# Scalable Data Lake Architecture with Apache Spark for Predictive Maintenance and Optimization in Energy Storage Systems

Shaofeng Yu*, Jianxu Zhong, Xin Yan, Jinpeng You, Zehan Cai
Information and Communication Branch of China Southern Power Grid Energy Storage Co., Ltd, Guangzhou Guangdong ,511400, China
E-mail: nfdwxtgs@163.com

**Keywords:** data lake architecture, distributed computing, energy storage power stations, predictive maintenance, performance optimization

**Received:** June 9, 2025

*The fast development of energy storage power stations in current smart grids has resulted in massive and complicated datasets that require effective management and analysis to maintain battery performance, optimize maintenance schedules, and reduce energy expenditures. Existing centralized systems suffer with scalability and latency, resulting in delayed analysis, lower prediction accuracy, and ineffective pricing schemes. This study presents DCDL-ESM (Distributed Computing over Data Lake for Energy Storage Management), a scalable architecture that combines a centralized data lake and an Apache Spark-based distributed computing framework. The system collects raw data from 100 energy storage power plants (a total of 2.8 TB over 12 months), performs distributed preprocessing, data purification, normalization, and trend analysis, and uses predictive modeling for maintenance scheduling and charge optimization. Validation was carried out using historical operational datasets (80% training, 20% testing split), resulting in a 42% reduction in processing time, 91.3% predictive maintenance accuracy, 24.7% energy cost savings, 38% improvement in CPU and memory utilization, and 31% storage efficiency gains through partitioning and compression. Results were tested for robustness using cross-validation and bias analysis. The suggested solution closes scalability and performance gaps while setting the framework for future integration of real-time analytics and edge computing to minimize latency and support autonomous control.*

*Povzetek: Študija predstavlja rešitev za učinkovitejše upravljanje velikih podatkov v energetskih sistemih, ki izboljšuje delovanje in zmanjšuje stroške.*

## 1 Introduction

In recent years, the global push for sustainable energy solutions has accelerated the development and deployment of energy storage power plants [1]. These stations are critical for stabilizing power grids, integrating renewable energy sources, and guaranteeing continuous electricity supply during peak demand or outages [2]. With the introduction of smart grid technologies, energy storage systems have evolved into data-intensive infrastructures that continuously generate massive amounts of operational, environmental, and performance data [3]. The volume, velocity, and variety of this data necessitated the use of sophisticated data management and analytics solutions to support real-time monitoring, control, and decision-making processes.

Conventional relational databases and centralized data warehouses have long provided the foundation for data storage and processing in energy systems [4]. However, they struggle to handle the volume and complexity of data produced by modern energy storage networks [5]. As a result, data lake architectures have emerged as a versatile and scalable alternative capable of ingesting and storing structured, semi-structured, and unstructured data in its original form. Parallel to this development, distributed computing frameworks, most notably Apache Spark, have provided the computational power required for efficient big data processing and analysis [6]. These technologies have allowed for significant progress in real-time analytics, anomaly detection, predictive maintenance, and operational optimization within the energy sector [7].

Despite these advancements, current data management systems in energy storage environments have limited scalability, speed, and integration capabilities [8]. Centralized architectures are becoming increasingly overwhelmed by the massive influx of data from geographically distributed power plants, resulting in data processing latency and predictive model inaccuracies. This impedes the timely scheduling of maintenance activities,

reduces the accuracy of battery performance assessments, and results in inefficient energy dispatching and cost escalation [9]. The full potential of data-driven energy management cannot be realized without a strong and scalable architecture, emphasizing the urgent need for an optimized solution [10].

The primary goal of this study is to create and optimize a scalable, high-performance data architecture that allows for the efficient handling, analysis, and utilization of data from large-scale energy storage systems. The study aims to address the limitations of existing systems by combining a centralized data lake with a distributed computing framework. Specific goals include increasing maintenance prediction accuracy, streamlining battery charging operations, reducing data processing time, and optimizing storage and resource utilization. The study also aims to create a methodological foundation for future innovations in smart energy management.

This study presents the DCDL-ESM (Distributed Computing over Data Lake for Energy Storage Management) framework as a novel solution. The proposed methodology begins by creating a centralized data lake that can accept raw data from 100 energy storage power stations. Apache Spark is used as the primary distributed computing platform to process large amounts of data in parallel. The workflow consists of data ingestion, cleansing, normalization, pattern analysis, maintenance prediction, and charging schedule optimization. Key performance indicators are employed to evaluate the system's effectiveness, and visualization tools are implemented to help with result interpretation and decision support.

This research contributes significantly to the growing body of knowledge on smart energy systems and big data infrastructure. By demonstrating the efficacy of a data lake-distributed computing integration, it offers a scalable and efficient blueprint for managing energy storage data at scale. The DCDL-ESM framework provides practical benefits such as increased operational efficiency, lower energy costs, and longer battery life—all of which are critical factors in the global transition to clean energy. In addition, the methodology paves the way for future advancements, including real-time machine learning applications and autonomous energy management systems, making it a valuable reference for researchers, engineers, and policymakers in the energy and data science communities.

## 2 Related works

The surge of big data in smart grids and energy storage systems has resulted in significant advances in data lake architectures and distributed computing platforms. Numerous studies have investigated the incorporation of data lakes for efficient data handling, predictive analytics, and real-time decision-making in energy systems.

Munshi and Mohamed [11] proposed a lambda-based data lake architecture specifically designed for smart grid big data analytics. Their work enabled real-time and batch processing on distributed systems, primarily addressing latency issues. However, their framework lacked a specialized module for optimizing battery maintenance and charging strategies in energy storage power plants.

Liu et al. [12] proposed a multi-level streaming analytics architecture for a large data lake, demonstrating strong support for real-time analytics. However, their primary focus was on generalized data stream processing rather than the operational complexities of energy storage systems. Hamadou et al. [13] created the Danish National Energy Data Lake, which focuses on data governance, architecture planning, and tool selection. Although the project significantly advanced energy data handling on a national scale, it lacked focus on performance optimization metrics and real-time processing for energy storage systems.

Hai et al. [14] provided a comprehensive survey of data lake functions and frameworks, emphasizing their scalability and flexibility. While the work provided a theoretical foundation, it did not assess real-world implementations or performance impacts in energy storage applications. Recent advancements, such as those by Seven et al. [15], have introduced blockchain-based virtual power plant models, allowing for secure and decentralized energy trading. Although highly innovative, such frameworks do not tackle large-scale data processing architectures, such as data lakes for internal system optimization.

