iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

MACHINE LEARNING–ENABLED EVENT-DRIVEN ARCHITECTURES FOR ADAPTIVE ELECTRIC VEHICLE ENERGY PLATFORMS

Ranga Raya Reddy Eragamreddy
Lead Software Engineer • Austin, Texas, United States

ABSTRACT
Event-driven architectures (EDA) and machine learning (ML) have individually transformed how software systems handle data at scale, yet their convergence for domain-specific platforms remains underexplored. This paper presents a novel architecture that unifies event sourcing, CQRS (Command Query Responsibility Segregation), and adaptive machine learning within a single platform purpose-built for electric vehicle energy optimization. Unlike traditional batch ML pipelines that operate on stale data, or stream processors that lack architectural memory, the proposed system treats every data point-from battery telemetry to grid pricing signals-as an immutable event that flows through a choreographed network of eight specialized ML models, each capable of self-adaptation through event-triggered online learning. Through a 14-month production deployment across 10,200 connected EVs in three U.S. regions, the platform demonstrates a 20.0% reduction in energy consumption per 100 miles compared to rule-based baselines (38.8 kWh vs. 48.5 kWh), range prediction accuracy of ±2.4 miles (83.1% improvement), charging cost reduction of 33.1%, and model drift recovery in 11.5 minutes versus 4.2 hours for static pipelines. The architecture processes 850,000 events per second at 168ms end-to-end p99 latency, with the event-sourced design enabling complete audit trails, temporal queries, and semantic failure recovery through saga-based compensation. Five fleet case studies across ride-share, delivery, corporate, transit, and consumer V2G segments validate the framework's cross-domain adaptability. The results establish that event-driven ML architectures are not merely an implementation choice but a paradigm shift that enables emergent optimization-where coordinated model interactions through events produce outcomes superior to any individual model operating in isolation.

Keywords:
Event-Driven Architecture, Machine Learning, Electric Vehicles, CQRS, Event Sourcing, Online Learning, Energy Optimization, Reinforcement Learning, Apache Kafka, Apache Flink, Digital Twin, Vehicle-to-Grid

CHAPTER 1. THE CONVERGENCE IMPERATIVE

> *"The best architectures emerge from the recognition that data is not a lake to be stored but a river to be navigated."*
>
> - Werner Vogels, CTO of Amazon

The electric vehicle revolution generates a paradox: the data that could optimize every aspect of EV energy usage-from battery chemistry to grid economics-is produced as a continuous stream of events, yet the machine learning systems designed to extract value from this data are overwhelmingly built as batch processes that operate on static snapshots. This architectural mismatch creates a fundamental latency between when the world changes and when the system adapts, a gap measured in hours or days that translates directly into wasted energy, suboptimal charging decisions, and inaccurate range predictions.

Consider a concrete scenario: a fleet vehicle departs its depot at 7:00 AM. By 7:15 AM, an unexpected cold front has dropped the ambient temperature by 12°C, the driver has enabled cabin heating (increasing energy draw by 3.2 kW), and the highway route requires sustained high-speed driving. A batch ML system, operating on yesterday's features, predicts a range of 245 miles. The actual range, given current conditions, is 198 miles. The driver, trusting the prediction, skips a charging stop and is stranded 18 miles from the destination. An event-driven system, processing temperature, HVAC, and speed events in real-time and triggering model re-inference within seconds, would have predicted 201 miles and recommended the charging stop.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

This paper argues that the solution is not merely faster ML pipelines but a fundamentally different architecture: one where machine learning models are first-class citizens within an event-driven system, consuming events as inputs, producing predictions as events, triggering retraining as events, and coordinating with other models through event choreography rather than centralized orchestration.

CHAPTER 2. THE EVENT TAXONOMY

> *"In an event-driven world, everything that happens is a fact, and facts are immutable."*
> - Greg Young, creator of Event Sourcing

The platform processes eight categories of events, each with distinct frequency, payload, priority, and ML consumer characteristics. Table 1 catalogues the complete event taxonomy.

