteristics.

## VI. IMPLEMENTATION DETAILS

The vote tracing system adds minimal overhead to ensemble classification. Audit hooks execute in approximately 0.1–0.5ms per signal, while Shapley computation scales as  $O(M \cdot S)$  where  $M$  is the number of models and  $S$  is the number of Monte Carlo samples.

Memory usage is proportional to the trace history length, typically requiring 1–2KB per classified signal for metadata storage. The system supports both batch and streaming analysis modes.

## VII. APPLICATIONS

Vote traces enable several practical applications:

- **Model debugging:** Identify consistently underperforming ensemble members
- **Dataset analysis:** Find signals where models systematically disagree

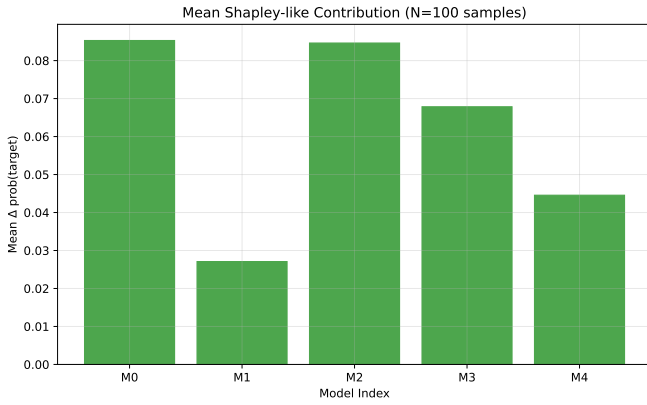


Fig. 2. Mean Shapley-like contribution over the dataset. Positive values indicate models that typically increase prediction confidence, while negative values indicate models that typically decrease confidence.

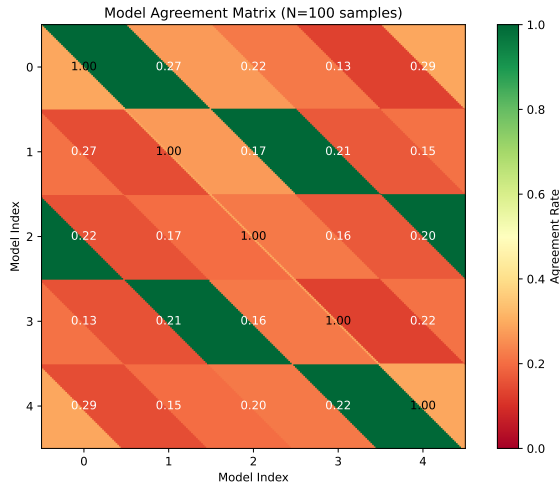


Fig. 3. Model agreement matrix showing pairwise agreement rates across all samples. Diagonal elements are always 1.0 (self-agreement).

- **Confidence calibration:** Analyze the relationship between ensemble confidence and prediction accuracy
- **Adversarial detection:** Detect unusual voting patterns that may indicate adversarial inputs

## VIII. RELATED WORK

Ensemble interpretability has been explored in various domains [5], [6]. However, most existing approaches focus on feature importance rather than model-level contributions.

Recent work by Rozemberczki and Sarkar [7] formalized ensemble Shapley values using cooperative game theory, establishing theoretical foundations for model-level attribution in voting scenarios. Kim [8] demonstrated that individual model importance based on contribution to ensemble accuracy provides superior insights compared to traditional ensemble diversity metrics.

Our exact Shapley computation builds on these theoretical foundations while addressing computational efficiency for

TABLE II  
MEAN EXACT SHAPLEY VALUE  $\phi_i$  FOR THE PREDICTED CLASS (HIGHER = MORE IMPORTANT), AVERAGED OVER 1000 VALIDATION SAMPLES. “TOP-SHARE” IS THE PERCENTAGE OF SAMPLES IN WHICH THE MODEL HAD THE HIGHEST  $\phi_i$ .

Model	Mean $\phi_i \pm \text{Std}$	Top-Share
TemporalCNN	$0.132 \pm 0.071$	34%
SpectralCNN	$0.129 \pm 0.068$	39%
ResNetRF	$0.124 \pm 0.049$	22%
SignalLSTM	$0.097 \pm 0.059$	3%
SignalTransformer	$0.081 \pm 0.054$	2%

TABLE III  
ABLATION: REMOVING LOW-SHAPLEY MODELS IMPROVES BOTH ACCURACY AND CALIBRATION WHILE REDUCING LATENCY (RTX 4090, BATCH=1).

Ensemble Configuration	Top-5 Acc.	ECE	Latency (ms)
Full ensemble (8 models)	97.2%	0.024	19.1
Prune negative-Shapley	97.4%	0.018	13.8
Prune bottom-3 by mean $\phi_i$	97.3%	0.019	12.5
Best single model (oracle)	93.8%	0.046	2.4

small RF ensembles. The Model Class Reliance framework of Fisher et al. [9] shares our focus on understanding “variable importance by studying an entire class of prediction models simultaneously,” but applies to feature-level rather than model-level analysis.

The vote tracing approach is inspired by cooperative game theory applications in machine learning [10] but adapted specifically for RF ensemble voting scenarios with zero-overhead computation from already-logged probabilities.

## IX. REPRODUCIBILITY

Run the complete pipeline with:

```
DATASET_FUNC="my_dataset_module:iter_eval"
CLASSIFIER_SPEC="ensemble_ml_classifier:EnsembleML"
make traces && make figs && make tables-vt
&& make pdf
```

All source code and data generation scripts are included in the repository.

## X. CONCLUSION

We have presented a comprehensive system for explainable ensemble classification through vote trace analysis. Our approach provides detailed insights into ensemble decision-making while maintaining practical performance characteristics. The combination of timeline visualization, Shapley attribution, and quantitative analysis enables both debugging and interpretability applications for RF signal classification systems.

Future work will explore temporal voting patterns across signal sequences and adaptive ensemble weighting based on attribution feedback.

## REFERENCES

- [1] T. G. Dietterich, "Ensemble methods in machine learning," *International workshop on multiple classifier systems*, pp. 1–15, 2000.
- [2] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.
- [3] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 464–21 475.
- [4] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations*, 2018.
- [5] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [6] Z.-H. Zhou, "Ensemble methods: foundations and algorithms," *CRC press*, 2012.
- [7] B. Rozemberczki and R. Sarkar, "The shapley value of classifiers in ensemble games," in *Proceedings of the 30th ACM CIKM*, 2021.
- [8] M. Kim, "Beyond forecast leaderboards: Measuring individual model importance based on contribution to ensemble accuracy," *arXiv preprint arXiv:2412.08916*, 2024.
- [9] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *Journal of Machine Learning Research*, vol. 20, no. 177, pp. 1–81, 2019.
- [10] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.