

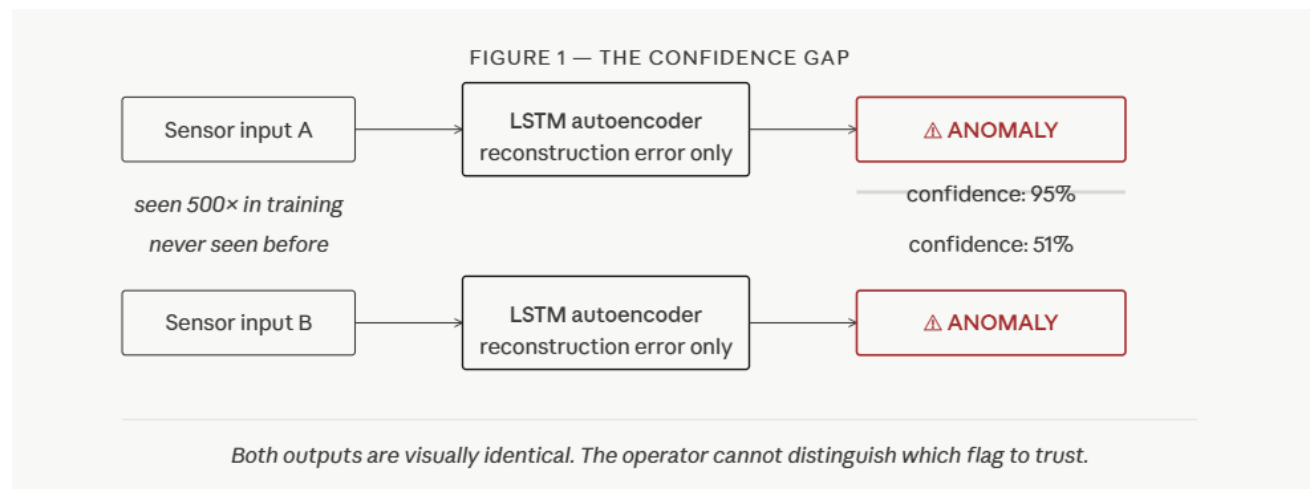
Uncertainty Quantification in Anomaly Detection as an AI Safety Primitive

Siddharth Mohan Jha

Apart Research Global South AI Safety Hackathon — June 2026

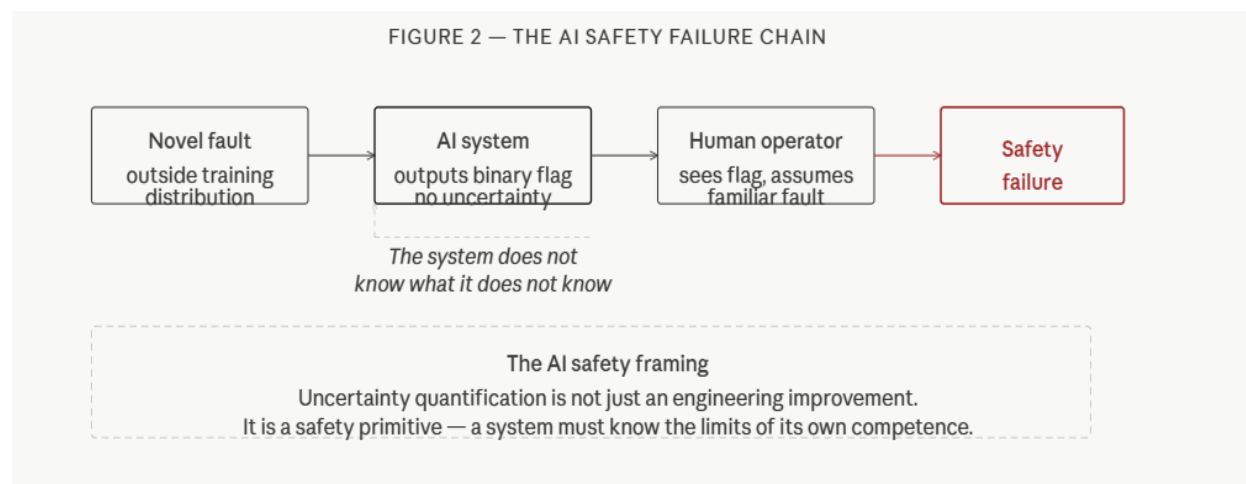
Section 1 : The Problem

Most anomaly detection systems, including the one I built, tell you that something is wrong but not how certain they are that it is wrong. A vehicle telemetry system that flags a fault with 95% confidence and one that flags it with 51% confidence look identical in their output i.e both say anomaly. In safety-critical domains like autonomous vehicles, industrial equipment, or agricultural systems, that missing confidence information is not a minor gap. It is itself a safety problem. A system that cannot say 'I am uncertain' will act or cause humans to act with false confidence. The consequences of a missed fault or a false alarm in these settings are not just inconvenient. They can be catastrophic.



