International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 6173-6189

https://doi.org/10.70517/ijhsa464453

Cloud Computing and Big Data-Driven Optimization Method for Benchmarking Power Indicators

Wei Tong[1], Xiaomeng Liu[1], Gang Wang[2,*], Zuohu Chen[2] and Zhenguo Peng[2]
[1] State Grid Gansu Electric Power Company, Gansu, 730000, China
[2] Gansu Tongxing Intelligent Technology Development Co., LTD., Gansu, 730050, China
Corresponding authors: (e-mail: Wgang1616@163.com).

Abstract The existing power index benchmarking evaluation platform has large deviations in evaluation results and poor real-time performance due to the difficulty in integrating multi-source heterogeneous data, lack of index standards and lagging analysis mechanisms, which affects the scientific evaluation of power system operation performance. To solve this problem, this paper proposes an optimization solution based on cloud computing and big data technology. The innovation lies in the deep integration of standardized index system construction and intelligent benchmarking algorithm into the platform architecture. In the method design, Kafka and Flume are used to access data sources such as SCADA (Supervisory Control and Data Acquisition) and metering systems in real-time. Hadoop and Spark are used to complete data preprocessing and unified modeling. A unified index data warehouse is built based on Hbase (Hadoop Database) and Hive. A benchmarking evaluation model with cluster analysis and weighted scoring as the core is designed, and visualization and intelligent recommendation are realized under the Spring Boot framework. The experiment is conducted on a measured dataset of a regional power grid. After optimization, the response time of the platform is reduced to 51.5 seconds at a data scale of 100GB. The accuracy of the indicator benchmarking for the industrial park scenario reaches 86.4%, and the accuracy of low voltage anomaly detection is increased to 94.8%. The research results show that this method has significant practical value in improving data processing efficiency, enhancing evaluation accuracy, and supporting management decisions, and has a positive role in promoting the construction of intelligent management platforms in the power industry.

Index Terms Power System, Performance Evaluation, Cloud Computing, Big Data Analysis, Benchmarking Platform

I. Introduction

As an important part of national infrastructure, the safe and stable operation and efficient management of the power system are of great significance to ensuring social and economic development. With the continuous advancement of the construction of smart grids [1], [2], a large number of sensor devices and automation systems have been widely used, generating a huge amount of operation data. Through scientific benchmarking and evaluation of power operation indicators [3], [4], it is possible to accurately grasp the operation status of the power system [5], [6], discover potential hidden dangers, guide the optimization of operation and maintenance strategies, and improve overall operation efficiency and reliability. However, the current power industry [7], [8] generally has the phenomenon of scattered indicator data, lack of standards and lagging evaluation mechanisms, which limits the accuracy and effectiveness of benchmarking and evaluation. Data collection ends often come from diverse sources, involving multiple links such as power generation, transmission, distribution and sales. The data formats are different, and the real-time requirements are high, resulting in obvious obstacles to data fusion and sharing. The indicator system [9], [10] lacks a unified definition and calculation standard, which makes it difficult to effectively compare and comprehensively analyze the evaluation results of different regions and units. In addition, the evaluation methods mostly rely on traditional experience and static indicators, lack the ability to make intelligent judgments based on large-scale data analysis, and cannot timely reflect the dynamic changes and abnormal conditions in the operation of the power system [11], [12]. The functions of the existing platforms [13], [14] mostly stay at the display of results, lack support for in-depth data mining and trend prediction, and cannot meet the needs of the transformation of modern power systems to intelligent and refined management. The power industry has put forward higher requirements for the real-time and accuracy of data processing and the scientific nature of indicator evaluation. It is urgent to use advanced cloud computing [15], [16] and big data technology [17], [18] to break the

information island and build a unified, standard, and intelligent indicator benchmarking evaluation platform to help the efficient operation of the power system and decision optimization [19], [20].

