



# ML-Powered IoT: From Warehouse Data to Production Intelligence in Real Time

## Elevator Pitch

By : Dinesh Kumar Garg | Honeywell, USA

Conf42 Machine Learning 2026

# Beyond Data Collection: The Real Challenge



Modern warehouses generate multiple terabytes of operational data monthly per facility. But raw data doesn't equal actionable intelligence.

**The Critical Gap:** Most ML projects fail to reach production because they can't bridge the divide between theoretical models and real-world reliability.

## Noisy Sensors

Extract reliable insights from imperfect readings

## Equipment Failures

Handle inevitable hardware issues gracefully

## Evolving Conditions

Maintain accuracy as operations change

# Why Traditional ML Approaches Fail in Production

Understanding common failure modes is the first step toward building systems that actually work. Production ML faces challenges invisible during development.

1

## Data Quality Issues

- Sensor failures and missing data
- Data drift from equipment aging
- Network interruptions
- Environmental interference

2

## Integration Challenges

- API incompatibilities with existing systems
- Latency mismatches between components
- System synchronization issues
- Inadequate error handling

3

## Performance Degradation

- Unexpected operational scenarios
- Out-of-distribution data
- Overconfident predictions
- Model drift over time

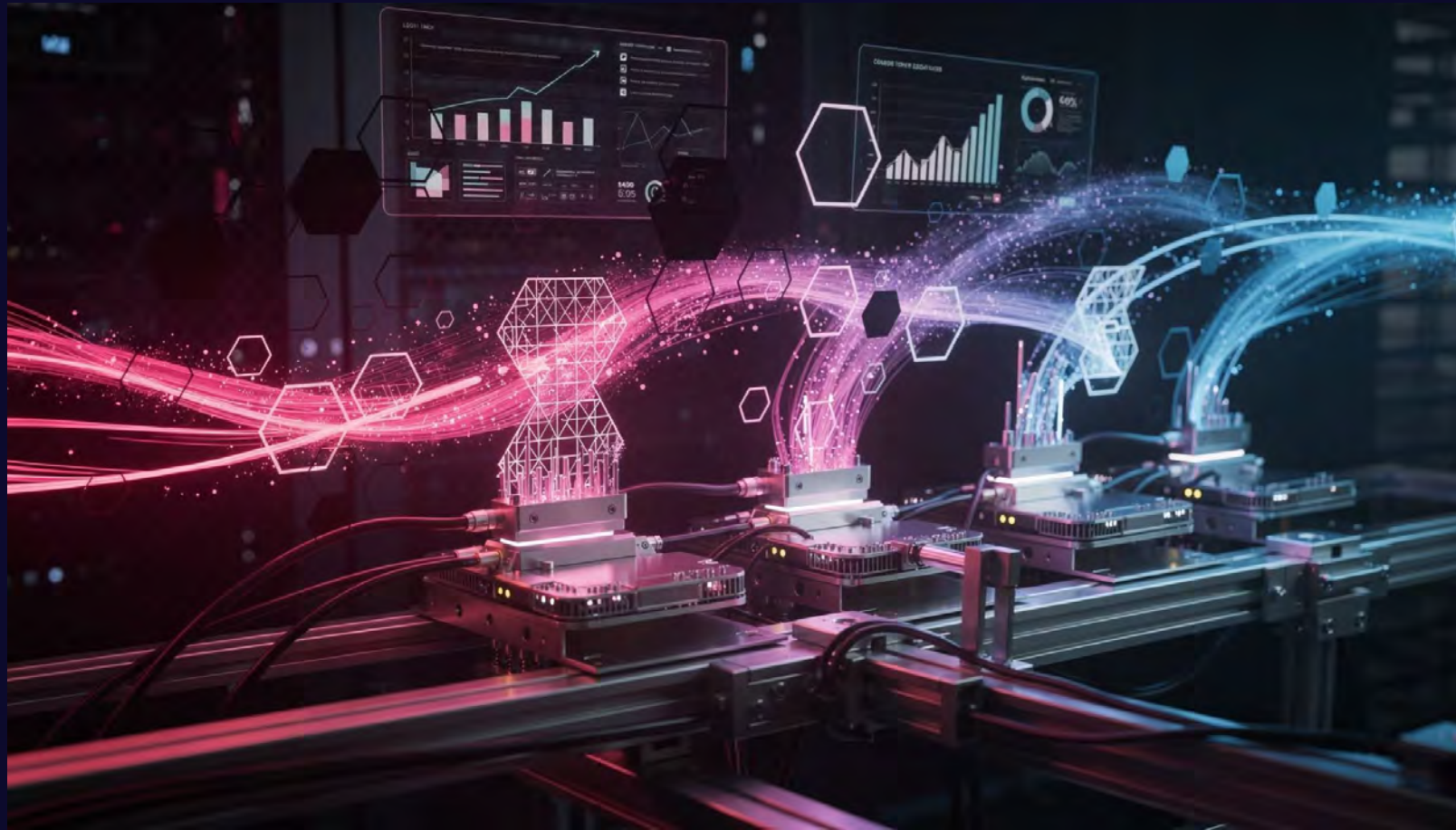
# From Raw IoT Data to ML Features

Feature engineering forms the foundation of production ML. Unlike lab environments, warehouse data arrives with temporal dependencies, missing values, and real-time computation constraints.