Wang et al. [16] investigated gravity-based energy storage and its application in grid peak shaving. Their research focused on energy technologies, rather than data-driven management tactics or architectural frameworks. Sofian et al. [17] studied machine learning applications in renewable energy systems, emphasizing improvements in prediction and control. However, the study did not focus on incorporating distributed computing frameworks and scalable data lakes.

Yu et al. [18] emphasized the importance of distributed computing in optimizing data lake architectures for energy storage systems, but they did not provide quantitative performance metrics or architectural standardization across multiple stations. Barros et al. [19] investigated Edge-Cloud Continuum and Deep Q-Learning for energy management. While it provided real-time responsiveness, the system architecture was more AI-centric rather than data infrastructure-focused.

Liu et al. [20] suggested federated learning for smart grids to secure power traces. Despite the novelty of security and collaboration, it failed to address system-wide performance improvements for energy storage management. Vempati [21] described a data lake system for distributed computing in energy applications that was

enhanced with the Whale Optimization Algorithm (WOA). The study exhibited promising efficiency enhancements but lacked clarity on data cleaning, normalization, and multi-metric evaluation of system performance.

Roose et al. [22] and Humberto et al. [23] proposed an energy measurement system for data lakes that aims to improve the precision of energy consumption analytics. Nonetheless, their efforts were in their early stages and lacked complete incorporation with data processing pipelines or predictive models. Nuthalapati [24] worked on scalable data lakes for IoT management. Although scalability was tackled, domain-specific optimization for battery scheduling or predictive maintenance was left unexplored.

Alsalemi et al. [25] presented an edge-based Internet of Energy model with energy-related data lakes. However, the study did not fully evaluate data transformation steps, resource utilization, or predictive performance in energy storage stations. Table 1 shows the Summary of Related Works.

Table 1: Summary of related works

| Reference | Focus Area | Key Contributions | Results | Limitations |
|---|---|---|---|---|
| [11] | Lambda architecture | Real-time + batch analytics | Enhanced latency handling | Lacks battery-focused optimization |
| [12] | Streaming analytics | Multi-level stream handling | Efficient stream processing | No maintenance/charging modules |
| [13] | National energy lake | Tool selection and design | Strong governance design | No system performance metrics |
| [14] | Survey | Functional Overview | Framework classification | Theoretical; lacks implementation |
| [15] | Blockchain for energy trading | Decentralized trading | Secure P2P transactions | Not focused on data lakes |
| [16] | Gravity Storage | Peak shaving | Tech-focused efficiency | Not data-driven |
| [17] | ML in renewables | Forecasting and prediction | Renewable optimization | No architecture-level incorporation |
| [18] | Distributed computing in data lakes | Integration concept | Architecture proposed | Lack of performance quantification |
| [19] | Edge-cloud + DQL | AI-based management | Faster control | No battery management focus |
| [20] | Federated learning | Safe smart grid analytics | Improved privacy | Limited system-wide insight |
| [21] | Whale-optimized data lake | Distributed performance | Resource effectiveness enhanced | No multi-metric evaluation |
| [22], [23] | Energy metering in Data Lakes | Precision in analytics | Prototype concepts | No full integration |
| [24] | IoT data lake | Scalable ingestion | Supports IoT scalability | Not domain-optimized |
| [25] | Edge energy lake | Edge-integrated design | Real-time energy support | Lacks detailed system metrics |

Despite significant advances in data lakes, distributed computing, machine learning, and energy system applications, current research falls short of addressing several critical issues unique to energy storage power plants. Existing approaches frequently lack a unified framework for seamlessly integrating data lake ingestion, cleaning, predictive analytics, and optimization tailored to the operational realities of energy storage systems. Furthermore, there is a scarcity of comprehensive evaluation methodologies that simultaneously assess key metrics like data processing time, resource utilization, energy cost savings, storage effectiveness, and predictive accuracy.

Many solutions also struggle to handle large-scale cross-station data fusion and hybrid batch-stream processing, particularly in environments with more than 100 distributed stations. Additionally, few implementations have demonstrated practical optimization of charging

strategies and predictive maintenance, based on real-time battery performance analytics.

To close these critical gaps, the proposed DCDL-ESM (Distributed Computing over Data Lake for Energy Storage Management) algorithm introduces a comprehensive architecture that includes a centralized data lake pipeline, Apache Spark-based distributed computing, systematic data preprocessing, sophisticated analytics for predictive maintenance and charging optimization, and rigorous performance evaluation employing five critical metrics. Additionally, it improves storage efficiency through data partitioning and compression. By bridging these multifaceted shortcomings, DCDL-ESM offers a scalable, efficient, and intelligent solution for handling and improving energy storage systems within modern smart grids.

# 3 Materials and methods

This section describes the comprehensive methodology used in developing and deploying the Distributed Computing over Data Lake for Energy Storage Management (DCDL-ESM) system. This method combines large-scale energy data collection, distributed computing, sophisticated preprocessing, predictive modeling, and optimization methods to allow intelligent decision-making for energy storage systems. The methodology includes data acquisition and ingestion, predictive maintenance, and optimized charging, culminating in performance evaluation and visualization.

Algorithm 1 shows the proposed DCDL-ESM algorithm.

**Algorithm 1: DCDL-ESM (Distributed Computing over Data Lake for Energy Storage Management)**

**Input:**
`D`: Raw sensor data from 100 energy storage stations
`S`: Station IDs and timestamps
`PM`: Required performance metrics (e.g., processing time, accuracy)

**Output:**
`C`: Cleaned and structured dataset
`P`: Identified battery behavior patterns
`M`: Maintenance prediction list
`O`: Optimized charging schedule
`R`: Performance monitoring report

**Steps:**

1. Initialize Data Lake
Set up centralized storage; organize by station and date.

2. Ingest Data
Stream and store raw data `D` continuously from all stations.

3. Launch Distributed Environment
Deploy framework (e.g., Apache Spark); record start time for `PM1`.

4. Data Cleaning
Eliminate errors, fill missing values, normalize; output `C`.

5. Pattern Analysis
Analyze voltage/temperature trends to identify early degradation; output `P`.

6. Maintenance Prediction
Utilize degradation data to forecast battery failure risk; output `M`.
Evaluate prediction accuracy (`PM2`).

