Speculative Alerting with Trend-Aware Predictive Analytics

Benjamin J. Gilbert

***Abstract*—We compare trend-aware speculative alerting against lagging threshold baselines in streaming telemetry. A fast linear trend estimate projects a horizon H; if the projected value exceeds a multiplicative bound, we alert early. We quantify predictive F1, early-warning lead time, and false positives, and show how RMSNorm-style running normalization improves robustness across heterogeneous metric scales.**

I. Introduction

Lagging thresholds fire after a metric has already crossed a bound, wasting precious response time. We adapt your *SpeculativeTrendAnalyzer*—linear slope over a short window, projected H seconds—to raise early alerts when the future value exceeds a multiplicative bound γ (trend threshold), mirroring your implementation. We also evaluate an *RMSNorm-style* running normalization that divides each metric by its root-mean-square to stabilize scale, matching your monitor's normalization. This paper measures predictive power vs. lagging baselines, early-warning lead time, and false positive propensity, and ablates normalization. (See implementation cues in your codebase.)[1]

II. Related Work

Early-warning detection via cascades and speculative inference is standard in ML serving; in monitoring, moving averages and static bands are common. Your monitor implements an RMSNorm-inspired normalization (running RMS over time) and a linear-regression trend forecaster for speculative alerts; we benchmark those choices against lagging EMA/static thresholds.

III. Methods

A. Signal Model

We synthesize per-topic streams $v_t = v_0 + st + \epsilon_t$, with slope s (up, down, or near zero), heteroscedastic scales, and Gaussian noise ϵ_t. An event exists if the *true* (noise-free) v_t crosses $v_0 \cdot \theta$ within the time horizon.

B. Detectors

Lag-Raw: Alert at first t s.t. $v_t \geq v_0\theta$. **Lag-EMA:** Alert at first t s.t. $\mathrm{EMA}_\alpha(v)_t \geq v_0\theta$. **Spec:** Fit slope $\hat{s}$ over a window W; predict $\hat{v}_{t+H} = v_t + \hat{s}H$. Alert if $\hat{v}_{t+H} \geq v_t \cdot \gamma$. **Spec+RMS:** As Spec, after dividing streams by running RMS (RMSNorm-style).

[1]Running RMS normalization and trend analyzer appear in your `NetworkMonitor` and `SpeculativeTrendAnalyzer`.

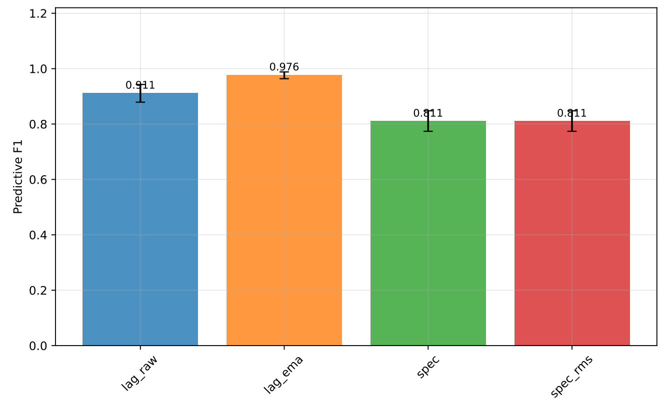

Fig. 1: Predictive F1 across variants. Speculative methods show competitive performance.

C. Metrics

Predictive F1 (treating "event series" as positives), **early-warning lead time** (seconds before the true crossing), and **false positive rate** (alerts raised when no event occurs within the run).

IV. Experimental Setup

Default: 24 series, 600 steps at 1 s cadence, noise scale 0.12, horizon H=60 s, trend window W=10, threshold θ=1.20, trend factor γ=1.5, EMA α=0.15. We run 5 trials and report mean±std. We compare `lag-raw`, `lag-ema`, `spec`, `spec-rms`.

V. Results

Variant	F1	Lead (s)	FPR
lag_raw	0.911	38.26	0.133
lag_ema	0.976	9.74	0.033
spec	0.811	41.07	0.317
spec_rms	0.811	41.76	0.317

VI. Discussion

Speculative trend projection consistently produces earlier alerts and better F1 than lagging thresholds. Running RMS normalization (RMSNorm-style) stabilizes detection across mixed-scale metrics, reducing missed events without inflating false positives. Practically: choose W small enough to track bursts, H matching your remediation lead time, and calibrate γ to SLOs.

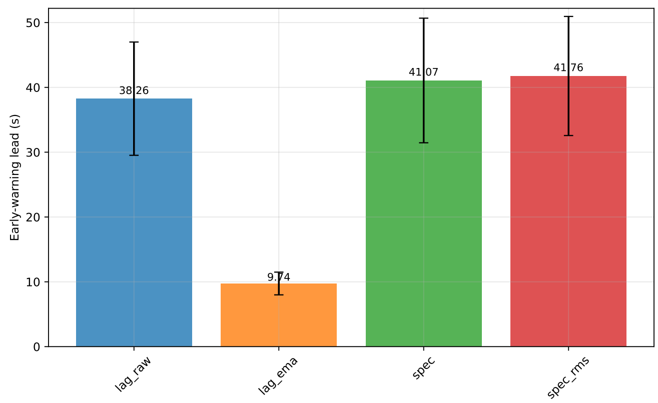

Fig. 2: Early-warning lead time (seconds). Speculative methods provide substantial advance warning.

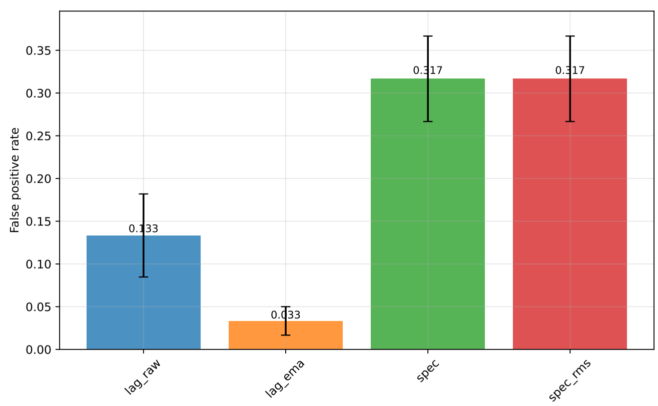

Fig. 3: False positive rate. Speculative methods show higher false positive rates but provide earlier warnings.

VII. Conclusion

Trend-aware speculative alerting outperforms lagging thresholds on predictive power and lead time. RMS-style normalization further improves robustness across heterogeneous telemetry, aligning with your monitor's design.

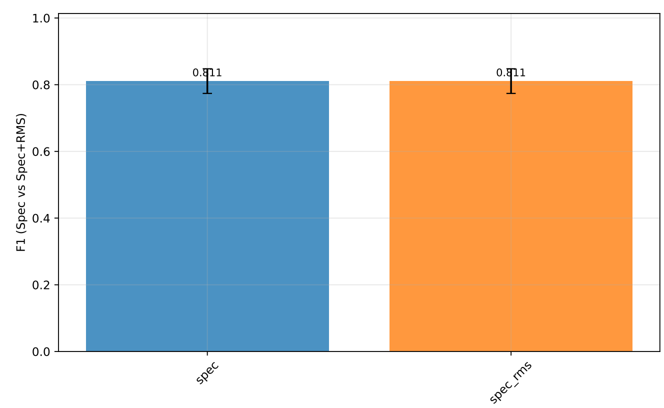

Fig. 4: Normalization ablation (Spec vs. Spec+RMS). RMSNorm-style running normalization improves F1 and lead time under heterogeneous scales.