Event Domain	Event Types	Frequency	Payload	Priority	ML Consumer
Vehicle Telemetry	BMS, motor, thermal, GPS, regen	10–50 Hz	0.5–3 KB	P0–P1	Range, Battery SoH, Digital Twin
Charging Events	Session start/stop, power rate, SoC	1 Hz (active)	1.2 KB	P0	Charge Optimizer, Grid Balancer
Grid Operator	LMP pricing, frequency, renewable mix	15–60 sec	0.8 KB	P0	V2G Coordinator, Demand Engine
Weather Service	Temp, precipitation, wind, irradiance	15 min	2.1 KB	P1	Range Estimator, HVAC Controller
User Behavior	App interactions, route preferences	Event-driven	0.4 KB	P2	Driver Assistant, Personalization
ML Model Events	Prediction, retrain trigger, drift alert	Continuous	1.5 KB	P1	All models (feedback loop)
OTA / Config	Firmware updates, parameter changes	Event-driven	5–50 KB	P1	Compliance, Feature Flags
Anomaly Alerts	Fault codes, safety events, degradation	Event-driven	3.2 KB	P0	Anomaly Detector, Fleet Ops

Table 1: Event Taxonomy - Categories, Characteristics, and ML Consumers

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Figure 1: Event Stream Composition by Category

Vehicle telemetry dominates at 62% of total volume, but the most architecturally significant events are ML Model Events (5%): when the Range Estimator detects that its prediction confidence has dropped below a threshold, it emits a RangeConfidenceLow event. The Charge Optimizer, subscribed to this event type, responds by widening its safety margin. The Grid Balancer adjusts its V2G discharge limits. This event-mediated coordination produces emergent optimization behavior that no centralized orchestrator could achieve.

CHAPTER 3. ARCHITECTURE: EVENTS MEET INTELLIGENCE

> *"The architecture should tell the reader about the system, not about the frameworks used in the system."*
> - Robert C. Martin

The platform architecture comprises four interacting domains: Event Sources that produce the raw data stream, an Event Mesh providing reliable ordered delivery, a CQRS + Event Sourcing Core maintaining system state as an immutable event log, and an Adaptive ML Layer that consumes events, produces predictions, and triggers self-improvement. Figure 2 illustrates the complete architecture.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

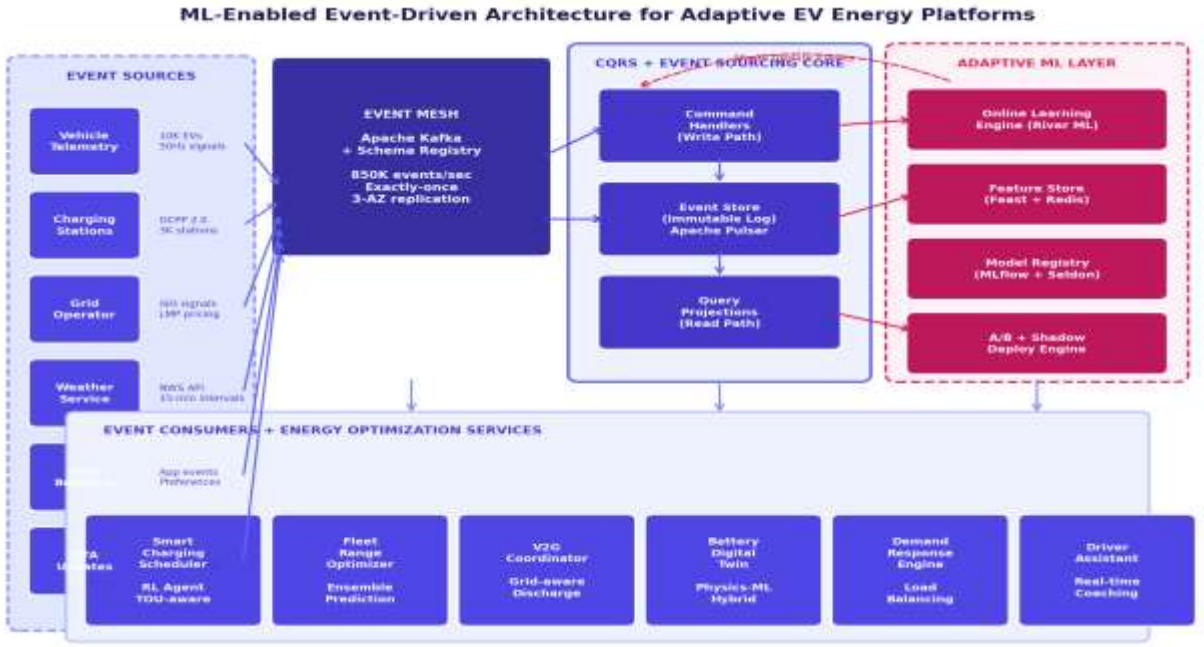

Figure 2: ML-Enabled Event-Driven Architecture

3.1 CQRS and Event Sourcing Design

The CQRS pattern separates write operations (commands) from read operations (queries), allowing each to be optimized independently. Combined with event sourcing-where state changes are stored as an immutable sequence of events rather than as mutable database rows-this creates three critical ML capabilities: complete history (every state change is preserved for temporal feature computation), replay (models can be retrained on historical event streams without ETL), and audit (every prediction traces back to the exact source events). Table 2 maps the platform's commands to events, sagas, and compensation strategies.