The choice between edge and cloud processing fundamentally shapes your feature engineering strategy and determines what's computationally feasible.

| Technique                 | Purpose                  | Application                     |
|---------------------------|--------------------------|---------------------------------|
| Rolling Windows           | Captures temporal trends | Short/medium/long-term patterns |
| Time-since-event tracking | Anomaly detection        | Equipment monitoring            |
| Time-series imputation    | Handles missing data     | Sensor failure scenarios        |
| Edge computation          | Low latency              | Real-time decisions             |

# Handling the Messiness: Missing Data & Sensor Failures



Continuous data loss in production environments is inevitable. Network interruptions and hardware failures are facts of life, and naive handling destroys model accuracy.

01

## Time-series-aware imputation

Considers temporal context and uses similar historical periods to maintain accuracy close to actual readings

02

## Multi-sensor fusion

Leverages correlations between sensor types to infer missing values from related sensors

03

## Metadata features

Treats missingness patterns as signals that often indicate equipment health issues

# MLOps Architecture for Production

Building systems that actually work at scale requires comprehensive MLOps infrastructure. The difference between a prototype and production isn't the algorithm—it's the engineering.



## Containerization & Model Serving

- Microservices architecture
- Low latency for real-time predictions
- High service availability
- Horizontal scaling for traffic spikes



