Deep Science Publishing, 2026
https://doi.org/10.70593/978-81-69589-32-1_6

DeepScience
Open Access Books

Chapter 6: Intelligent Data Engineering for Infrastructure Operations

6.1. Introduction

Infrastructures affect the society by providing crucial services that enhance quality of life and enable economic activities. Infrastructure operations are responsible for managing the continual day-to-day responsibilities of these services, such as railway signalling, power transmission switching, and telecoms call switching. While operations can be seen as a way of keeping the system as it should be, incidents can still occur that require attention. Operations rely on humans to interpret information in order to understand the full situation and take appropriate actions for the system to return to a healthy state; people are thus central to operations.

For system operations to be effective, trustworthy information must be available and consequently, accurate data contributing to this information is needed. Such data is managed by a data engineering layer, which establishes reliable and efficient methods for collecting, processing, managing, and distributing data. This data engineering layer uses various intelligent techniques that apply machine learning algorithms to provide more insightful information to operations and improve the system operation process. An overview of the data engineering layer is proposed, covering foundational principles and reference architectures, as well as the role of data engineering in operations and reliability.

6.1.1. Data Quality and Provenance

Data quality is an established data management discipline and a dimension of the overall data governance framework. Similar to reliability in engineering, quality can be defined in many different ways, depending on the context and purpose of the data under consideration. As a consequence, there is an extensive literature defining, modeling, assessing, and assuring data quality, covering several generic and use-case-specific

perspectives. The latter are typically summarized into a few common dimensions, namely accuracy, completeness, consistency, timeliness, validity, and uniqueness. Despite being vendor-agnostic, the framework of the Data Management Association remains particularly accessible and widely used.

Data provenance pursues the aim of enabling trustworthy data, thus assuring one of the foundations of any quality dimension. Provenance semantics capture the origin, history, and derivation processes of data elements. Standard schemas have been proposed to facilitate provenance documentation in terms of logical description. Provenance capture techniques can be classified according to whether they operate at the data-producing or --consuming side, the former being the most common approach. Provenance-aware data lakes are emerging, and data lineage is increasingly monitored, as these aspects are crucial for reproducibility in analytics and AI settings. Provenance-free data, however, exhibit minimal trust potential whatsoever; a fact that is particularly worrisome in the context of rapidly spreading misinformation.

Developing and implementing suitable solutions for documenting provenance and assessing the quality of the data engine operations become an imperative for both service providers and consumers. Dedicated solutions allow tracking data quality at various levels of granularity, including source definitions during ingestion, metadata catalogs summarizing governance rules, and lineage information captured through monitoring activities. Additionally, the development of quality dashboards and specific auditing workflows enables official editing.

Fig 6.1: Intelligent Data Engineering for Infrastructure Operations

6.1.2. Real-Time vs. Batch Processing Architectures

The significance of the real-time versus batch radar is determined by the specific data processing use case. The two architectures are compared across five dimensions—latency, throughput, accuracy, storage, and cost—with the respective values for each architecture represented qualitatively. The values for these architectural choices are subsequently reviewed and put into context.

The capability for low-latency ingestion and processing has driven the emergence of real-time analytics in many domains; for example, automated trading in the financial services domain. Use cases that demand real-time data availability typically have low throughput, as there is usually only a small amount of incoming data. A further distinctive facet of real-time data processing in these scenarios is that data does not have to remain completely accurate. As shown in the summation of dimensions, real-time data processing is nevertheless preferable for a range of use cases, especially those that depend on alerting or near real-time reaction. Nevertheless, batch processing remains significant as many historical functions and a range of low-latency ingestion use cases are either infeasible or uneconomical in real time or low-latency mode. A product of this demand is the emergence of hybrid patterning of streaming, low-latency, and batch jobs when the value of each latencies is exploited.

6.2. Foundations of Data Engineering for Infrastructure

For any infrastructure or industrial domain, it is a necessary prerequisite to establish an infrastructure that embodies the principles of data engineering, together with a blueprint or a reference architecture. Data engineering encompasses the processes needed to design, develop, maintain, and operationalize data pipelines. It consists of the activities required to manage and facilitate data operations, analytics, machine learning, and artificial intelligence (AI). Operational data engineering is, therefore, not a separate team but a set of activities that enable and empower all users and producers of data in a particular area of the business.

