



Explainable Machine Learning for Intelligent Big Data Analytics in Cloud Computing Environments

George Emmanuel

Independent Researcher, Midwestern, United States

Email ID: George_emmohio3102@yahoo.com

Abstract

Machine learning (ML) applied to big data in the cloud now drives decisions about credit, maintenance, fraud, health, and operations. The most accurate models are often the hardest to understand, and at big data scale the problem becomes harder still. Explaining millions of predictions is computationally expensive, explanations must reach audiences with very different needs, and regulators increasingly expect every automated decision to be traceable. This article proposes an architecture for explainable ML in cloud-based big data analytics. It first identifies seven challenges that arise when explainability meets scale. It then sets out a layered architecture that begins with an interpretability-first model selection gate. The architecture provides explanation services using feature attribution and counterfactual methods, and places explanation computation according to latency and cost. It aggregates local explanations into global understanding, delivers explanations tailored to each audience, and routes decisions through configurable workflows. Explanations are stored as audit evidence, and bounded autonomous agents that must themselves explain their actions operate the system. A mapping of audiences to questions, methods, and formats guides explanation design. Using a design-oriented approach grounded in ten studies published between 2016 and 2025, the article illustrates the architecture with an energy utility predicting equipment failures across its grid. It argues that explainability at scale must be designed into the analytics architecture, not added to individual models afterward.

Keywords: explainable AI, interpretable machine learning, big data analytics, cloud computing, SHAP, LIME, counterfactual explanations, auditability, compliance, autonomous agents, enterprise workflow



1. Introduction

Big data analytics has moved from reporting what happened to predicting what will happen and recommending what to do. Cloud computing supplies the storage and computing capacity to train complex models on large datasets and apply them continuously. Gradient-boosted ensembles and deep neural networks often achieve the best accuracy on these tasks, but they do not reveal how they reach their outputs.

This opacity creates practical problems. Engineers cannot tell whether a model has learned real patterns or data artifacts. Managers hesitate to act on recommendations they cannot question. People affected by automated decisions cannot understand or contest them. Regulators and auditors cannot verify that decisions are fair and lawful. Doshi-Velez and Kim (2017) argue that interpretability is needed when a problem is not fully specified, meaning when accuracy alone cannot capture everything that matters, such as safety, fairness, or legal compliance. Most consequential applications of big data analytics fall into this category.

Explainable ML research has produced useful methods, including local explanations (Ribeiro et al., 2016), efficient attribution for tree models with ways to combine local explanations into global understanding (Lundberg et al., 2020), and counterfactual explanations (Wachter et al., 2018). Using these methods at big data scale in the cloud raises new questions. Where should explanation computation run, and at what cost? How can millions of explanations be stored, searched, and summarized? How should explanations differ for engineers, managers, affected individuals, and auditors? How can explanations be protected against manipulation (Slack et al., 2020)? How should they connect to decision workflows (Basireddy, 2022) and compliance records (Basireddy, 2023)?

This article asks: How can explainable ML be delivered systematically for big data analytics in cloud environments? It contributes an analysis of the challenges of explainability at scale, a layered architecture, an audience-oriented explanation design model, and an illustrative application.

2. Related Work

2.1 Why and when to explain

Doshi-Velez and Kim (2017) propose a framework for the science of interpretability. They

distinguish evaluation based on applications with real users, evaluation based on simplified human tasks, and evaluation based on formal proxies. They argue that explanations are needed when goals such as safety, fairness, or scientific understanding cannot be fully captured in the model's objective.

2.2 Interpretable models versus post-hoc explanation

Rudin (2019) argues that for high-stakes decisions, inherently interpretable models should be preferred to black-box models explained after the fact. Post-hoc explanations may not faithfully represent what the model does, and interpretable models often perform nearly as well on structured data.

2.3 Feature attribution

Ribeiro et al. (2016) introduced LIME, which explains an individual prediction by fitting a simple model around it. LIME works with any model type. Lundberg et al. (2020) developed an exact, efficient method for computing SHAP values for tree-based models. Tree ensembles are widely used in big data analytics, so this matters. They also showed how to combine many local explanations into global insights, such as overall feature importance, interaction effects, and patterns across groups of instances. Efficient computation and the step from local to global are both essential at big data scale.