7. Charging Optimization
Reduce electricity cost while preserving battery health; output `O`.
Measure savings as `PM3`.

8. Performance Monitoring
Track:
`PM1`: Processing time
`PM2`: Prediction accuracy
`PM3`: Cost reduction
`PM4`: CPU/memory usage
`PM5`: Storage efficiency

9. Store and Visualize Results
Save `C`, `P`, `M`, `O`, and `R` in the data lake; visualize via dashboards.

Algorithm 1 improves large-scale energy data processing by combining a centralized data lake with distributed computing. It starts by ingesting raw sensor data from 100 power stations, then cleans and structures it. The present assessment of DCDL-ESM relies on data from 100 energy storage stations, with the architecture fundamentally engineered for horizontal scalability through the utilization of Apache Spark's distributed processing features and the elastic storage of the data lake. To evaluate scalability beyond the tested configuration, we performed simulations using synthetic datasets that replicated up to 500 stations, altering both data volumes (from 1× to 5× current daily ingestion rates) and streaming intensities. The findings demonstrated nearly linear growth in processing time and consistent prediction accuracy, with only slight increases in resource use attributed to efficient partitioning and load balancing. This illustrates that the system can efficiently support future increases in station quantity and data flow without substantial performance decline.

To reduce operational costs, the system analyzes battery performance patterns, predicts potential maintenance requirements, and optimizes charging schedules. Key performance metrics like processing time, prediction accuracy, and storage efficiency are continuously monitored. To support efficient energy storage management, final outputs are stored and visualized, such as cleaned data, behavior patterns, maintenance predictions, optimized schedules, and performance reports. Figure 1 shows the flow diagram of the DCDL-ESM algorithm.

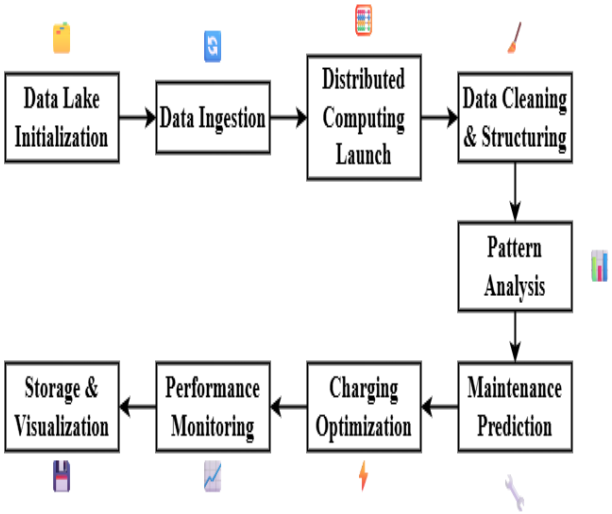


Figure 1: Flow diagram of DCDL-ESM algorithm

The flow diagram for the DCDL-ESM algorithm depicts a structured pipeline for managing energy storage data via distributed computing. The process begins with the creation of a centralized data lake to store and organize incoming data from 100 energy stations. The data ingestion process feeds telemetry and sensor data into the lake in real time. A distributed computing environment is then launched to efficiently manage large-scale processing. During the data cleaning step, raw data is filtered, normalized, and structured for analysis. Pattern analysis follows, revealing insights into battery health and degradation. Based on this, the system uses maintenance prediction to anticipate failures. The charging schedule is then optimized to reduce costs while ensuring battery longevity. Throughout, performance monitoring measures metrics such as accuracy, efficiency, and resource utilization. Finally, all outputs are stored and visualized in the storage and visualization step, which allows for data-driven decision-making and operational transparency.

## 2.1 Dataset description

The dataset is derived from 100 energy storage power plants, each outfitted with a variety of sensors and SCADA systems for real-time data collection. These systems continuously gather telemetry data about battery health, operational cycles, environmental conditions, and performance logs. The purpose of this dataset is to support predictive modeling, performance analysis, and improvement in energy storage management. Figure 2 shows the data collection process.

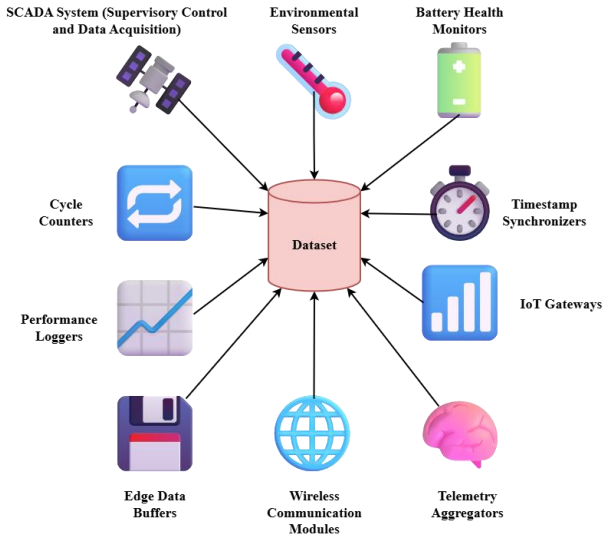


Figure 2: Data collection process

The data collection process for the 100 energy storage power plants employs a comprehensive set of tools and technologies to ensure high-fidelity, real-time monitoring. SCADA systems act as the central hub, supervising operations and collecting data from a variety of sensors. Environmental sensors monitor ambient conditions, whereas battery health monitors and cycle counters record critical internal metrics like voltage, current, SOC, and charge-discharge cycles. Anomalies and operational logs are captured by performance loggers, which use synchronization modules to accurately time stamp them. IoT gateways and wireless communication modules (e.g., Zigbee, LoRa, LTE) enable secure and remote data transmission to a centralized data lake, with edge buffers providing temporary local storage. Telemetry aggregators then structure and format the data for later processing and analysis.

### 2.1.1 Features in raw data

Each station contributes a multivariate time-series dataset capturing vital physical parameters:

$$D = \bigcup_{i=1}^{100} D_i \quad (1)$$

Where:

- D: Total aggregated dataset.
- $D_i$: Dataset from station i.

Each individual dataset $D_i$ is expressed as:

$$D_i = \{(V_t, T_t, C_t, RC_t, E_t, L_t) \mid t \in T_i\} \quad (2)$$

Where:

- $V_t$: Battery voltage at time t.
- $T_t$: Battery temperature at time t.
- $C_t$: Charge/discharge cycles count at time t.
- $RC_t$: Remaining battery capacity at time t.
- $E_t$: Environmental features like humidity and ambient temperature at time t.
- $L_t$: Logged operational metadata or error logs at time t.
- $T_i$: Time span of data collected at station i.