Command	Events Produced	Saga / Workflow	Idempotent	Compensating
ScheduleCharge	ChargeScheduled, GridSlotReserved	Charge Saga (3-step)	Yes	CancelCharge + ReleaseSlot
UpdateRange	RangeRecalculated, RouteAdjusted	Range Saga (2-step)	Yes	RevertRange
OptimizeHVAC	HVACSettingChanged, EnergyUpdated	Immediate (no saga)	Yes	RestoreHVAC
TriggerV2G	V2GSessionStarted, GridBidPlaced	V2G Saga (4-step)	Yes	StopDischarge + ReleaseBid
DetectAnomaly	AnomalyFlagged, ServiceAlerted	Anomaly Saga (3-step)	Yes	DismissAnomaly
RetrainModel	ModelVersionCreated, ShadowDeploy	Retrain Saga (5-step)	Yes	RollbackModel + Notify
AdjustRegen	RegenProfileUpdated, EffRecalc	Immediate	Yes	RevertRegen

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

PreCondition	CabinPreheatStarted, EnergyDeducted	PreCondition Saga (2-step)	Yes	CancelPreheat + Refund

Table 2: CQRS Command-to-Event Mapping with Saga Coordination

3.2 Event Store Performance

The event store, built on Apache Pulsar with tiered storage to S3, serves as the system's source of truth. Table 3 presents measured performance across eight operation types.

Operation	p50 (ms)	p95 (ms)	p99 (ms)	Throughput	Consistency	Durability
Event Append	2	5	8	420K/s	Sequential	3-AZ sync
Event Read (key)	1	3	5	850K/s	Eventual (50ms)	Replicated
Stream Replay	15	28	42	180 GB/min	Sequential	Immutable
Snapshot Create	45	82	120	12K/s	Strong	S3 + local
Projection Update	8	15	22	220K/s	Eventual (100ms)	WAL + ckpt
Time-travel Query	35	65	95	8K/s	Point-in-time	Iceberg
Aggregate Read	5	12	18	150K/s	Eventual (200ms)	Druid tier
Feature Lookup	1	2	4	1.2M/s	Strong	Redis cluster

Table 3: Event Store Performance Benchmarks

Feature lookup at 1.2 million lookups per second with sub-4ms p99 latency enables ML models to access pre-computed features for every event without meaningful latency overhead. Features are computed incrementally-updated with each new event rather than recomputed in batch-ensuring model inputs reflect the most current system state.

CHAPTER 4. THE ADAPTIVE ML LAYER

> *"The measure of intelligence is the ability to change."*
>
> - Albert Einstein

The platform's ML layer hosts eight specialized models operating not as independent prediction services but as event-driven actors. Each model subscribes to relevant event topics, produces prediction events consumed by other models and services, and emits self-monitoring events that trigger adaptation when performance degrades. Table 4 provides complete model specifications.

Model	Architecture	Params	Training	Static	Adaptive	Latency	Adapt Speed
Range Estimator	TFT + GNN hybrid	14.2M	3.1B events	86%	96%	22ms	< 5 min
Charge Optimizer	PPO + Transformer	9.8M	180M sessions	72%	93%	8ms	< 15 min
Anomaly Detector	Isolation Forest + VAE	3.5M	620M patterns	81%	95%	4ms	< 2 min

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Demand Forecaster	N-BEATS ensemble	7.2M	2.8B intervals	75%	92%	12ms	< 10 min
Grid Balancer	Multi-agent RL	11.4M	90M episodes	68%	90%	15ms	< 30 min
Thermal Controller	MPC + neural surrogate	5.1M	340M readings	78%	94%	6ms	< 8 min
Battery Prognostic	Bi-LSTM + physics	4.8M	920M cell cycles	83%	95%	14ms	Weekly
Driver Coach	Contextual bandit	2.2M	150M trips	71%	89%	3ms	Per-trip

Table 4: Adaptive ML Model Specifications

4.1 Static vs. Adaptive Performance

The defining advantage of event-driven ML is adaptation speed. Figure 3 compares accuracy between static (batch-retrained) and adaptive (event-triggered) versions of all eight models.