2.4 Counterfactual explanations

Wachter et al. (2018) proposed counterfactual explanations, which describe the smallest change to the input that would have produced a different outcome, for example "the loan would have been approved if income had been higher by a certain amount." Counterfactuals give affected individuals something they can understand and act on without revealing the model's internals. The authors also discuss their relevance to individuals' rights under the EU General Data Protection Regulation.

2.5 Risks of post-hoc explanations

Slack et al. (2020) showed that LIME and SHAP can be fooled. A model can be built to behave in a biased way on real data while showing harmless explanations when probed by the explanation method. This shows that explanations can be manipulated, and that explanation



methods need safeguards and independent checks, especially where explanations serve as compliance evidence.

2.6 Cloud architecture

Sannidhanam (2020) presents a comparative analysis of cloud computing architectures for enterprise applications, showing trade-offs in scalability, performance, cost, and security. Explanation computation can cost more than prediction itself, so architectural choices about where and when explanations are computed have large effects on cost and latency.

2.7 Autonomous operations

Sannidhanam (2025) examines autonomous AI agents for cloud infrastructure operations, which monitor systems and take corrective actions within defined limits under human oversight. Such agents can manage explanation services, for example scaling computation when explanation demand rises. The agents themselves take consequential actions, so they too must explain what they did and why.

2.8 Workflows and audit

Basireddy (2022) proposes configurable enterprise workflow architecture for digitizing risk management, with defined steps for assessment, approval, escalation, and remediation and complete audit trails. Explanations make such workflows more effective by showing reviewers why a case was flagged. Basireddy (2023) proposes an audit-ready compliance platform architecture for enterprises that use large language models in regulated settings, which keeps traceable records of model inputs, outputs, versions, and decisions. Extending such records to include explanations lets auditors see not only what a model decided but why.

2.9 Research gap

These strands have developed largely separately, leaving limited guidance on delivering explainability systematically across large-scale cloud analytics.

3. Research Methodology

The study follows a design-oriented research approach. The problem, that explainability methods designed for individual predictions do not by themselves provide explainability across

large-scale cloud analytics, was identified from the literature. Challenges were analyzed (Section 4) and translated into requirements. A layered architecture and an audience-oriented design model were developed from ten studies and demonstrated through an illustrative energy utility case. The demonstration shows feasibility and internal consistency. Empirical evaluation is left to future work.

4. Challenges of Explainability at Big Data Scale

Table 1 summarizes seven challenges that arise when explainable ML is applied to big data in the cloud.

Table 1. Challenges of explainable ML at big data scale in the cloud

Challenge	Cause	Architectural response	Supporting studies
Computational cost	Many explanation methods need many model evaluations per prediction	Efficient exact methods for tree models, selective and precomputed explanations	Lundberg et al. (2020); Sannidhanam (2020)
Latency	Real-time decisions cannot wait for slow explanations	Fast attribution at decision time, detailed explanations on demand	Lundberg et al. (2020)
Volume	Millions of explanations are too many to inspect	Aggregate local explanations into global and group-level views	Lundberg et al. (2020)
Audience diversity	Engineers, managers, affected people, and auditors need different answers	Audience-specific explanation types and formats	Doshi-Velez & Kim (2017); Wachter et al. (2018)
Fidelity and manipulation	Post-hoc explanations can be unfaithful or gamed	Interpretable models where possible, independent explanation checks	Rudin (2019); Slack et al. (2020)
Actionability	Explanations do not change decisions unless they reach decision-makers	Explanations delivered within workflows	Basireddy (2022)
Accountability	Regulators require evidence of why decisions were made	Explanations stored as versioned audit evidence	Basireddy (2023)

5. Architecture for Explainable Big Data Analytics

The architecture has eight layers (Table 2).