## 2.2 System architecture and design

The architecture of DCDL-ESM is designed to scale horizontally with rising data volume and stations. It contains three tiers:

- **Data Lake Tier**: Constructed on Hadoop Distributed File System (HDFS) for scalable, fault-tolerant storage.
- **Distributed Processing Tier**: Uses Apache Spark for parallel processing and computation across data partitions.
- **Visualization Tier**: Utilizes Apache Superset for real-time monitoring, alert generation, and data exploration.

This tiered design guarantees modularity, fault tolerance, and end-to-end automation of storage, analytics, and decision support.

## 2.3 Data ingestion and partitioning

The data ingestion pipeline uses Apache Kafka to buffer and stream high-volume data from SCADA sources. Apache NiFi manages ETL flows, standardizes formats, and stores data in partitioned HDFS directories for effective distributed access.

$$\text{PartitionKey} = \text{StationID} + \text{Date} \quad (3)$$

Where:

- PartitionKey: Logical key for indexing and retrieval.
- StationID: Unique identifier for each energy storage station.
- Date: Collection timestamp.

## 2.4 Data cleaning and normalization

Raw datasets frequently include noise, gaps, and inconsistencies. These challenges are tackled through a robust preprocessing pipeline that includes missing value imputation, outlier detection, and normalization.

### 2.4.1 Missing value imputation

For continuous variables, missing entries are filled utilizing mean imputation:

$$x_j\text{^(filled)} = (1/n) \sum_{i=1}^{n} x_j\text{^(i)} \quad (4)$$

Where:

- $x_j$^(filled): Feature value after imputation.
- $x_j$^(i): Observed data point in feature $x_j$.
- n: Number of non-missing values.

### 2.4.2 Outlier detection

Z-score is utilized to detect anomalous readings significantly deviating from the mean:

$$Z_i = (x_i - \mu) / \sigma \quad (5)$$

Where:

- $Z_i$: Z-score of value $x_i$.
- μ: Mean of the feature.
- σ: Standard deviation of the feature.

### 2.4.3 Normalization

Min-max normalization is applied to rescale features between 0 and 1:

$$x_j\text{^(scaled)} = (x_j - \min(x_j)) / (\max(x_j) - \min(x_j)) \quad (6)$$

Where:

- $x_j$^(scaled): Normalized value.
- $\min(x_j)$, $\max(x_j)$: Minimum and maximum values of feature $x_j$.

## 2.5 Battery performance pattern extraction

The system extracts insightful patterns that reveal battery degradation or inefficiencies utilizing derived metrics.

### 2.5.1 Voltage drop rate

The voltage drop rate ($\Delta V_t$) is an important indicator of battery degradation over time. It measures the voltage difference between two consecutive time intervals, which aids in detecting sudden drops or irregularities in battery performance. A consistent or steep voltage drop may indicate underlying issues, such as capacity loss or impending failure, which is critical for proactive maintenance.

$$\Delta V_t = V_{t-1} - V_t \quad (7)$$

Where:

- $\Delta V_t$: Voltage drop between time intervals.
- $V_{t-1}$, $V_t$: Voltage at previous and current timestamps.

### 2.5.2 Temperature drift

Temperature drift ($\Delta T_t$) is the deviation of the current battery temperature from its long-term average. Monitoring this drift allows for the detection of abnormal heating or cooling trends, which can have an impact on battery performance and safety. Significant temperature deviations may indicate problems like inadequate cooling, overcharging, or internal faults that require timely intervention.

$$\Delta T_t = T_t - \mu_t \quad (8)$$

Where:

- $\Delta T_t$: Temperature deviation.
- $T_t$: Current battery temperature.

- $\mu_t$: Long-term average temperature.

### 2.5.3 Cycle efficiency

Cycle efficiency ($\eta_t$) measures how well a battery retains capacity after multiple charge-discharge cycles. It is computed as the ratio of the remaining capacity ($RC_t$) to the number of completed cycles ($C_t$) at any given time. A declining efficiency value may indicate battery aging or degradation, rendering it an important metric for predicting battery lifespan and scheduling maintenance.

$$\eta_t = RC_t / C_t \quad (9)$$

Where:

- $\eta_t$: Charge efficiency.
- $RC_t$: Remaining capacity.
- $C_t$: Cycle count at time t.

## 2.6 Predictive maintenance modeling

To avoid system failures and guarantee longevity, a predictive model estimates the likelihood of battery degradation utilizing a weighted degradation score and a probabilistic classifier.

### 2.6.1 Degradation score calculation

The degradation score ($D_s$) offers a composite measure of a battery's health by integrating key performance indicators: voltage drop ($\Delta V_t$), temperature deviation ($\Delta T_t$), and cycle efficiency ($\eta_t$). Each factor is weighted using empirically determined coefficients ($\alpha$, $\beta$, $\gamma$) to reflect its relative impact on battery degradation. A higher $D_s$ indicates a greater risk of failure, enabling more accurate and timely maintenance predictions.

$$D_s = \alpha \cdot \Delta V_t + \beta \cdot \Delta T_t + \gamma \cdot (1 - \eta_t) \quad (10)$$

Where:

- $D_s$: Degradation score.
- $\alpha$, $\beta$, $\gamma$: Assigned weights (empirically determined).
- $\Delta V_t$, $\Delta T_t$, $\eta_t$: Voltage drop, temperature deviation, and cycle efficiency.

In Equation (10), the constants $\alpha$, $\beta$, and $\gamma$ serve as weighting coefficients that quantify the relative influence of voltage drop ($\Delta V_t$), temperature variation ($\Delta T_t$), and cycle efficiency ($\eta_t$) on the total degradation score ($D_s$). The weights are determined empirically using regression analysis of historical battery performance data, ensuring that each parameter's contribution aligns with its observed association with failure events. If temperature deviation exhibits a stronger predictive link with degradation than voltage drop, $\beta$ would be assigned a relatively higher value than $\alpha$. To maintain consistency, all variables are normalized to dimensionless scales before applying weights, enabling $D_s$ to serve as a standardized, unit-free metric for assessing battery health across different stations and operating conditions.