Section 2 : Why This Is an AI Safety Problem

The inability of a system to express uncertainty about its own outputs is not unique to anomaly detection. It is a specific instance of a broader problem that sits at the heart of AI safety: systems that do not know what they do not know. A model trained on normal vehicle telemetry has never seen certain fault patterns. When it encounters one, it does not say 'this is outside my experience.' It either flags it or doesn't, with the same confidence it brings to patterns it has seen a thousand times. This is a form of miscalibration. The system's expressed confidence does not reflect its actual reliability. In AI safety terms, this matters because human operators downstream trust the system's outputs. A binary anomaly flag with no uncertainty estimate forces the operator to treat all flags equally. The familiar fault and the completely novel one look the same. This is how automation creates new failure modes rather than eliminating them. Uncertainty quantification is therefore not just an engineering improvement. It is a safety primitive: a foundational property that any AI system deployed in high-stakes settings should be required to have.



Section 3 : A Framework for Uncertainty-Aware Anomaly Detection

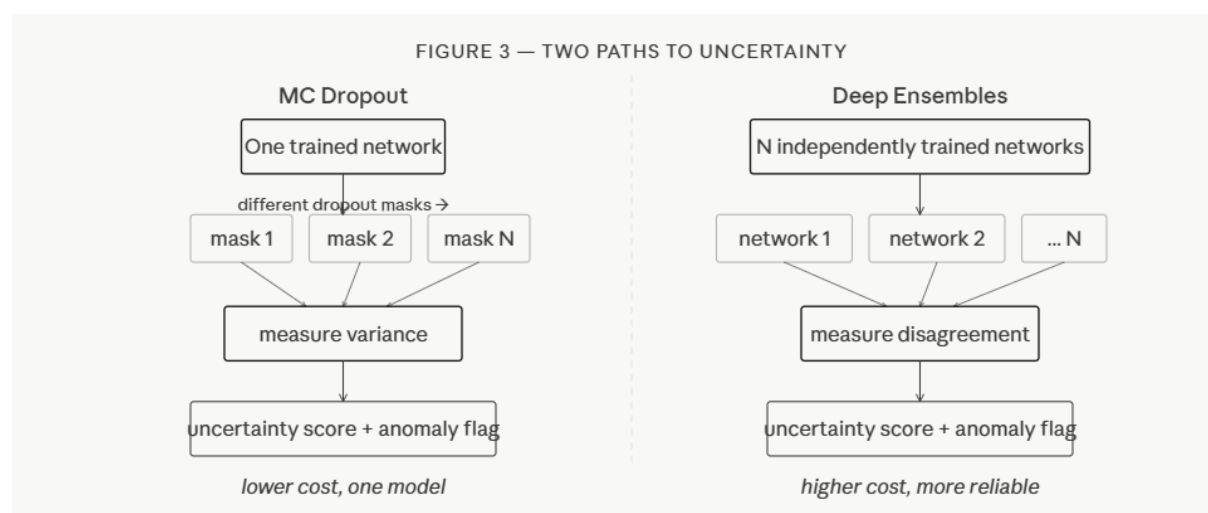
The core proposal is simple: instead of running one anomaly detector and trusting its output, run many and measure how much they disagree. Disagreement is uncertainty made visible.

Two existing methods achieve this.

The first is Monte Carlo Dropout. A standard LSTM Autoencoder uses dropout during training to prevent overfitting by randomly deactivating neurons so the network doesn't rely too heavily on any single path. Normally dropout is switched off at inference time. MC Dropout keeps it active. Running the same input through the network multiple times with different neurons randomly dropped produces multiple different reconstruction errors for the same input. If these errors are consistently high and tightly clustered, the network agrees the input is anomalous. If they are wildly spread, the network is uncertain. The variance across runs is a direct measure of epistemic uncertainty: uncertainty that comes from the model not having seen enough data like this input before.

The second is Deep Ensembles. Instead of one network with dropout, train multiple completely separate networks on the same normal data. Each learns a slightly different representation of normality. At inference, all networks produce a reconstruction error for the same input. High disagreement across networks signals that the input is genuinely unlike anything in the training distribution which is the most dangerous category of anomaly, because it is the one the system has no basis for evaluating.

Together these two methods transform a binary anomaly flag into a three-part output: is this anomalous, how confident are we, and is this uncertainty coming from noise or from genuine novelty? That third question, noise or novelty, is the one that matters most for safety-critical deployment. A novel fault the system has never encountered deserves immediate human attention regardless of the reconstruction error magnitude. A noisy but familiar pattern can be handled automatically. Current systems cannot make this distinction. This framework can.