Given its importance for core operations and reliability, the data engineering foundations for infrastructure operations are based on five pillars: data ingestion and integration, data modeling and metadata management, data quality and provenance, real-time vs. batch processing architectures, and data operation tooling. Within the domain of infrastructure operations, these aspects are considered within data flows that facilitate both online and offline intelligent techniques.

6.2.1. Data Ingestion and Integration

Infrastructure systems generate diverse datasets from various sources feeding multiple use cases. On top of the database systems hosting structured and relational datasets, a wealth of data may be ingested and stored for analytics supporting business intelligence and other reporting applications—from data lakes containing raw non-relational datasets, to distributed file storage for unifying data used in machine learning experiments. Complex infrastructures typically require data from multiple sources collected in real-time, while low-latency streaming systems connect with external data sources and enable bidirectional data flows.

Data may be ingested by pulling data from third-party services or by having these services push the data automatically to a receiving application. Pulling data from a service means that the receiving application needs to poll—check for new data on a defined schedule—so that all available data is eventually collected; this approach introduces latency, running costs (computational power for keeping a polling application running), and also bears the risk of missing important updates if the appropriate polling frequency is not established. When a service pushes data, a connector on the receiving side subscribes to the service data stream so that new data is immediately received. In both approaches, connectors should cope with schema evolution, performing data cleansing and deduplication when needed; a CDC (Change Data Capture) strategy can be used to avoid capturing the same data more than once.

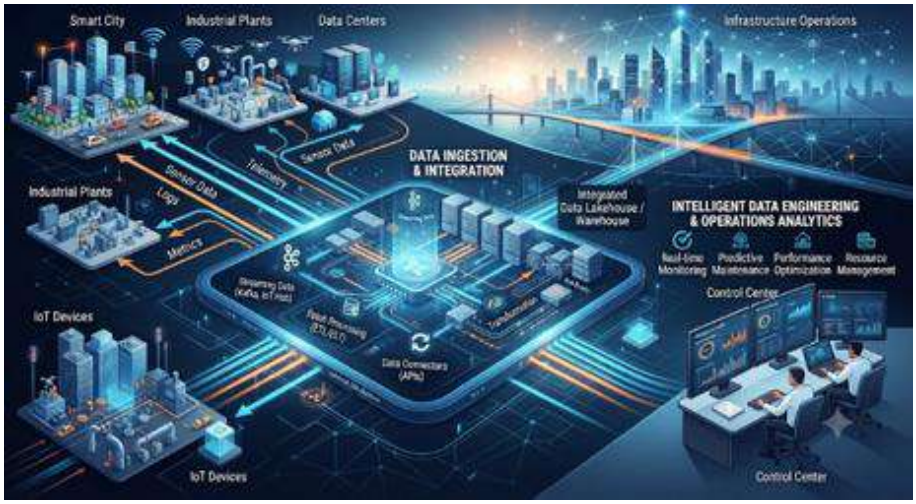

Fig 6.2: Data Ingestion and Integration

6.2.2. Data Modeling and Metadata Management

Data modeling establishes canonical data representations that reduce redundancy and enable data-sharing across applications, analysts, and data scientists. Expanding upon the schema-centric data modeling of business intelligence, for infrastructure, a specialized hierarchy of canonical models, data domain and data sources, provides semantic and structural harmony while addressing the distinctive requirements of big data systems. These are conceptually and dimensionally classified along business processes: discovery, surveillance and monitoring, maintenance, repair, and restoration (DISM-R3). Metadata management, typically comprising catalogues for assets and semantic models, is foundational for compatibility and compliance, stewarding the framing, interpretation, and governance of stored and processed data.

Canonical models for infrastructure operations revolve around improving semantic consistency. Though general-purpose schemas—dimensional (star, snowflake, galaxy) and normalized databases or tables—suffice for the analytical needs of supervisors and managers, operational needs differ. Analysts involved in DISM-R3 operations advocate dedicated models for the source data and variables needed to report accurately. Semantic modelling omits detailed definitions, such as transformation logic, custody rules, data blinds, and data rules, typically housed in metadata catalogues and data governance frameworks. Such integrations foster semantically coherent infrastructures. The infrastructure asset domain covers all physical elements and their support systems that house, regulate, and manage the services consumed or offered. The integration source domain encompasses second-order sources required to manage first-order data integrations. Data source descriptions precisely and unambiguously define each of the primary data sources.