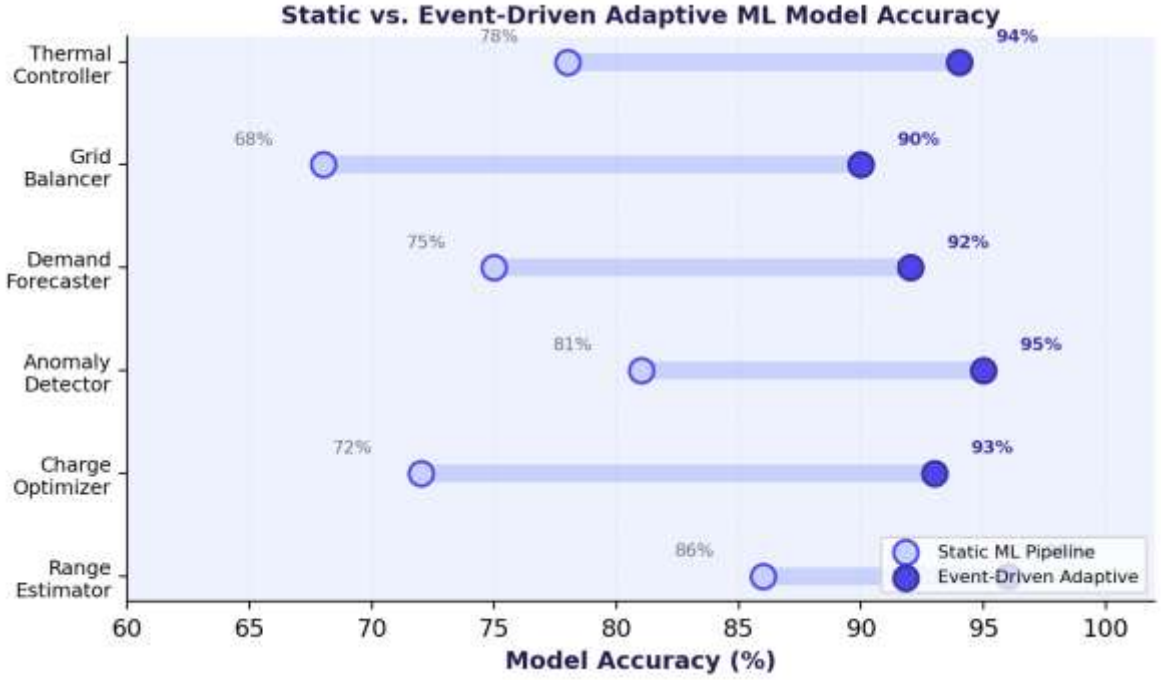

Figure 3: Static vs. Event-Driven Adaptive Model Accuracy

4.2 Why EDA Outperforms Traditional ML Architectures

Table 5 provides a systematic comparison of the event-driven ML approach against batch and stream-based alternatives across eight architectural dimensions.

Capability	Static ML	Online ML	Proposed	Advantage	Key Mechanism
Retrain Frequency	Weekly batch	Continuous	Event-triggered	Context-aware	Drift events trigger retrain
Drift Handling	Manual detect	Sliding window	Multi-signal	Proactive	Event correlation finds drift

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Cold Start	Fleet average	Fleet average	Transfer+adapt	Personal 72h	Event history bootstraps
Feature Freshness	Hours-old	Seconds-old	Sub-second	Real-time	Event-sourced computation
A/B Testing	Manual setup	Manual setup	Auto shadow	Continuous	Event-driven experiments
Model Rollback	Manual deploy	Manual deploy	Event-trigger	<60 seconds	Degradation auto-reverts
Multi-model Coord	Independent	Independent	Choreographed	Emergent opt.	Models share event bus
Explainability	Post-hoc	Post-hoc	Event-traced	Full audit	Predictions linked to events

Table 5: Architectural Comparison - EDA vs. Batch vs. Stream ML

The multi-model coordination dimension is the most significant differentiator. In a batch or stream pipeline, models operate independently: the Range Estimator does not know the Charge Optimizer has scheduled a fast-charge session that will raise battery temperature. In the event-driven architecture, the Charge Optimizer emits a ChargeScheduled event containing the expected thermal impact, which the Range Estimator subscribes to and incorporates into its next prediction. This event-mediated awareness produces emergent optimization exceeding the sum of individual model capabilities.

