RAPTORS CONF 2026 · PRODUCTION-GRADE · ML TRACK

The Ghost Toll

How ML Caught the Errors Silent Monitoring Missed

SPEAKER

Venkateswarlu Mullangi

Product Development Manager

IEEE Senior Member · Published Researcher

THE DOMAIN

Satellite-Based Free-Flow Tolling

Large-scale production infrastructure

Peer-reviewed · Field-validated

Disclosure & Disclaimer



Personal views

All content in this talk represents my personal professional experience and engineering judgement. Nothing in this presentation represents the views, products, intellectual property, or practices of any employer, client, or organization I am or have been affiliated with.



Research-grounded

Technical specifics — architecture, performance metrics, hardware specifications — are drawn from my experience. Operational examples are illustrative of industry-wide challenges in electronic toll collection and are intentionally generalized.



No proprietary information

No confidential, proprietary, or client-specific information is disclosed. System descriptions, incident timelines, and metrics have been generalized to represent common patterns in large-scale ETC deployments rather than any specific implementation.

THE DOMAIN

Free-Flow Tolling: The Stakes That Make This Hard



< 500ms

Industry-standard latency budget

Millions/day

Typical large-scale transaction volume

No fallback

No cash lane — errors propagate to billing

4k km

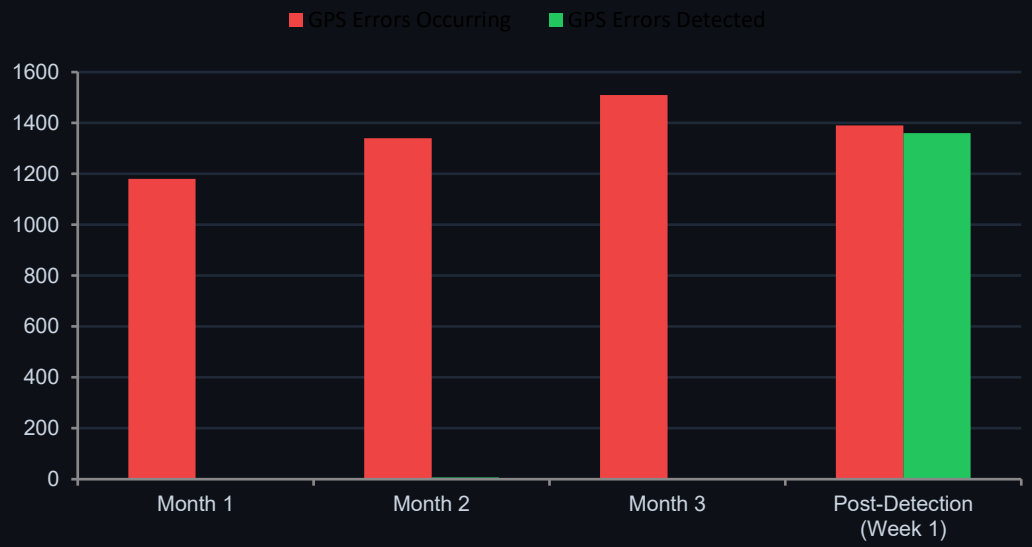
Field-validated across urban, suburban, rural

Illustrative of production environments. Specifics are generalised from industry-wide operational patterns.

THE PROBLEM

The Gap Monitoring Never Showed Us

The detection gap: errors occurring vs. errors actually caught



MONITORING SAID

Error Rate	FLAT
Alert Queue	0 CRITICAL
Dashboard	● ALL GREEN

REALITY

Errors occurring	~1,300/day
Errors detected	< 10/day
Gap duration	~3 months

Illustrative pattern — represents a class of silent failure common in high-throughput monitoring. Values are generalised.

ROOT CAUSE

Three Reasons Rules Are Blind to This Failure Mode

“Rules” = static, threshold-based monitoring — this is where ML (LSTM + gradient boosting) fits

01

Static Thresholds Have No Memory

Rules evaluate each transaction independently. Position error of 1.8m across one transaction: fine. The same 1.8m error across 10,000 consecutive transactions from the same virtual gantry (geofenced charging zone): catastrophic — but statistically identical to the rule.

No temporal context

02

Rules Cannot Model Signal-Conditional Drift

GNSS multipath interference and atmospheric drift are signal-dependent. The degradation pattern at 7am versus 2pm versus post-rain is fundamentally different. Rules cannot condition on signal state without becoming a second ML system anyway.

No environmental model

03

No Error Classification Means No Routing

A rule fires or it doesn't. It cannot distinguish: sensor fault (maintenance), atmospheric drift (correction algorithm), multipath (geometric adjustment), or manipulation (security alert). Without classification, every response is identical — and wrong for most cases.