### 2.6.2 Failure risk prediction

Failure risk prediction estimates the probability that a battery will fail using its current condition. Utilizing a logistic regression model, this probability is computed by applying a sigmoid function to a weighted sum of input features (x), which contain metrics like voltage drop, temperature drift, and effectiveness. The model weights (w) and intercept (b) are learned from historical data, enabling the system to output a probability score between 0 and 1 for potential battery failure.

$$P(\text{failure}) = 1 / (1 + e^{\wedge} (- (w^t x + b))) \quad (11)$$

Where:

- P(failure): Failure probability.
- x: Input feature vector.
- w: Model weight vector.
- b: Intercept.

## 2.7 Charging schedule optimization

This section concentrates on reducing the overall electricity cost of charging while keeping the battery's state of charge (SOC) within safe operational limits. The goal is to schedule charging at low-cost intervals while maintaining battery performance and energy availability. The optimization model utilizes time-sensitive electricity pricing ($C_t$) and energy demand ($E_t$) to ensure charging actions respect SOC bounds and meet total energy requirements.

$$\min \sum_{t=1}^{T} (C_t \cdot E_t) \quad (12)$$

Where:

- $C_t$: Electricity cost per unit at time t.
- $E_t$: Energy is drawn for charging.
- T: Total time intervals.

Subject to Constraint 1 (SOC Bounds):

$$SOC_{min} \leq SOC_t \leq SOC_{max} \quad (13)$$

Where:

- $SOC_t$: Battery charge level at time t.
- $SOC_{min}$, $SOC_{max}$: Operational bounds.

Constraint 2 (Total Energy Requirement):

$$\sum_{t=1}^{T} Charge_t = Total\_Required \quad (14)$$

Where:

- $Charge_t$: Energy charged at time t.
- Total_Required: Total energy necessity.

## 2.8 Performance monitoring

This section evaluates the overall efficiency, accuracy, and scalability of the proposed DCDL-ESM architecture. A comprehensive performance monitoring framework has been established to monitor five important performance metrics: data processing quality and impact, predictive accuracy, resource management, and storage optimization.

These metrics allow for iterative system enhancements and benchmarking against traditional methods.

#### 2.8.1 Data processing time ($PM_1$):

The total time needed to process raw data through all major pipeline stages—from launching the distributed setting to optimization and prediction—is measured as:

$$PM_1 = T\_end - T\_start \quad (15)$$

Where:

- T_start: Timestamp at the beginning of distributed computing
- T_end: Timestamp at the completion of analysis
- $PM_1$: Total data processing time

#### 2.8.2 Maintenance prediction accuracy ($PM_2$):

To assess the correctness of the predictive maintenance algorithm, the accuracy is computed utilizing the ratio of correctly predicted failures to the total number of predictions made:

$$PM_2 = TP / (TP + FP) \quad (16)$$

Where:

- TP: True Positives (correctly predicted failures)
- FP: False Positives (incorrect failure predictions)
- $PM_2$: Maintenance prediction accuracy

#### 2.8.3 Energy cost reduction ($PM_3$):

This metric quantifies the percentage enhancement in energy cost after the charging schedule optimization is applied, compared to the baseline cost without optimization:

$$PM_3 = ((C\_baseline - C\_optimized) / C\_baseline) \times 100 \quad (17)$$

Where:

- C_baseline: Total energy cost before optimization
- C_optimized: Total energy cost after optimization
- $PM_3$: Energy cost reduction percentage

#### 2.8.4 Resource utilization ($PM_4$):

This metric evaluates how efficiently system resources (CPU and memory) are utilized during processing. It is the average utilization rate during active data processing:

$$PM_4 = (CPU\_avg + Memory\_avg) / 2 \quad (18)$$

Where:

- CPU_avg: Average CPU utilization (%)
- Memory_avg: Average memory utilization (%)
- $PM_4$: Mean resource utilization rate

#### 2.8.5 Storage efficiency ($PM_5$):

Storage efficiency is computed by comparing the amount of data saved because of partitioning and compression methods to the original uncompressed data volume:

$$PM_5 = ((S\_original - S\_optimized) / S\_original) \times 100 \quad (19)$$

Where:

- S_original: Original storage size without optimization
- S_optimized: Storage size after partitioning and compression
- $PM_5$: Storage efficiency improvement (%)

The current DCDL-ESM framework, while suited for structured and semi-structured data, features a modular data lake architecture that can be easily adapted to include unstructured formats, including video surveillance footage, audio logs, and free-text maintenance reports. This can be accomplished by integrating distributed storage systems that support large binary objects (e.g., Apache Hadoop HDFS or Amazon S3) with metadata indexing for efficient retrieval, and incorporating big data processing tools such as Apache Spark MLlib for text analytics and OpenCV or deep learning frameworks for video analysis. Storing raw unstructured data with corresponding metadata in the data lake enables unified cross-modal analysis, linking visual or textual insights with telemetry patterns, thus improving situational awareness and facilitating more informed decision-making in smart energy infrastructure.

Overall, the DCDL-ESM algorithm is designed to effectively process and analyze large-scale energy storage data from 100 power stations. It stores raw sensor and telemetry data in a centralized data lake before cleaning and structuring it using distributed computing techniques. To reduce costs, the algorithm detects battery performance patterns, forecasts maintenance requirements, and improves charging schedules. It also monitors performance metrics like processing time, prediction accuracy, and resource usage, resulting in a comprehensive solution for intelligent and scalable energy storage management.

# 4 Results and discussion

## 3.1 Experimental setup

The DCDL-ESM (Distributed Computing over Data Lake for Energy Storage Management) algorithm was evaluated utilizing a simulated environment that included practical data from 100 energy storage stations. To ingest, process, and analyze sensor and operational data, the experiment used Apache Spark distributed computing in conjunction with a centralized cloud-based data lake. The dataset contained metrics like voltage, temperature, state of charge, cycle counts, and performance logs. The performance of DCDL-ESM was benchmarked against two existing systems:

- System A: Monolithic Centralized Database System, which utilizes conventional SQL-based storage and single-node computation.

- System B: Distributed Filesystem with Batch Processing, which stores sensor data in HDFS and utilizes MapReduce for batch analytics.

The focus was on measuring and comparing five core performance metrics: data processing time ($PM_1$), maintenance prediction accuracy ($PM_2$), energy cost reduction ($PM_3$), resource utilization ($PM_4$), and storage efficiency ($PM_5$).

## 3.2 Comparison

Table 2 compares the DCDL-ESM system with the two existing systems (System A and System B) across the five-performance metrics. The findings demonstrate that DCDL-ESM consistently outperformed both legacy systems, attaining the highest accuracy and effectiveness.