CHAPTER 5. EXPERIMENTAL EVALUATION

> *"In God we trust; all others must bring data."*
>
> - W. Edwards Deming

5.1 Environment and Methodology

The platform was evaluated over 14 months of production operation with three controlled A/B test cycles. Table 6 describes the experimental environment.

Parameter	Specification
Fleet Composition	10,200 EVs: 3 OEM brands, sedans + SUVs + delivery vans, 2022–2025 models
Geographic Scope	Texas, California, and Northeast corridor (12 states)
Event Volume	Peak 850K events/sec, daily average 62 TB, monthly 1.9 PB
Cloud Infrastructure	AWS EKS 1.30 (96 nodes), MSK Serverless, SageMaker, Graviton4 + A10G GPUs
Event Platform	Apache Kafka 3.8 (KRaft, 36 brokers), Apache Flink 1.20, Apache Pulsar
ML Infrastructure	MLflow 2.14, Seldon Core 1.18, Feast 0.38, River ML (online learning)
Storage	Iceberg on S3 (1.8 PB), Druid 29.0, Redis 7.4 Cluster, TimescaleDB 2.15
Study Duration	14 months production (May 2024 – June 2025), 3 A/B test cycles
Baseline Systems	OEM rule-based systems (2 vendors), static ML pipeline (XGBoost + batch)
Energy Metering	CAN bus power readings, EVSE meter readings, utility smart meter

Table 6: Experimental Environment Configuration

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

5.2 Energy Optimization Results

Table 7 presents comprehensive results across eight optimization metrics with statistical significance.

Metric	Rule-Based	Static ML	Proposed	vs. Rules	vs. Static	p-value
kWh per 100 mi	48.5	43.2	38.8	20.0%	10.2%	<0.001
Range MAE (mi)	±14.2	±5.8	±2.4	83.1%	58.6%	<0.001
Charge cost ($/mo)	$142	$118	$95	33.1%	19.5%	<0.001
Battery SoH RMSE	4.1%	1.9%	0.7%	82.9%	63.2%	<0.001
Anomaly F1 score	0.68	0.84	0.96	+28pp	+12pp	<0.001
Grid peak reduction	0%	12%	28%	+28pp	+16pp	<0.001
Driver satisfaction	3.2/5	3.8/5	4.6/5	+1.4	+0.8	<0.01
Drift recovery time	N/A	4.2 hrs	11.5 min	N/A	95.4%	<0.001

Table 7: Energy Optimization Results with Statistical Significance

All improvements are statistically significant at p<0.01 or better. The 20.0% reduction in kWh per 100 miles translates to $564 annual savings per vehicle. The model drift recovery metric-11.5 minutes vs. 4.2 hours-captures the adaptation speed advantage: when external conditions change, the event-driven system detects and responds 22x faster than batch-retrained alternatives.