The existing benchmarking platforms generally adopt a monolithic deployment structure, with high coupling between system interfaces, which cannot effectively support the rapid access of multi-source data and flexible calling between modules. The data processing process is still mainly based on batch processing, which leads to insufficient real-time performance. When facing high-frequency sampling and dynamic monitoring tasks, the response delay is significant, affecting the timeliness of anomaly identification and strategy adjustment. The platform function focuses on result presentation, lacks an in-depth analysis mechanism for abnormal trends and potential problems, and is not conducive to forming targeted optimization suggestions and executable strategies. There are problems such as inconsistent definitions, inconsistent structures, and large differences in granularity in data standards, which hinder the horizontal comparison and vertical tracking of benchmarking indicators. To improve the platform's capabilities in data fusion, indicator modeling, and intelligent analysis, it is necessary to optimize the system architecture, data model, and algorithm modules in an all-round way to achieve more precise, efficient, and intelligent indicator evaluation goals. This paper systematically verifies the actual effect and feasibility of the optimization scheme by building a unified data processing architecture, establishing a standardized indicator system, applying multidimensional clustering and weighted scoring mechanisms, and quantitatively analyzing the changes in response time and evaluation accuracy before and after platform optimization in experiments.

In view of the above difficulties, this paper designs an optimization solution for the power index benchmarking evaluation platform based on cloud computing and big data technology. The system builds a unified data access layer, uses Kafka as a real-time message middleware, and combines Flume for log data collection, ensuring the timely access and synchronous transmission of various heterogeneous data in the power system. In the data processing layer, a data lake is built based on the Hadoop distributed storage framework, and Spark is used for parallel computing and cleaning to convert raw data into standardized indicator data. To solve the problem of inconsistent indicator systems, a hierarchical indicator modeling solution covering operation, quality, safety, energy efficiency, and other dimensions is developed. By defining a unified indicator caliber and calculation rules, the comparability of indicators across regions and units is achieved. The data storage adopts a combination of HBase and Hive to achieve efficient storage and flexible query of indicator data. The evaluation layer designs a classification algorithm based on unsupervised clustering, combined with a weighted scoring model, to objectively quantify and dynamically compare the indicator performance of power units. The front end of the platform is based on the Spring Boot framework, and a responsive visualization interface is built to support diversified chart display and custom queries to enhance the user interaction experience. The system also integrates an intelligent early warning module, which can automatically detect and prompt abnormal indicators through time series analysis of historical data and assist management to quickly locate problems. This method is closely combined with the characteristics of the power business, systematically solving the problems of difficult data fusion, large differences in evaluation standards, and insufficient intelligent analysis, and promoting the development of power indicator benchmarking evaluation towards real-time, standardized, and intelligent directions.

II. Related Work

In the study of the benchmarking evaluation system of power indicators, multi-dimensional explorations have been carried out around system performance, evaluation methods, and low-carbon transformation needs, showing a trend of expanding evaluation dimensions and integrating methods. Liang et al. [21] integrated the macro development goals of the new power system with the operating characteristics of load-intensive cities, constructed a comprehensive indicator evaluation system covering dimensions such as safety, efficiency, and cleanliness, and verified its applicability and quantitative ability in supporting low-carbon decision-making in the context of megacities. To respond to the growing sustainability requirements of the power system, Deng et al. [22] systematically sorted out the development path of performance evaluation in the past two decades from the perspective of the Planning-Searching-Screening-Reporting-Reflecting cycle, pointed out that current work generally focuses on environmental and economic indicators and is highly dependent on Data Envelopment Analysis and Multi-Criteria Decision Making methods, and then proposed future optimization directions of strengthening social dimensions and integrating multiple methods. In terms of coping with system dynamic changes and operational risks, Liu et al. [23] constructed a frequency regulation coordination mechanism that integrated wind turbines and traditional thermal power units, and applied a multi-time scale evaluation framework to quantify the correlation characteristics between frequency stability and reliability parameters based on uncertain operation scenarios, and verified the accuracy of its modeling and analysis with the help of actual system cases. Overall, existing research still has shortcomings in the systematic construction of the evaluation framework, the

balance of evaluation dimensions, and the improvement of method adaptability, making it difficult to support the comprehensive benchmarking and intelligent decision-making needs of complex power systems [24]-[26].

To further address the adaptability of platform architecture in complex power scenarios, Xu et al. [27] systematically sorted out the inverter support capabilities in distributed renewable energy systems and conducted comparative analysis around voltage and frequency regulation and ride-through control strategies. They revealed the integration difficulties of existing algorithms under high penetration conditions and pointed out that functional integration and regulation algorithm optimization are the key directions for the evolution of benchmarking platform architecture. At the same time, in terms of benchmarking model design, Zhang et al. [28] proposed a clustering analysis framework for decision-making objectives, optimized the division of energy management scenarios by extracting key data attributes, and demonstrated strong strategy generation capabilities in real-time electricity price control and power dispatch, breaking through the structural isolation between traditional clustering models and evaluation objectives. In addition, in terms of multi-index comprehensive evaluation, Zubiria et al. [29] constructed an improved model of Technique for Order of Preference by Similarity to Ideal Solution based on triangular fuzzy numbers, integrated the opinions of multiple experts to achieve the optimal decision on energy storage technology, and verified its practicality in matching frequency regulation with inertial response capabilities in microgrid service scenarios, providing a feasible path for multi-criteria fusion and dynamic weight allocation in the platform.