6.3. Intelligent Techniques in Infrastructure Operations

Machine learning and statistics are powerful tools for building intelligent systems, but few algorithms are uniformly useful. Areas such as natural language processing, reinforcement learning, and general game playing have a long history, a large body of literature, and groundbreaking results in the general-purpose capabilities of learned systems. Within the scope of infrastructure operations and engineering, a number of classes of problems have been identified that are amenable to solution by learning-based techniques, and these problems may be modeled using either supervised or unsupervised learning approaches.

Intelligent data engineering is often applied to identify anomalies in sporadic, streaming, or time-series data. Anomaly detection methods, commonly implemented using a supervised learning approach, can facilitate event detection such as cyberattacks,

equipment failure, and sudden environmental changes. Anomaly detection techniques are often integrated with knowledge graphs or other semantic representations, allowing for detection beyond simple pattern matching. Another family of intelligent techniques focuses on system state and risk assessment, including health state monitoring, prognostics, and predictive maintenance. Health state monitoring and prognostics approaches support assessments of the remaining useful life of physical assets and the prediction of future risks.

6.3.1. Anomaly Detection and Prognostics

A three-step framework covers anomaly detection for continuously monitored numeric signals. Current and historical observations are analyzed to mark values as anomalous or not, using methodology appropriate to the domain or the specific data set. Such anomalies serve as an additional signal to focus specialized investigation. For failure prognostics, the objective of prediction can vary—from estimating remaining useful life to recognizing predicted maintenance windows. Supervised (or semi-supervised) feature engineering helps capture the context of historical failures, while unsupervised feature engineering may be useful for shorter-term predictions. In both cases, susceptibility or likelihood of failure is the primary predicted variable, and inference can affect scheduling and resource allocation in operations.

Fig 6.3: Anomaly Detection and Prognostics

Anomaly detection and early-warning systems can furthermore help operational teams perform continuous, live monitoring of the health status of their services. Anomaly-detection algorithms flag real anomalies. The existence of an anomaly can thus trigger alerts, but the value of the signal lies in its role as an additional dimension in an evolving environment. Early warnings risk being seen as a dilution of responsibility, especially when they come from a machine learning-infused data model that has delivered a string

of false positives. Acceptance of the signal requires appropriate parallelism with deeper domain knowledge, experience in anomaly symptom analysis, and confirmation from dedicated teams focused on fault isolation and root-cause analysis.

6.3.2. Predictive Maintenance and Scheduling

Predictive maintenance and scheduling approaches exploit information about imminent component failures to reduce unplanned downtime and optimize maintenance resource allocation and scheduling. The prediction task is concerned with estimating the time remaining until failure (RTF) for a component undergoing a monitored degradation process. Maintenance windows are then defined according to a target effect size or threshold and are ultimately scheduled to minimize maintenance cost, resource usage, and downtime.

Condition monitoring data from different infrastructure assets has been shown to include patterns associated with various types of impending failures, supporting the application of supervised learning approaches for RTF estimation. Available labeled data is, however, usually limited to a single type of failure or domain and is not always representative of the monitored RTF, presenting challenges for supervised techniques. Unsupervised learning methods have therefore been proposed to search for characteristic signatures of remaining useful life in condition data recorded before failure without requiring labeled datasets.

In addition to component-level RTF predictions, failures can also be monitored separately or jointly at a higher system or network level. The resulting maintenance windows can be prioritized based on infrastructure reliability, maintenance cost, and logistics considerations and can be scheduled to minimize overall maintenance expenditure, resource usage, or unplanned downtime.