## Model Versioning & Deployment

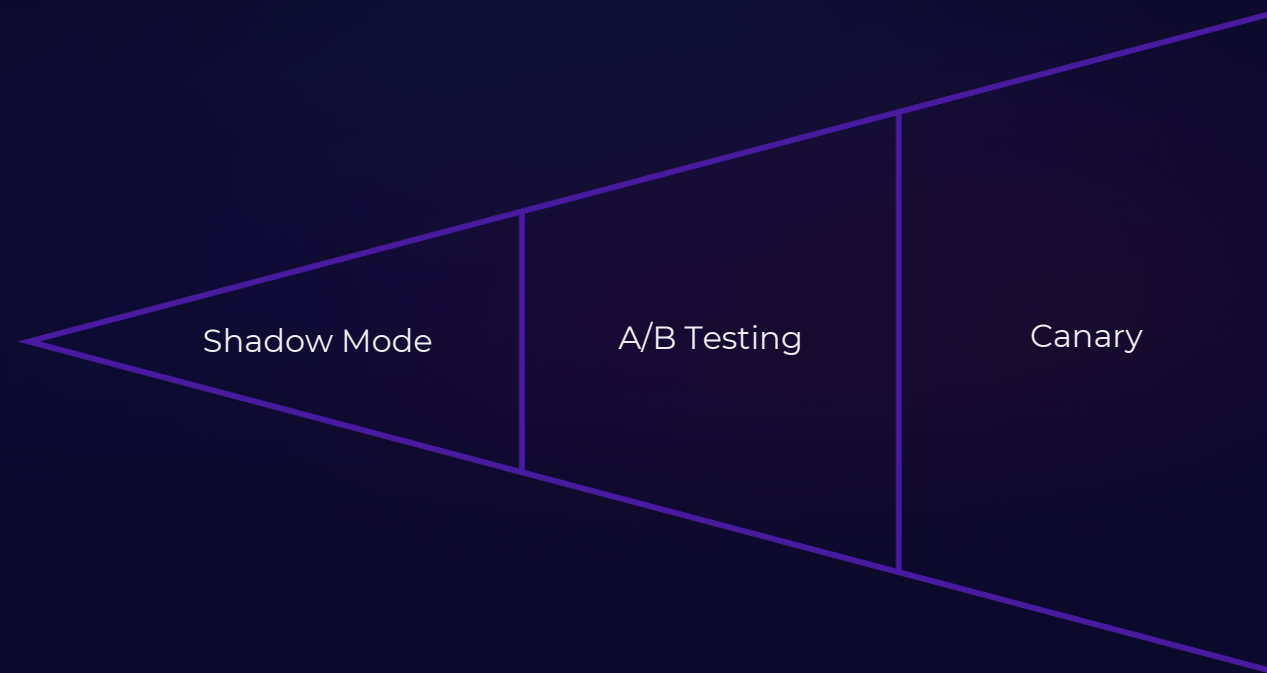
- Comprehensive model registries
- Metadata tracking
- Canary deployments with limited traffic
- Automated rollback capabilities



## Continuous Monitoring

- Distribution drift detection
- Performance degradation alerts
- Automated retraining triggers
- Business outcome tracking

# Deployment Strategies That Reduce Risk



Safe production rollout requires progressive deployment with safety gates at each stage. This approach validates performance under real conditions while maintaining the ability to quickly rollback if issues emerge.

## Shadow Mode

Run predictions alongside existing systems with no operational impact to compare with current practices

## Canary Deployment

Route small percentage of production traffic with intensive monitoring and quick rollback capability

## A/B Testing

Split traffic randomly for statistical comparison and validation of improvement claims

## Full Rollout

Complete traffic migration with standard monitoring and continuous improvement cycles

# Model Drift Detection & Retraining

The accuracy of predictive models can degrade over time due to the dynamic nature of warehouse environments. Key factors include shifting seasonal patterns, the introduction of new products, equipment aging, and evolving operational processes.

To proactively combat this degradation, our automated response strategy continuously monitors for model drift. Upon early detection, retraining pipelines are seamlessly triggered, incorporating the most recent production data to ensure sustained model relevance and accuracy.

# Edge vs. Cloud: Distributed ML Architecture

## Edge Inference

**Use Case:** Time-critical operations like AGV navigation, real-time routing, and collision avoidance

**Characteristics:** Ultra-low latency with lightweight, optimized models. Quantization and pruning enable deployment on constrained hardware.

**Trade-off:** Slight accuracy reduction for speed

---

## Cloud Inference

**Use Case:** Complex analytics including demand forecasting, strategic optimization, and multi-facility insights

**Characteristics:** Higher latency acceptable with full model power and precision ensembles

**Trade-off:** No accuracy compromise



**Hybrid Benefits:** Dramatic latency reduction for critical operations, lower cloud computing costs, reduced bandwidth requirements, and maintained prediction quality across all applications.

# Production ML Applications in Action



## Predictive Maintenance

Monitor equipment parameters continuously using vibration sensors, temperature, and power consumption. Classification models predict failure risk while regression models estimate remaining useful life.

**Impact:** Significant downtime reduction and extended equipment life



## Demand Forecasting & Inventory Optimization

Ensemble models combine time series for stable patterns, gradient boosting for complex interactions, and deep learning for long-term dependencies.

**Impact:** Reduced excess inventory and fewer stockouts



## Intelligent Picking & Routing

Real-time optimization using reinforcement learning for strategy discovery and graph neural networks for path optimization.