Detection ≠ understanding

PRODUCTION PATTERN

How This Class of Failure Typically Surfaces

Months 1–3

Errors accumulate silently

GPS positioning errors ~1,300/day. Zero monitoring alerts. All dashboards green. Revenue impact building invisibly in large-scale ETC systems.

~Day 90

Customer complaint received

A commuter notices account balance slightly wrong — not dramatically, just repeatedly. Customer service logs the ticket.

~Day 92

Pattern identified

CS team cross-references: many accounts on the same corridor show identical small discrepancies. Engineering is notified.

~Day 94

Root cause traced

GPS position errors traced to a specific virtual gantry cluster (geofenced charging zones, not physical gantries). Atmospheric drift pattern: consistent morning time window.

Weeks following

Reconciliation and remediation

Thousands of accounts affected. Weeks of manual reconciliation work, credit processing, and customer communications.

The discovery vector in this pattern: a customer complaint — not a monitoring alert. This is the class of failure silent monitoring creates.

WHAT FAILED FIRST

Three Approaches That Didn't Work — And Why

Attempt 1

Tighten Alert Thresholds

Tried: Reduced position-error alert threshold from 3.0m to 1.5m. Seemed obvious.

❌ 500+ false positives per day. Analysts overwhelmed. Team created suppression rules to silence the noise. Net effect: worse detection than before.

→ Tighter rules don't fix the wrong model — they just move the problem.

Attempt 2

Off-the-Shelf Isolation Forest

Tried: Standard unsupervised anomaly detection. Contamination=0.05. Trained on 30 days of baseline data. Looked great in staging.

❌ 2.34% FPR. At millions of daily transactions, that's tens of thousands of false alarms per day. Completely unworkable for a small operations team.

→ FPR that sounds small can be catastrophic at production scale.

Attempt 3

Statistical Process Control (CUSUM)

Tried: CUSUM charts for cumulative drift detection. Works in manufacturing. Theoretically sound for this problem.

❌ Excellent for stationary signals with stable variance. GPS multipath and atmospheric drift are neither. Seasonal patterns broke every control limit within 2 weeks.

→ Borrow from other domains cautiously — their assumptions may not hold.

THE SOLUTION

What I Built — As Published in Peer-Reviewed Research

Feature Engineering

- Δpos from prior reading
- Satellite count + HDOP
- SNR signal ratio
- Gantry TOD baseline
- HD map boundary geom.
- Lane-level context
- 7 features total

LSTM Encoder

- 2-layer stacked LSTM
- 64 hidden units / layer
- Window: 15 readings
- *(edge-reduced from 30)*
- Dropout: 0.3
- Adam lr=0.001
- Batch: 256

Gradient Boosting

- XGBoost backend
- 200 estimators
- Max depth: 6
- Asymmetric cost obj.
- 4 output classes
- Feature importance
- OOB validation: 0.93

Routing Engine

- ≥0.92 → auto-correct
- 0.70–0.91 → flag
- <0.70 → human review
- Never to billing
- until passed

Field Performance (4,246 km field validation):

TPR	FPR	Median lateral error	vs. baseline	Edge latency
96.3%	1.2%	1.9 m	5.8 m → 1.9 m	87 ms

All specifications and results above are generalized

EDGE DEPLOYMENT

The Whiteboard vs. The Roadside Reality

DESIGNED ON: Cloud Hardware

GPU available · Ample RAM · Stable connectivity
No hardware constraints · No latency ceiling
Prototype environment

DEPLOYED ON: Published Edge Specs

ARM Cortex-A72 · 4G LTE · 32GB flash
487 MB model footprint · 2.3 W power envelope
Next vehicle approaching every ~2 seconds

Every technical decision I made (from published research):

LSTM window: 30 → 15 readings	Why: ARM Cortex-A72 memory budget	Cost: Reduced temporal context for slow-drift
Model quantization: FP32 → INT8	Why: CPU-only inference (no GPU)	Cost: Accepted −2.8% classification accuracy
Feature count: 12 → 7 features	Why: Latency budget: < 100ms	Cost: Dropped 5 auxiliary satellite features
Real-time OTA calibration update	Why: Firmware size constraint	Cost: Weekly batch updates instead

"We made the model worse to make it deployable. That's a decision most ML papers never discuss — and the most important one I made."