Table 2. Architecture for explainable ML in cloud-based big data analytics

Layer	Function	Key mechanisms	Supporting studies
1. Interpretability	Prefer interpretable	Compare interpretable and	Rudin (2019)



gate	models where accuracy allows	complex models before deployment	
2. Explanation engine	Generate explanations	Tree SHAP, LIME, counterfactual search	Lundberg et al. (2020); Ribeiro et al. (2016); Wachter et al. (2018)
3. Computation placement	Balance explanation cost and latency	Batch precomputation, on-demand services, elastic scaling	Sannidhanam (2020)
4. Explanation store and aggregation	Store and summarize explanations	Indexed explanation store, global and group-level summaries	Lundberg et al. (2020)
5. Audience delivery	Present explanations suited to each user	Role-specific views and language	Doshi-Velez & Kim (2017); Wachter et al. (2018)
6. Workflow integration	Use explanations in decisions	Explanations attached to cases, explanation-based routing	Basireddy (2022)
7. Audit and assurance	Make explanations reliable evidence	Versioned records, fidelity checks, manipulation tests	Basireddy (2023); Slack et al. (2020)
8. Autonomous operations	Operate explanation services and explain agent actions	Bounded agents with action rationales	Sannidhanam (2025)

Interpretability gate. Before a model is deployed, it is compared with an interpretable alternative, such as a sparse linear model or a small decision tree. If the interpretable model performs within an agreed margin, it is chosen (Rudin, 2019). A complex model is used only when its extra accuracy justifies the explanation effort it requires.

Explanation engine. For tree ensembles, exact and efficient SHAP computation provides feature attributions (Lundberg et al., 2020). For other model types, LIME provides local explanations (Ribeiro et al., 2016). For decisions that affect individuals, counterfactual search finds the smallest change that would alter the outcome (Wachter et al., 2018).

Computation placement. Following the architectural trade-offs described by Sannidhanam (2020), explanations are computed where they cost least while meeting latency needs. Fast attributions are computed alongside real-time predictions. Detailed explanations and counterfactuals are computed on demand, on elastic cloud capacity. For large batch analytics, explanations are precomputed during off-peak processing.

Explanation store and aggregation. Explanations are stored with their predictions, model versions, and input data references. Aggregation turns millions of local explanations into global feature importance, interaction effects, and group-level summaries (Lundberg et al., 2020). Analysts can then see what drives the model overall and how drivers differ across regions, customer groups, or asset types.

Audience delivery. Each audience receives the explanation type and format suited to its

questions (Section 6).

Workflow integration. Explanations are attached to cases in configurable workflows (Basireddy, 2022). Reviewers see why a case was flagged. Workflows can also route cases by explanation content, for example sending cases driven mainly by a sensitive attribute to specialist review.

Audit and assurance. Explanations are stored as versioned audit evidence alongside inputs, outputs, and decisions (Basireddy, 2023). Because post-hoc explanations can be unfaithful or manipulated (Slack et al., 2020), assurance checks compare explanations against model behavior on held-out and perturbed data. Unusual inconsistencies are flagged for investigation.

Autonomous operations. Bounded agents scale explanation services with demand, restart failed jobs, and monitor explanation latency (Sannidhanam, 2025). Each agent action is recorded with a rationale that states what the agent observed, what it did, and why. Automated operations are then held to the same standard of explainability as the models they support.

6. Matching Explanations to Audiences

Different people ask different questions of the same model. Table 3 maps five audiences to their questions, suitable methods, and formats.

Table 3. Explanation design by audience

Audience	Typical question	Suitable methods	Format
Data scientists	Is the model learning real patterns?	Global SHAP summaries, interaction effects, fidelity checks	Detailed charts and statistics
Business decision-makers	Why is this case flagged, and should I act?	Local feature attribution for the case	Short ranked list of main factors in plain language
Affected individuals	Why this outcome, and what would change it?	Counterfactual explanations	Plain-language statement of actionable changes
Auditors and regulators	Were decisions consistent, fair, and traceable?	Stored explanations, group-level summaries, assurance results	Evidence packs linked to decision records
Operations reviewers	Why did the agent take this action?	Agent action rationales	Logged rationale with observed signals

The mapping follows Doshi-Velez and Kim's (2017) view that explanation requirements depend on the purpose and the user. It also uses counterfactuals where people need to understand what they could do differently (Wachter et al., 2018). A single explanation method rarely serves all audiences, so the architecture supports several methods and formats.

7. Illustrative Case: Predicting Equipment Failure at an Energy Utility

Consider a regulated electricity utility that monitors thousands of transformers and other grid assets. Sensors report temperature, load, oil quality, and vibration every few minutes, which produces billions of readings a year. The utility uses ML to predict which assets are likely to fail in the next 90 days and to prioritize maintenance.

Model choice. The data science team compares a small, interpretable scoring model with a gradient-boosted ensemble. The ensemble is substantially more accurate at identifying high-risk assets, so it passes the interpretability gate with a requirement for full explanation support (Rudin, 2019).