6.4. Data Platforms and Tooling for Operations

Data engineering for infrastructure relies on a multi-layer technology stack that incorporates special-purpose components but is built primarily on general-purpose systems, some of which are hosted in the cloud. The choice of componentry is driven by the requirements imposed by the data sources and intended use cases. Data ingestion is accomplished with custom connectors that pull from operational systems and common cloud-based services, as well as a mixture of standard open-source tools (Debezium, Kafka Connect) that support push-based capture from database changesets. A streaming platform using Apache Kafka enables real-time streaming analytics and notifications of anomalous conditions, while a data lake built on Apache Iceberg provides a batch-

oriented analytics environment that best supports the most common business intelligence usage patterns. A combination of these patterns—tricking data into being streamed, at least partially, with windowed aggregations or using other data-preparation services such as dbt—are preferred whenever data freshness is required. Proper governance of data lake operations ensures the correctness of the underlying data, regardless of the pattern used to access it.

Atop the integration and storage layer sits an orchestration layer, which brings together ingest connectors, tools for running data-preparation jobs on streaming data (such as dbt), and support for analytics and machine-learning workloads. Such a layer ensures the regular execution of repeated tasks that do not need to execute in real time, together with the requisite monitoring, logging, security, and audit facilities. Given the privacy implications of the assets being monitored, it also enforces strict access-control policies and ensures the correct use of data during development and experimentation by the data-science team.

Fig 6.4: Data Platforms and Tooling for Operations

6.4.1. Streaming Platforms and Data Lakes

A streaming platform is an infrastructure component designed to collect, process, and disperse event data in real time. Event-driven systems leverage such platforms to achieve low-latency interactions, minimizing wait time between request and response. Multiple sources continuously emit event data to the platform, which buffers and organizes these inputs for efficient pattern detection. The identified events trigger automated responses and may also generate enriched data through auxiliary models. Unlike request-response architectures where direct data use is infrequent, data creation in event-driven systems

is continuous. Consequently, event data often serve multiple consumers, resulting in significant logs. To support reliable, scalable, and fault-tolerant operations, appropriate data storage and management are essential. This necessitates a data lake.

Data lakes enable seamless storage and governance of event data, supporting numerous data engineering tasks. However, their lack of governance policies can lead to inefficiency. Organizations must implement suitable governance rules and management policies to enhance the lake's requested usage while preventing unrequested CRUD (Create, Update, Delete) actions. Although access control policies are often granular, alert status tracking is generally coarse, hindering data use for ML model training, business intelligence, and planning analysis. Often overlooked, SLO compliance monitoring in event-driven architectures is costly and difficult; it adapts SLA parameters to consider maintenance periods and deviations in SLO compliance. Event data lakes must therefore support access control based on shared users rather than operation zones.

6.4.2. Orchestration, Governance, and Security

Specific tools and components in a data platform stack serve operations and infrastructure support elements, fulfilling the needed technological functions. Orchestration automates complex workflows for data collection, feature generation, model execution, labeling, and evaluation. Governance assures that the expected data quality is attained and maintained. Data security guarantees the confidentiality of sensitive information and limits access to trusted parties.

Data orchestration tools define and control a variety of data processing pipelines. These are frequently based on reusable building blocks and suitable for the periodic execution of complex workflows. In infrastructure operations, orchestrators integrate data from multiple sources and support models or services that require feature engineering, retraining, or periodic re-evaluation.

Data governance functions make the data platform compliant with predefined policies. Specific tools enable data quality checks on ingestion; control the introduction of sensitive data from open data sources; apply labeling and evaluation functions to monitoring data; and maintain metadata catalogs. Validation processes and procedures for the open-source data sources used help mitigate the risk of introducing errors into the models or services.

Governance capabilities become increasingly important when the use of data analytics is expanded to ad-hoc queries or shared datasets. In these scenarios, data can be used by unauthorized parties for unintended or even malicious purposes. Data governance tools control access to inappropriate data and provide evidence of security compliance for sensitive information.

Security tools monitor and control data access and usage. They log accesses and changes to sensitive data for traceability, check that usage patterns are coherent with data policies, and enforce data policies — for example, by compiling a list of sanctioned and allowed sources for open data.

6.5. Conclusion

A coherent data-engineering strategy for infrastructure operations supports both traditional and machine-learning techniques—both are driven by data and depend on effective ingestion, integration, modeling, and management. A principled approach ensures data quality and enables the discovery of actionable intelligence. The discussion encompasses infrastructure operations in general; particular use cases influence subsequent sections that address intelligent solutions and data-platform design.