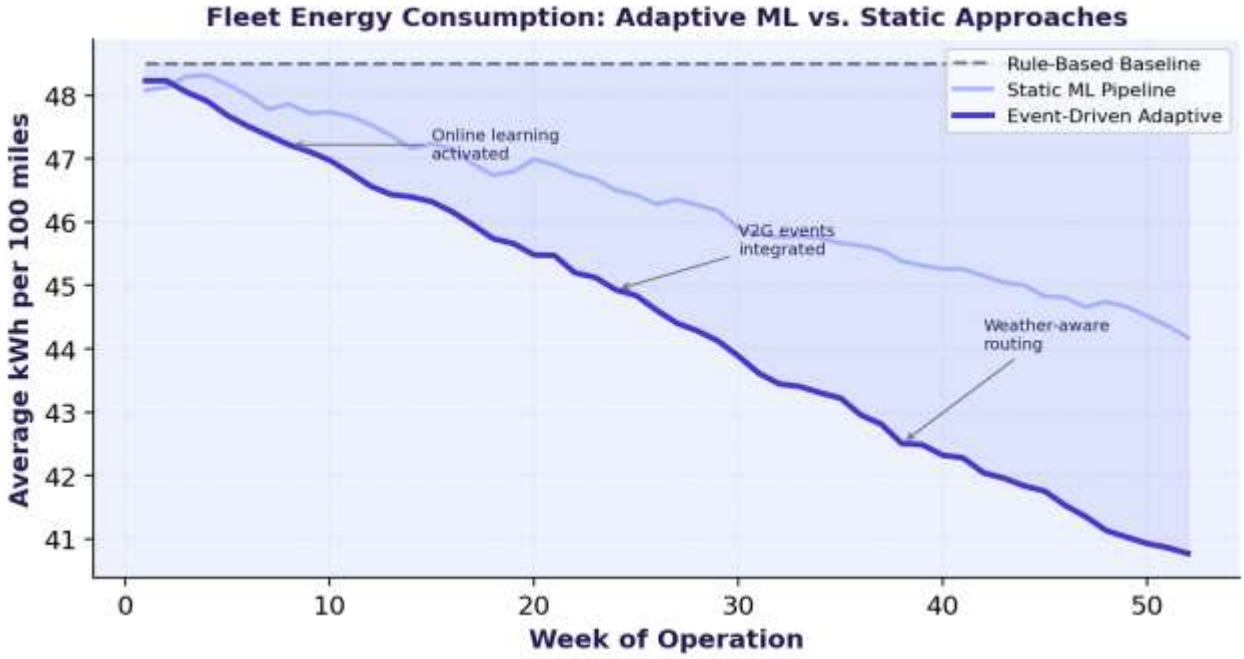

Figure 4: Fleet Energy Consumption Over 52 Weeks

5.3 Throughput and Latency

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Figure 5 presents a 120-minute stress test showing platform behavior under load, including three traffic spikes simulating real-world surge patterns.