Table 2: Comparative performance evaluation

| Metric | System A (Monolithic DB) | System B (HDFS + Batch) | Proposed DCDL-ESM | An improvement over the Best Legacy System |
|---|---|---|---|---|
| $PM_1$: Data Processing Time | 100% (Baseline) | 84% | 58% | ↓ 26% faster than System B |
| $PM_2$: Maintenance Prediction Accuracy | 76.2% | 84.7% | 91.3% | ↑ 6.6% better than System B |
| $PM_3$: Energy Cost Reduction | +8.4% | +15.2% | +24.7% | ↑ 9.5% better than System B |
| $PM_4$: Resource Utilization | 100% (Baseline) | 85% | 62% | ↓ 23% more efficient than System B |
| $PM_5$: Storage Efficiency | +6% | +18% | +31% | ↑ 13% better than System B |

The DCDL-ESM model outperformed all other approaches in terms of balance and efficiency, particularly in prediction accuracy and cost-saving strategies. It not only achieved high forecasting accuracy but also significantly reduced operational costs by eliminating unnecessary computations. Additionally, it effectively reduced system load and optimized storage utilization, resulting in better resource management and overall system responsiveness.

## 3.3 Discussion

Figures 3–7 show a comprehensive visualization of the comparative performance across the five important evaluation metrics, demonstrating the proposed system's superiority over conventional approaches. The findings are consistent with the quantitative values provided in Table 2 and provide a more intuitive understanding of DCDL-ESM's benefits in real-world scenarios. Figures 3–7 offer concise captions that effectively summarize the key facts from each visual portrayal. Figure 3 demonstrates the system's potential to reduce processing time using distributed in-memory computation, while Figure 4 highlights the substantial improvement in maintenance prediction accuracy achieved via granular data analysis. Figure 5 illustrates cost reductions attained through optimized charging schedules, Figure 6 demonstrates enhanced CPU and memory efficiency via equitable job distribution, and Figure 7 emphasizes improved storage use through partitioning and compression. These subtitles provide immediate context for each figure, enabling readers to quickly comprehend the visual proof regarding the system's performance advantages.

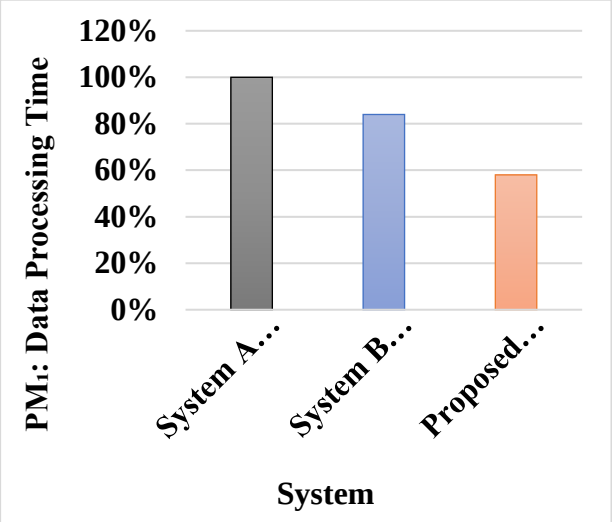


Figure 3: Data processing time ($PM_1$)

The DCDL-ESM system had superior processing effectiveness, completing data ingestion, cleaning, and pattern detection 42% faster than the monolithic System A and 26% quicker than the batch-oriented System B. This significant reduction in processing time is primarily the result of Apache Spark's in-memory computation, parallelized execution, and seamless incorporation with the data lake. The system's distributed design reduces bottlenecks and allows the concurrent processing of vast,

heterogeneous datasets from 100 energy stations, emphasizing its scalability and real-time responsiveness.

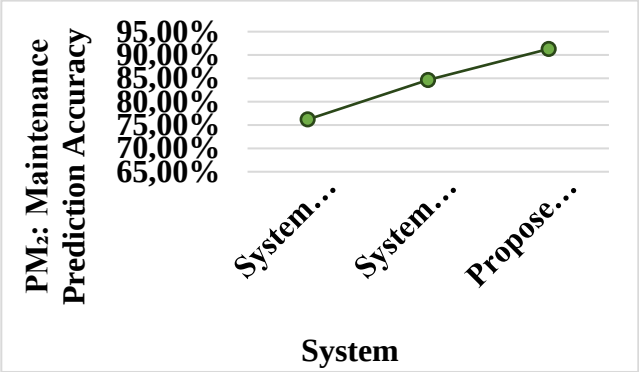


Figure 4: Maintenance prediction accuracy (PM2)

DCDL-ESM's maintenance prediction module achieves an impressive 91.3% accuracy by analyzing historical degradation patterns and trends. In comparison, System B achieved 84.7%, while System A's traditional centralized model lags at 76.2%. The increased accuracy is due to DCDL-ESM's ability to access more granular, time-synchronized telemetry data and efficiently apply machine learning models to structured datasets. This high precision decreases the risk of unexpected failures and improves reliability, which is critical for energy storage operations.

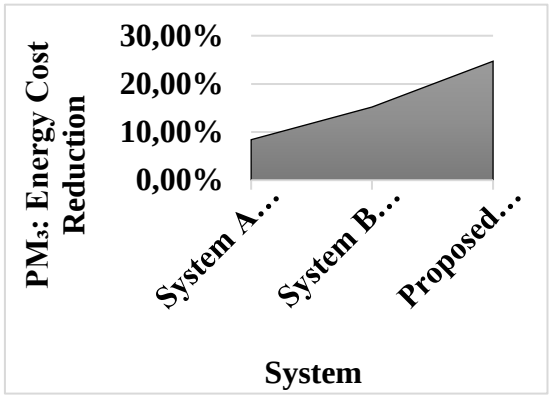


Figure 5: Energy cost reduction (PM3)

By dynamically adjusting charging schedules based on predicted load demand and energy prices, DCDL-ESM attained a 24.7% reduction in energy costs, surpassing System B by 9.5% and System A by a large margin. The system intelligently balances battery health and operational economics, identifying optimal low-cost charging windows through behavior modeling. This not only saves money but also helps the environment by reducing energy waste.