The proposals act as a compass for infrastructure operations, shaping data-engineering strategy to support intelligent systems and decision-making. Trade-offs between latency and throughput, together with the differing quality requirements for real-time and batch data, define the orchestration technology stack and patterns. Data provenance equipped with a set of quality dimensions enables a thorough exploration of trust requirements, integrating the data-governance component and outlining the necessary auditing policy framework. Systematic use of quality dimensions permits the definition of quality scorecards and helps identify and eliminate causes of poor-quality data.

References

Nayak, P., Loganathan, R., & Jayaraj, V. (2026, April). Scale-Aware Dilated Lightweight Convolutional Network Improving Solar Panel Defect Classification through Efficient Electroluminescent Image Analysis Techniques. In 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-7). IEEE.

Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández-Moctezuma, R. J., Lax, R., McVeety, S., Mills, D., Perry, F., Schmidt, E., & Whittle, S. (2015). The dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. Proceedings of the VLDB Endowment, 8(12), 1792–1803.

Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., Stoica, I., Wendell, P., Xin, R., & Zaharia, M. (2015). Scaling Spark in the real world: Performance and usability. Proceedings of the VLDB Endowment, 8(12), 1840–1843.

Rao, C. S., Bandi, V. D. V. K., Brahmandam, L. M. K., Gantikota, S., & Kannan, K. (2026, April). Topology-Aware Hypergraph Neural Network Framework for Intelligent Decision Support Systems in Network Operations. In 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-7). IEEE..

Borthakur, D. (2007). The Hadoop distributed file system: Architecture and design. Hadoop Project Documentation, 11, 21.

Padma, G., Reddy, V. A. R., A, S. Devi. K., Buvaneswari, B., & Kumar, K. S. S. (2026). Privacy-Preserved Face Recognition Biometric Authentication Using FaceNet and Zero-Knowledge Proofs for Secure, Access Control on Decentralized Blockchain Networks. In 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1–8). IEEE. 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN). https://doi.org/10.1109/iciscn67954.2026.11566528

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), Article 15.

Rajamanickam, V., Singireddy, S., Davuluri, P. N., Sheelam, G. K., Aitha, A. R., & Vakkalagadda, T. (2026). AI-Enabled Visual Evidence Intelligence for Detecting Manipulated Digital Media. International Journal of Special Education, 41(14s), 115-124.

Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). The Google file system. Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, 29–43.

Jameel, M., & Mangalampalli, B. M. (2026). AI-DRIVEN DESIGN OPTIMIZATION OF SUSTAINABLE STRUCTURAL MATERIALS FOR RESILIENT INFRASTRUCTURE. Advanced Engineering Sciences, 58(1), 2233-2254.

Li, J., Tufte, K., Shkapenyuk, V., Papadimos, V., Johnson, T., & Maier, D. (2008). Out-of-order processing: A new architecture for high-performance stream systems. Proceedings of the VLDB Endowment, 1(1), 274–288.

Kolla, T. (2026). Blockchain for Health Records Management Policy, Legal, and Technical Perspectives. Journal of Health Politics, Policy and Law.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D. B., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M. J., Zadeh, R., Zaharia, M., & Talwalkar, A. (2016). MLlib: Machine learning in Apache Spark. Journal of Machine Learning Research, 17(34), 1–7.

Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30–39.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 28, 2503–2511.

Stonebraker, M., Çetintemel, U., & Zdonik, S. (2005). The 8 requirements of real-time stream processing. ACM SIGMOD Record, 34(4), 42–47.

Raj, S., Prashanthi, P., Kolla, S. K., Bandaru, S., & Bhardwaj, N. (2026, April). Lightweight Cryptographic Schemes for Resource-Constrained IoT and Healthcare Devices. In 2026 International Conference on Multidisciplinary Innovations For Smart & Sustainable Future (MISSF) (pp. 1-6). IEEE.

Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2013). Discretized streams: Fault-tolerant streaming computation at scale. Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, 423–438.

Naveen, K., Lingam, R., Kunduru, G. R., Mangala, N., Reddy, N. N., & Balaji, P. (2026). Intelligent Loan Approval System: Machine Learning for Home and Education Loan

Eligibility. In 2026 Contemporary Computing Innovations Conference (CCIC) (pp. 1–6). IEEE. 2026 Contemporary Computing Innovations Conference (CCIC).