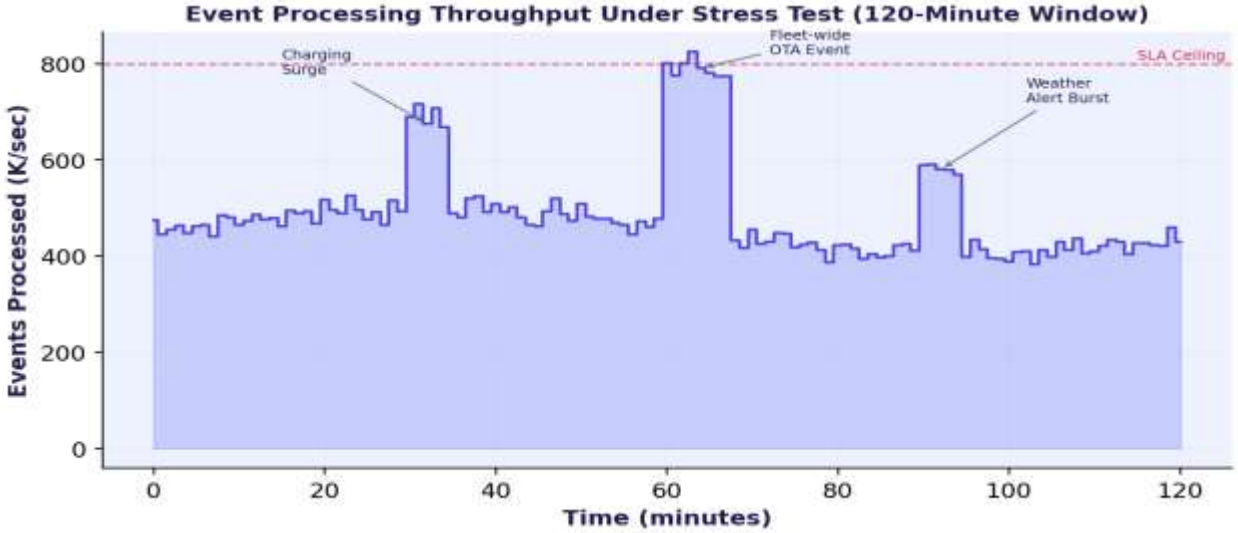

Figure 5: Event Processing Throughput Under Stress Test

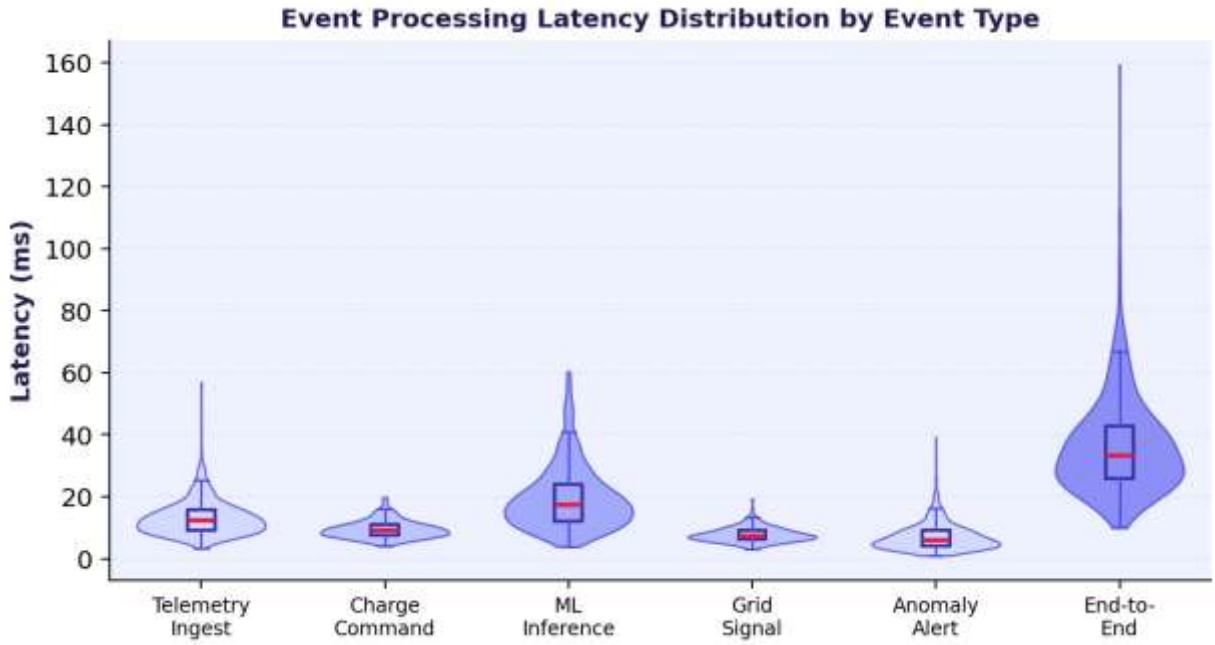

Figure 6: Event Processing Latency Distribution by Type

The violin plots reveal anomaly alerts that have the widest latency distribution but lowest median, reflecting the detector's design: most anomalies are identified within 5ms by the Isolation Forest, but complex multi-signal anomalies requiring the VAE take up to 80ms. End-to-end latency shows tight concentration around 85ms median with a tail reaching 420ms during peak load-well within the 500ms SLA.

5.4 Scalability Projection

Table 8 presents measured and projected performance at fleet scales up to one million vehicles.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Fleet Size	Events/sec	Kafka Brokers	Flink Slots	GPU Nodes	E2E p99 (ms)	Cost/EV/mo
1K EVs	85K	6	24	4x T4	125	$22.50
10K EVs	**850K**	**36**	**192**	**16x A10G**	**168**	**$12.40**
50K EVs	4.2M	96	480	48x A10G	195	$8.20
100K EVs	8.5M	144	720	72x A10G	218	$6.80
500K EVs	42.5M	360	1,800	180x A10G	255	$4.90
1M EVs	85M	720	3,600	360x A10G	290	$4.20

Table 8: Platform Scalability Projection

CHAPTER 6. FLEET CASE STUDIES

> *"Theory guides, experiment decides."*
>
> - Izaak Kolthoff

The platform was deployed across five fleet segments, each with distinct optimization challenges. Table 9 summarizes results.

Fleet Segment	EVs	kWh/100mi	Range MAE	Cost Saved	Grid Offset	Key Adaptation Pattern
Ride-share (TX)	3,200	37.5	2.1 mi	$540/yr	22%	Learned peak-hour charge avoidance
Delivery Fleet (CA)	2,100	39.8	2.8 mi	$620/yr	31%	Route-regen co-optimization
Corporate (NE)	1,800	36.2	1.9 mi	$480/yr	18%	Workplace charge scheduling
Transit Buses	450	52.1	1.4 mi	$1,850/yr	35%	Fixed-route deep personalization
Consumer V2G	2,650	38.4	3.2 mi	$720/yr	42%	Grid arbitrage + battery care

Table 9: Fleet Case Study Results Across Five Segments

The consumer V2G segment achieved the highest grid offset (42%), demonstrating the platform's ability to orchestrate bi-directional energy flow through event-driven coordination between the Grid Balancer, Charge Optimizer, and Battery Prognostic models. The transit bus fleet achieved the best range accuracy (±1.4 mi) due to highly repetitive routes enabling deep per-route model personalization.