**Impact:** Reduced travel time and improved picking accuracy

# Fairness, Explainability, and Trust

Automated decisions affect worker assignments, performance evaluations, and resource allocation. Trust is essential for operational acceptance, and interpretability is required for adoption.

## Fairness & Bias Mitigation

**Techniques:** Pre-processing, in-processing, and post-processing approaches

**Monitoring:** Demographic parity and equalized odds metrics

**Goal:** Equitable treatment across workforce

## Model Explainability

**Global Explanations:** Feature importance and model behavior patterns

**Local Explanations:** SHAP values and counterfactuals for individual predictions

**Goal:** Operations teams understand recommendations

## Rigorous Validation

**Offline:** Temporal validation and cross-facility testing

**Online:** Shadow mode, canary deployments, and A/B tests

**Adversarial:** Edge case testing and stress scenarios

**Goal:** Reliable performance across all conditions

# Model Optimization for Edge Deployment

Edge devices have limited computational resources, battery constraints, and real-time inference requirements—yet still need sophisticated model capabilities.



## Quantization

Reduce numerical precision for significant memory reduction and faster inference with minimal accuracy loss



## Pruning

Remove redundant parameters using structured pruning for hardware efficiency and dramatic acceleration



## Knowledge Distillation

Train compact "student" from large "teacher" to retain accuracy with substantially fewer parameters

## Production Results

These optimization techniques enable sophisticated models on industrial IoT gateways while maintaining production-quality predictions.

# Emerging Trends Reshaping Warehouse ML



## AutoML & Automated Feature Engineering

Democratizes ML for operations teams with rapid development cycles. Deep feature synthesis discovers non-obvious patterns, enabling automated pipelines from raw data to deployed models.



## Multimodal Learning & Sensor Fusion

Integrates diverse sensor types—computer vision, RFID, and environmental sensors—to learn cross-modal relationships. Enables richer operational understanding and better anomaly detection.



## 5G-Enabled Real-Time Intelligence

Ultra-low latency communication with massive device connectivity. Network slicing for ML workloads enables autonomous vehicle coordination and sub-millisecond decision-making.



## Sustainable ML & Green Computing

Energy-efficient model architectures with carbon-aware training schedules. Hardware optimization and workload management reduce environmental impact without compromising performance.

# Lessons from the Field: Success Patterns

## What Separates Winners from Failed Projects

### Incremental Deployment Strategy

Start with limited scope pilots to prove value before scaling. Refine processes and build organizational confidence.

**Result:** Faster deployment, fewer incidents

### Cross-Functional Integration

Embed data scientists with operations for deep business context understanding. Regular feedback loops enable joint model validation.

**Result:** Higher adoption, better alignment

### MLOps Investment

Dedicate significant resources to production engineering, automated testing, monitoring, deployment, and documented procedures.

**Result:** Higher uptime, faster issue resolution

## Anti-Patterns to Avoid

- Big-bang deployments
- Siloed data science teams
- Algorithm-only focus
- Insufficient monitoring
- Reactive approach to issues

📌 **Key Insight:** Successful teams invest as much in deployment infrastructure and cross-functional collaboration as they do in model development.

# From Insights to Implementation

## 1 Production ML $\neq$ Academic ML

Data quality, integration, and monitoring comprise most of the work. MLOps investment is non-negotiable. Production engineering matters more than algorithms.

## 2 Start Small, Scale Smart

Incremental deployment reduces risk while proving value. Build organizational confidence progressively and learn before scaling.

## 3 People + Process + Technology

Cross-functional teams outperform siloed approaches. Explainability drives adoption, and collaboration beats sophistication.

## 4 Architecture Matters

Edge-cloud hybrid optimizes latency and cost. Continuous monitoring maintains accuracy across distributed operations.

## Next Steps

01

Assess your current ML maturity level

02

Identify high-value use case for pilot

03

Invest in MLOps infrastructure early

04

Build cross-functional implementation team

05

Plan deployment strategy with safety gates