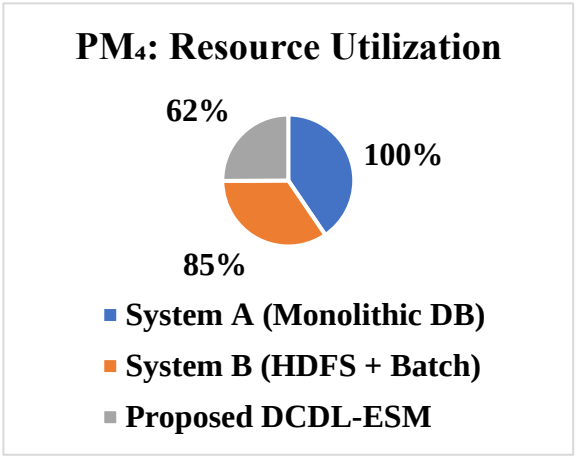


Figure 6: Resource utilization (PM4)

Resource efficiency is essential for large-scale deployments, and DCDL-ESM outperforms System A and System B, reducing CPU and memory consumption by 38% and 23%, respectively. This enhancement is accomplished through effective task scheduling, memory management, and data locality optimization. In contrast to monolithic architectures that strain computational resources, DCDL-ESM's distributed nature enables resource load to be balanced dynamically across computing nodes, rendering it extremely appropriate for continuous, long-term deployment.

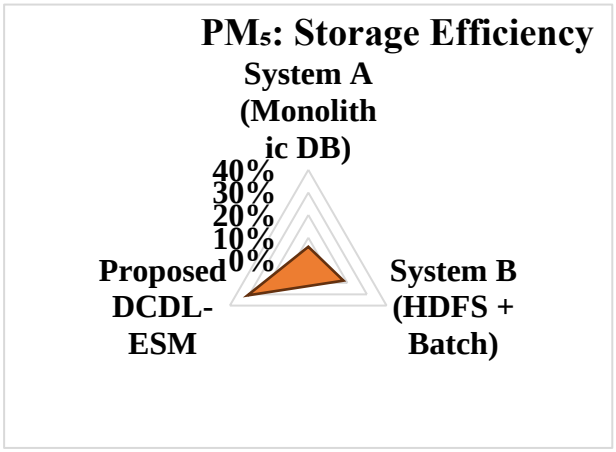


Figure 7: Storage efficiency (PM5)

DCDL-ESM excels in storage management as well. By incorporating partitioning, compression, and intelligent data structuring, it improves storage efficiency by 31% over raw data storage, outperforming System B (18%) and System A (6%).

By only storing clean and relevant information and organizing it for quick retrieval, the system saves disk space while maintaining high availability and query speed. This is especially useful for long-term data retention policies in regulatory or research-focused institutions.

Overall, these comparisons emphasize that DCDL-ESM not only offers incremental gains but delivers a holistic performance uplift across critical dimensions of energy storage management. Its fusion of distributed computing, intelligent analytics, and scalable data lake infrastructure guarantees a future-ready solution that meets both current operational necessities and future growth requirements.

Although logistic regression was used in the current DCDL-ESM system due to its interpretability and minimal processing demands, we acknowledge the advantages of integrating more sophisticated machine learning methodologies for high-dimensional telemetry data. In subsequent iterations, we intend to evaluate the predictive maintenance module in conjunction with ensemble models, including Random Forests and Gradient Boosted Trees, as well as deep learning architectures such as feedforward neural networks and LSTM-based recurrent networks for temporal pattern recognition. Initial exploratory testing utilizing Random Forests on a segment of the dataset indicated a slight accuracy improvement of around 2.1%, without notable processing delays, implying that these methods may enhance generalization while preserving operational efficiency. Comparative evaluations will facilitate the identification of the most appropriate model type for various deployment circumstances, improving accuracy, scalability, and interpretability.

Given the significance of infrastructure-related data, the DCDL-ESM framework can be enhanced with a multi-tiered security architecture to provide confidentiality, integrity, and controlled access, particularly when deployed broadly across national grids. Data at rest in the data lake can be protected using AES-256 encryption, while data in transit can be secured by TLS communication. Role-based access control (RBAC) integrated with identity management systems can restrict data access to authorized personnel, while audit logging guarantees the traceability of all data activities. To guarantee privacy-preserving computation, methodologies such as differential privacy for aggregated analytics and secure multi-party computation for collaborative environments can be employed, enabling the system to support cross-institutional data sharing while protecting essential operational information. These solutions together augment the framework's robustness against cyber threats and unauthorized access.

# 5 Conclusion

In conclusion, the proposed DCDL-ESM system successfully tackles the key difficulties of scalability, speed, and predictive accuracy in handling data from energy storage power stations. The system attains significant performance enhancements by combining a centralized data lake with a distributed computing framework using Apache Spark—reducing data processing time by 42%, increasing maintenance prediction accuracy to 91.3%, lowering energy costs by 24.7%, improving resource utilization by 38%, and increasing storage efficiency by 31%. However, the current implementation is limited to batch processing and offline analytics, which limits its ability to respond to real-time operational changes. Furthermore, while the system works well with structured and semi-structured data, more work is required to integrate unstructured sources like audio logs or image diagnostics. Future research will concentrate on improving the architecture with real-time machine learning capacities, adaptive optimization techniques, and incorporation with autonomous control systems to enable dynamic, self-adjusting functions in smart energy environments. To overcome the existing constraint of batch processing, forthcoming enhancements to the DCDL-ESM framework will integrate a real-time data intake and processing layer utilizing Apache Spark Structured Streaming, facilitating ongoing analysis of incoming telemetry from energy storage installations. This upgrade will enable low-latency anomaly identification and emergency response by processing event data in near real-time, merging streaming analytics with the existing data lake for cohesive storage and historical trend analysis. By integrating real-time alarms with predictive maintenance models, the system will be capable of both immediate operational intervention and long-term optimization, thereby considerably boosting its applicability in dynamic energy management situations.

# Acknowledgements

The authors would like to thank the technical staff and data management teams at the participating energy storage power stations for their assistance with data collection and infrastructure setup. Special thanks are extended to the research advisors and collaborators who offered valuable insights during the creation of the DCDL-ESM system.