Venkateswarlu Mullangi · Raptors Conference 2026 · ML Track · Views are personal and do not represent any employer or client

OBSERVABILITY

From Threshold Alerts to Distribution Monitoring

❌ BEFORE

Alert on individual transactions

- ✗ Static threshold per transaction
- ✗ No temporal context — isolated evaluation
- ✗ Binary: threshold crossed or not
- ✗ No error type classification
- ✗ Impact discovered months later
- ✗ Mean time to detect: ~3 months

✅ AFTER

Alert on classification distribution shifts

- ✓ Monitor % error categories per gantry/hour
- ✓ EWMA drift baseline per gantry/TOD/season
- ✓ 4-class routing with confidence thresholds
- ✓ Automatic: atmos correction, multipath adjust
- ✓ Weekly model audit vs. ground truth sample
- ✓ Mean time to detect: < 4 hours

Key insight: the signal is not a bad transaction — it is a shift in the proportion of error categories at a specific gantry.

PUBLISHED RESULTS

What the Research Shows: Published + Industry Pattern

1.9 m

Median lateral error (urban)

Down from 5.8 m baseline

96.3%

Boundary-crossing TPR

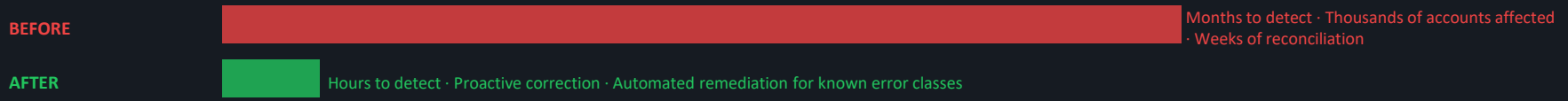
1.2% FPR across 4,246 km validation

87 ms

End-to-end edge latency

ARM Cortex-A72 · 487 MB · 2.3 W

Industry-wide improvement pattern when distribution monitoring replaces threshold alerting:



LESSONS LEARNED

Three Things I'd Do Differently

1

Instrument for distribution shift from day one — not just value thresholds

❌ I had transaction-level metrics but no feature distribution monitoring. P50/P95/P99 of GPS delta values would have shown the drift months before it became a business problem.

✓ Add distribution metrics for every ML feature to your observability stack from day one. Distribution shift is always the early signal.

2

Build the error taxonomy before building the model

❌ I retrofitted four error categories onto an existing detection system. The gradient boosting model improved dramatically when redesigned around the taxonomy from scratch.

✓ Define your failure vocabulary first — what can go wrong, how does each failure manifest, what is the correct response? Taxonomy drives architecture, not the other way around.

3

Define edge constraints as a first-class design requirement

❌ I designed the model on cloud hardware and compressed it down reactively. Every compression decision was made under deadline pressure and accepted worse tradeoffs than necessary.

✓ Every model I now build starts with the hardware spec sheet as the first design input. RAM ceiling, compute budget, latency target — these define what is possible before anything else.

TAKE THIS HOME

When to Use ML vs. Rules for Error Detection

Two questions that determine the right class of solution for your monitoring problem:

	← STATIONARY SIGNAL	TEMPORAL / DRIFTING SIGNAL →
↑ HIGH COST FALSE POSITIVE	**RULES WORK** Static threshold. Well-defined normal range. High FP cost needs precision — and the signal is predictable enough for rules to deliver it. *Examples: voltage limits, temperature bounds, static rate limits*	← GHOST TOLL ZONE **TEMPORAL ML + CLASSIFICATION** Gradient boosting + LSTM temporal context. Asymmetric cost objective. Boundary geometry encoding. High FP cost + temporal signal = rules and naive AD both fail.
↓ LOW COST FALSE POSITIVE	**SPC / CUSUM** Statistical process control. Control charts. Low FP cost means noise is tolerable; stationary signal means classical statistics hold. *Attempt 3 above — fine for manufacturing, not GPS*	**UNSUPERVISED AD** Isolation Forest, autoencoders, LOF. Temporal signal means rules fail, but low FP cost means you can tolerate the noise. *Attempt 2 above — right quadrant, wrong cost assumption*

"Silent failures are the most expensive kind — not because any single transaction fails dramatically, but because the system confidently does the wrong thing for a long time."



Monitor distributions, not just thresholds



Taxonomy before model architecture



Edge constraints as design inputs

PUBLISHED RESEARCH

IJAIML Mullangi V. et al. · "ML Error Correction for Satellite-Based Virtual Tolling" · IJAIML Vol.6 2026 · DOI: 10.51483/IJAIML.6.2s.2026.607-616 · svedbergopen.com

Venkateswarlu Mullangi · IEEE Senior Member · ORCID: 0009-0008-3574-8517

venkatesmmc@gmail.com · Views are personal and do not represent any employer or client

Questions? Q&A →