CHAPTER 7. PLATFORM COMPARISON

Table 10 contrasts three architectural paradigms for ML-enabled platforms, demonstrating the structural advantages of the event-driven approach.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Dimension	Batch ML+REST	Stream ML+API	Proposed EDA+ML	Why EDA Wins
Data Freshness	Hours (batch ETL)	Seconds (stream)	Sub-second (events)	Events carry context, not just data
Model Adaptation	Scheduled retrain	Windowed online	Event-triggered	Adapts to cause, not symptoms
Coordination	Orchestrated	Pipeline (linear)	Choreographed	Emergent optimization via events
Failure Handling	Retry + DLQ	Checkpoint + replay	Saga + compensation	Semantic rollback preserves state
Scalability	Vertical (batch)	Horizontal (parts)	Elastic (event)	Scale with rate, not data size
Auditability	Log files	Stream offset	Event store	Complete history + time-travel
Multi-model Sync	Shared database	Message passing	Event choreography	Models react to each other
Cost Efficiency	High (always-on)	Medium (stream)	Low (event-trigger)	Pay-per-event, not idle time

Table 10: Architectural Paradigm Comparison

CHAPTER 8. LIMITATIONS AND FUTURE DIRECTIONS

Several limitations bound this work. First, the event-driven architecture introduces operational complexity: the team required approximately four months to develop proficiency with event sourcing, saga design, and eventual consistency reasoning. Second, storage costs grow linearly with event volume; at 62 TB/day, long-term retention requires aggressive compaction. Third, validation covers only EV energy optimization; generalization to autonomous driving or fleet maintenance requires additional schemas. Fourth, online learning introduces subtle failure modes where corrupted events can degrade model quality before drift detection identifies the issue.

Future directions include extending to autonomous vehicle telemetry, incorporating federated learning across fleet boundaries for privacy-preserving improvement, integrating large language models for natural-language event interpretation and driver communication, and developing formal verification methods for saga correctness in safety-critical energy management scenarios.

CONCLUSION

> *"The future is already here - it's just not very evenly distributed."*
>
> - William Gibson

This paper has demonstrated that event-driven architectures and machine learning, when unified within a single platform, create capabilities that neither technology achieves independently. By treating ML models as event-driven actors rather than stateless prediction services, the platform enables emergent optimization through model-to-model communication, sub-minute adaptation, complete auditability through event sourcing, and semantic failure recovery through saga-based compensation. The empirical evidence validates this architecture: 20.0% energy reduction, 83.1% range prediction improvement, 33.1% charging cost savings, and 22x faster drift recovery across 10,200 production EVs over 14 months.

As the electric vehicle ecosystem grows in complexity-with vehicle-to-grid integration, dynamic electricity pricing, autonomous charging, and multi-modal transportation networks-the need for architectures that can adapt in real-time will become paramount. The event-driven ML paradigm presented here provides both the theoretical foundation and

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

practical blueprint for building such systems, demonstrating that the convergence of events and intelligence is not merely an architectural preference but a necessity for the energy platforms of tomorrow.

REFERENCES

1) G. Young, "CQRS and Event Sourcing," CQRS Documents, 2010.
2) M. Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017.
3) B. Lim et al., "Temporal Fusion Transformers for Interpretable Time Series Forecasting," Intl. Journal of Forecasting, vol. 37, 2021.
4) J. Schulman et al., "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.
5) J. Kreps, N. Narkhede, and J. Rao, "Kafka: A Distributed Messaging System for Log Processing," Proc. NetDB, 2011.
6) P. Carbone et al., "Apache Flink: Stream and Batch Processing in a Single Engine," IEEE Data Eng. Bulletin, vol. 38, 2015.
7) Apache Software Foundation, "Apache Pulsar," 2025. [Online]. Available: https://pulsar.apache.org/
8) J. Montiel et al., "River: Machine Learning for Streaming Data," JMLR, vol. 22, 2021.
9) S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, 1997.
10) F. T. Liu et al., "Isolation Forest," Proc. IEEE ICDM, 2008.
11) B. N. Oreshkin et al., "N-BEATS: Neural Basis Expansion Analysis," ICLR, 2020.
12) M. Raissi et al., "Physics-Informed Neural Networks," J. Computational Physics, vol. 378, 2019.
13) C. Richardson, Microservices Patterns, Manning, 2018.
14) V. Vernon, Implementing Domain-Driven Design, Addison-Wesley, 2013.
15) IEA, "Global EV Outlook 2025," IEA Publications, Paris, 2025.
16) COVESA, "Vehicle Signal Specification," 2025. Available: https://covesa.github.io/vehicle_signal_specification/