Section 4 : Evaluation Criteria

Proposing a framework without specifying how to test it is not research, it is speculation. Three evaluation criteria would determine whether this framework actually works in practice.

The first is calibration error. A well-calibrated uncertainty estimate means that when the system says it is 90% confident, it is right 90% of the time. Calibration error measures the gap between expressed confidence and actual accuracy. A system with low calibration error is trustworthy. A system with high calibration error is potentially more dangerous than no uncertainty estimate at all because it gives operators false confidence in its confidence.

The second is uncertainty-fault correlation. If the framework is working correctly, higher uncertainty should correlate with higher likelihood of a genuinely novel or serious fault. This can be tested by introducing known fault patterns into the telemetry data, some familiar, some deliberately novel and then measuring whether the uncertainty signal correctly identifies the novel ones as requiring more attention.

The third is false negative rate at different uncertainty thresholds. A fixed anomaly threshold treats all flags equally. An uncertainty-aware system allows operators to set different response protocols depending on confidence level - automatic handling for high-confidence familiar faults, human review for uncertain or novel ones. The evaluation question is whether this tiered response reduces dangerous false negatives without flooding operators with unnecessary alerts.

These three criteria together answer the question that matters for safety-critical deployment: not just does the system detect anomalies, but does it know when to trust itself?

FIGURE 4 — EVALUATION CRITERIA

1. Calibration error	2. Uncertainty-fault corr.	3. Tiered FN rate
Does stated confidence match true accuracy? <i>90% confident should be right 90% of the time</i>	Does high uncertainty predict novel faults? <i>inject known faults, some familiar, some novel</i>	Does tiered response reduce missed faults? <i>auto-handle confident flags, escalate uncertain ones</i>
→ measure gap between stated vs true rate	→ novel faults should trigger higher uncertainty	→ fewer dangerous false negatives

Core question: does the system know when to trust itself?

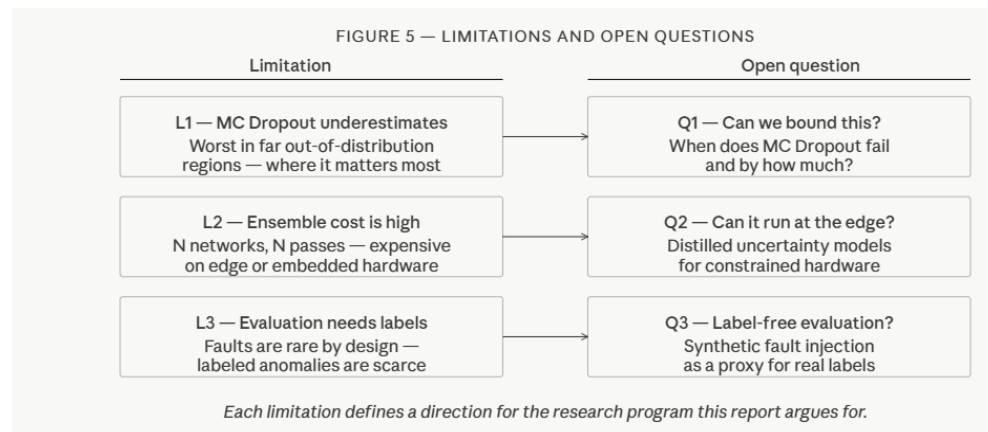
Section 5 — Limitations and Open Questions

Intellectual honesty requires naming what this framework does not solve. First, MC Dropout has known limitations as an uncertainty estimator. It tends to underestimate uncertainty in regions far from the training distribution which are exactly the cases where uncertainty matters most. Deep Ensembles are more reliable but computationally expensive, requiring multiple full networks to be trained and run at inference. In resource-constrained deployment environments —like edge devices on vehicles or agricultural equipment, this overhead may be prohibitive.

Second, the framework assumes that disagreement between networks is a reliable proxy for genuine novelty. This is not always true. Networks can agree confidently on a wrong answer if they were all trained on the same biased data. Uncertainty quantification does not solve the problem of training data quality, it assumes the training distribution of normal data is itself representative.

Third, the evaluation criteria proposed here require labelled fault data to test properly. In real safety-critical deployments, labelled anomalies are scarce by definition i.e if faults were common, the system would not be safety-critical. This creates a fundamental evaluation challenge that this framework does not resolve.

These limitations point to the open questions this work raises rather than answers. Can uncertainty estimates be made computationally cheap enough for edge deployment? Can disagreement be distinguished from bias? Can evaluation frameworks be designed that do not depend on labelled anomaly data? These are the directions a full research program would pursue. This report establishes the case for why that program is worth pursuing.



References

- 1)Gal, Y. & Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. ICML.
- 2)Lakshminarayanan, B. et al. (2017). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. NeurIPS.
- 3)Pang, G. et al. (2021). Deep Learning for Anomaly Detection: A Review. ACM Computing Surveys.