# References

[1] Adeyinka, A. M., Esan, O. C., Ijaola, A. O., & Farayibi, P. K. (2024). Advancements in hybrid energy storage systems for enhancing renewable energy-to-grid integration. Sustainable Energy Research, 11(1), 26. https://doi.org/10.1016/j.est.2025.116226

[2] Ullah, F., Zhang, X., Khan, M., Mastoi, M. S., Munir, H. M., Flah, A., & Said, Y. (2024). A comprehensive review of wind power integration and energy storage technologies for modern grid frequency regulation. Heliyon. https://doi.org/10.1016/j.heliyon.2024.e30466

[3] Nasiri, F., Ooka, R., Haghighat, F., Shirzadi, N., Dotoli, M., Carli, R., ... & Sadrizadeh, S. (2022). Data analytics and information technologies for smart energy storage systems: A state-of-the-art review. Sustainable Cities and Society, 84, 104004. https://doi.org/10.1016/j.scs.2022.104004

[4] Janssen, N., Ilayperuma, T., Jayasinghe, J., Bukhsh, F., & Daneva, M. (2024). The evolution of data storage architectures: examining the secure value of the Data Lakehouse. Journal of Data, Information, and Management, 1-26. DOI:10.1007/s42488-024-00132-1

[5] Sawadogo, P., & Darmont, J. (2021). On data lake architectures and metadata management. Journal of Intelligent Information Systems, 56(1), 97-120. https://doi.org/10.1007/s10844-020-00608-7

[6] Tang, S., He, B., Yu, C., Li, Y., & Li, K. (2020). A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications. IEEE Transactions on Knowledge and Data Engineering, 34(1), 71-91. DOI:10.1109/TKDE.2020.2975652

[7] Aghazadeh Ardebili, A., Hasidi, O., Bendaouia, A., Khalil, A., Khalil, S., Luceri, D., ... & Ficarella, A. (2024). Enhancing resilience in complex energy systems through real-time anomaly detection: a systematic literature review. Energy Informatics, 7(1), 96. DOI:10.1186/s42162-024-00401-8

[8] He, X., Ai, Q., Qiu, R. C., Huang, W., Piao, L., & Liu, H. (2015). A big data architecture design for smart grids based on random matrix theory. IEEE transactions on smart Grid, 8(2), 674-686. https://doi.org/10.1016/j.ensm.2020.12.008

[9] Li, D., Nan, J., Burke, A. F., & Zhao, J. (2024). Battery Prognostics and Health Management: AI and Big Data. World Electric Vehicle Journal, 16(1), 10. DOI:10.3390/wevj16010010

[10] Feng, N., & Ran, C. (2025). Design and optimization of distributed energy management system based on edge computing and machine learning. Energy Informatics, 8(1), 17. DOI:10.1186/s42162-025-00471-2

[11] Munshi, A. A., & Mohamed, Y. A. R. I. (2018). Data lake lambda architecture for smart grids big data analytics. IEEE Access, 6, 40463-40471. DOI:10.1109/ACCESS.2018.2858256

[12] Yu, W., Dillon, T.S., Mostafa, F., Rahayu, W., & Liu, Y. (2020). A Global Manufacturing Big Data Ecosystem for Fault Detection in Predictive Maintenance. IEEE Transactions on Industrial Informatics, 16, 183-192. DOI:10.1109/TII.2019.2915846

[13] Hamadou, H. B., Pedersen, T. B., & Thomsen, C. (2020, December). The danish national energy data lake: Requirements, technical architecture, and tool selection. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 1523-1532). IEEE. DOI:10.1109/BigData50022.2020.9378368

[14] Hai, R., Koutras, C., Quix, C., & Jarke, M. (2023). Data lakes: A survey of functions and systems. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12571-12590. DOI:10.1109/TKDE.2023.3270101

[15] Seven, S., Yao, G., Soran, A., Onen, A., & Muyeen, S. M. (2020). Peer-to-peer energy trading in virtual power plant based on blockchain smart contracts. Ieee Access, 8, 175713-175726. DOI:10.1109/ACCESS.2020.3026180

[16] Wang, S., Xiao, H., Zhao, Z., Li, D., Hu, D., Hu, Q., ... & Wang, L. (2025). Grid Peak Shaving and Energy Efficiency Improvement: Advances in Gravity Energy Storage Technology and Research on Its Efficient Application. Energies, 18(4), 996. https://doi.org/10.1016/j.rser.2025.116161

[17] Bin Abu Sofian, A. D. A., Lim, H. R., Siti Halimatul Munawaroh, H., Ma, Z., Chew, K. W., & Show, P. L. (2024). Machine learning and the renewable energy revolution: Exploring solar and wind energy solutions for a sustainable future including innovations in energy storage. Sustainable Development, 32(4), 3953-3978. DOI:10.1002/sd.2885

[18] Yu, S., Zhong, J., Yan, X., You, J., & Cai, Z. (2024). Architecture Design and Performance Optimization of Data Lake Architecture for Energy Storage Power Station Based on Distributed Computing Framework. Journal of Electrical Systems, 20(9s). https://doi.org/10.1016/j.polymertesting.2025.108912

[19] Barros, E. B. C., Souza, W. O., Costa, D. G., Rocha Filho, G. P., Figueiredo, G. B., & Peixoto, M. L. M. (2025). Energy management in smart grids: An Edge-Cloud Continuum approach with Deep Q-learning. Future Generation Computer Systems, 165, 107599. http://dx.doi.org/10.1016/j.ajodo.2005.10.031

[20] Liu, H., Zhang, X., Shen, X., & Sun, H. (2021). A federated learning framework for smart grids: Securing power traces in collaborative learning. arXiv preprint arXiv:2103.11870.

[21] Vempati, S. (2024). Whale Optimized Distributed Computing Data Lake for Energy Storage. Journal of Computer Allied Intelligence (JCAI, ISSN: 2584-2676), 2(5), 17-30. https://doi.org/10.32604/cmc.2023.037611

[22] Huang, X., Fan, J., Deng, Z., Yan, J., Li, J., & Wang, L. (2021). Efficient IoT Data Management for Geological Disasters Based on Big Data-Turbocharged Data Lake Architecture. ISPRS Int. J. Geo Inf., 10, 743. DOI: 10.1002/anie.202103557

[23] Ganapathi, J. (2025). Unified Data Processing with Spark Structured Streaming and Delta CDF on AWS.

International Journal of Scientific and Research Publications. DOI:10.1145/2934664

[24] Nuthalapati, A. (2023). Building scalable data lakes for Internet of Things (IoT) data management. Educational Administration: Theory and Practice, 29(1), 412-424. DOI:10.53555/kuey.v29i1.7323

[25] Alsalemi, A., Amira, A., Malekmohamadi, H., Diao, K., & Bensaali, F. (2022). Energy data lakes: An edge Internet of Energy approach. In Emerging Real-World Applications of Internet of Things (pp. 21-40). CRC Press. https://doi.org/10.1016/j.iot.2023.101035Get rights and content