Explanations at scale. Each week, failure risk and tree SHAP attributions are computed for every asset in a batch job on elastic cloud capacity, then the capacity is released (Lundberg et al., 2020; Sannidhanam, 2020). Aggregated explanations show that oil quality and load cycling are the main drivers overall. For one older transformer model, internal temperature matters much more than for others, an insight engineers use to adjust inspection guidance.

Decision-maker view. An asset manager reviewing the week's highest-risk assets sees, for each one, a short list of main factors in plain language, such as "oil moisture rising for six weeks" and "frequent overload in the evening peak."

Counterfactuals. For a transformer just below the maintenance threshold, a counterfactual shows that reducing evening peak load by a modest amount through load transfer would lower its risk rating (Wachter et al., 2018). This gives the operations team an alternative to immediate replacement.

Workflow. High-risk assets open maintenance cases in the configurable workflow with explanations attached (Basireddy, 2022). Cases whose risk is driven mainly by a single sensor are routed first to a data quality check, because a faulty sensor could be the real cause.

Audit. The energy regulator reviews the utility's maintenance prioritization. The utility provides evidence packs showing, for each decision period, the model version, predictions, explanations, and resulting maintenance actions (Basireddy, 2023). Assurance checks confirm that explanations were consistent with model behavior on perturbed test data (Slack et al., 2020).

Operations. During a heatwave, the utility runs daily rather than weekly risk scoring. The operations agent scales explanation computation within approved limits and logs its rationale: rising job queue times and a scheduled increase in scoring frequency (Sannidhanam, 2025).

The case shows how explainability operates across the whole analytics system. It shapes model

choice, scales efficiently in the cloud, serves different audiences, drives workflows, provides audit evidence, and extends to automated operations.

8. Discussion

8.1 Implications

For research, the article moves explainability from a property of individual predictions to a property of analytics architectures. It identifies scale-specific challenges, including cost, volume, audience diversity, manipulation, and accountability, and links each to an architectural response. It also extends explainability to the autonomous agents that increasingly operate cloud systems.

For practitioners, the key steps are to apply an interpretability gate before accepting black-box models, place explanation computation to balance cost and latency, and store explanations as audited evidence with checks against manipulation.

8.2 Limitations and future work

The architecture was demonstrated through an illustrative case and draws on a focused set of ten studies. Future work should implement and evaluate it, measuring explanation cost and latency at scale, user understanding and decision quality for each audience, and the effectiveness of assurance checks against manipulated explanations. Further research is needed on efficient counterfactual search for large datasets, on standards for explanation evidence in regulatory audits, and on how autonomous agents should explain their actions to human operators.

9. Conclusion

As ML becomes central to big data analytics in the cloud, explainability must scale with it. This article identified seven challenges of explainability at scale and proposed an eight-layer architecture. It prefers interpretable models where possible, generates efficient and appropriate explanations, places their computation wisely in the cloud, and aggregates them into global insight. It also tailors explanations to audiences, embeds them in workflows, treats them as audit evidence, and applies the same standard to autonomous operations. The energy utility case showed the architecture in use from model selection to regulatory review. The central conclusion is that explainability at scale is an architectural property that must be designed across the analytics system, not added to individual models afterward.



References

1. Basireddy, S. R. (2022). Digitizing risk management through configurable enterprise workflow architecture. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 14(4), 231–239.
2. Basireddy, S. (2023). Audit-ready compliance platform architecture for large language model regulated enterprises. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 15(01), 226–232. <https://doi.org/10.18090/samriddhi.v15i01.40>
3. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning (arXiv:1702.08608). arXiv.
4. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67.
5. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144).
6. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215.
7. Sannidhanam, A. H. (2020). Comparative analysis of cloud computing architectures for enterprise applications. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 12(02), 169–180. <https://doi.org/10.18090/samriddhi.v12i02.16>
8. Sannidhanam, A. H. (2025). Autonomous AI agents for cloud infrastructure operations. *Journal of Contemporary Science and Technology Management*, 1(01), 83–100. <https://jcstm.info/index.php/jcstm/article/view/20>
9. Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)* (pp. 180–186).
10. Wachter, S., Mittelstadt, B., & Russell, C. (2018). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2), 841–887.