As power systems increasingly rely on data-driven and intelligent dispatching, the integrated application of cloud computing and big data technologies has gradually become a key path to support the transformation of their operation and management. Al-Jumaili et al. [30] built a hypothetical model of core performance indicators of power systems in big data processing scenarios based on the performance characteristics of mainstream parallel programming models such as Hadoop and Spark, and proposed a real-time data management solution based on cloud computing architecture, aiming to break through the technical bottlenecks of traditional parallel computing in terms of energy efficiency and response delay. On this basis, AL-Jumaili et al. [31] proposed a cloud-based intelligent power management system design solution, combining DC (direct-current) power devices, renewable energy hybrid access, and optimization algorithms, and built a management architecture for multi-source data analysis, verifying its adaptability and effectiveness in reducing system energy consumption and operating costs. To further promote the unification of industry-level standards and implementation strategy specifications, Zhang et al. [32] established a multi-party collaboration mechanism covering power grid companies, utilities, and cloud service providers, systematically sorted out the practical motivations and challenges of the power industry's adoption of cloud technology from multiple aspects such as business needs, risk management, and implementation processes, and summarized executable migration path recommendations based on typical cases, which effectively supported the continuous improvement of industry regulatory rules such as the North American Electric Reliability Corporation. Although existing research has made progress in architecture design, performance optimization, and standard promotion, it still has the problem of insufficient application depth in dealing with real-time processing of high-frequency data streams, collaborative scheduling between platforms, and compatibility with multiple business scenarios [33]-[35].

III. Methods

III. A. Data Collection and Access

To achieve unified collection and efficient access of multi-source heterogeneous power system data, a data stream access mechanism based on Kafka and Flume is constructed. At the system architecture layer, by configuring the Flume Source module, the SCADA system, metering system, and automation terminal devices are directly connected to realize multi-channel and asynchronous data capture. Flume Channel adopts the dual-channel strategy of Memory and File to improve the cache capacity of high-frequency data streams and avoid delayed transmission problems caused by data congestion. At the data aggregation layer, Kafka is used as the message middleware to classify and distribute various types of power business data in Topic units, and the Producer side performs data compression encoding to ensure data integrity and efficiency during network transmission. To solve the problems of inconsistent data frequency and inconsistent format structure, the Avro serialization protocol is applied to complete structured packaging on the Kafka producer side to ensure data consistency and platform compatibility.

During the access process, the Zookeeper cluster is deployed to achieve load balancing and high availability of Kafka, and the data writing throughput is improved by reasonably dividing the number of Partitions and Replicas. To ensure timeliness and sequence, a consumption displacement recording mechanism is configured on the Kafka consumer side, combining the dual index of timestamp and offset to locate key data segments in streaming data and read them dynamically. The platform access layer integrates Flume Sink and Kafka Consumer, uses Spring

Boot as a container to manage the interaction logic of intermediate components, and implements modular deployment and dynamic parameter tuning through a unified YAML configuration file.

In view of the communication delay and unstable sampling frequency of some power data sources, a distributed time synchronization model is constructed, and a logical timestamp mechanism is applied to make the timestamp of any data packet:

$$T_i = max\{t_i, \underset{j \in N_i}{max}(t_j + \delta_{ij})\} \tag{1}$$

Among them, T_i represents the timestamp after the final calibration; t_i is the original sampling time; N_i is the set of adjacent nodes that communicate with node i; δ_{ij} is the estimated delay between nodes.

At the system scheduling layer, adaptive resource allocation of the collection and access process is realized, and a load balancing control strategy is designed to optimize thread allocation and resource scheduling by minimizing the following objective function:

$$\underset{x_{ij}}{min} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot c_{ij} \text{subjectto} \sum_{j=1}^{n} x_{ij} = 1, x_{ij} \in \{0,1\} \tag{2}$$

Among them, x_{ij} represents whether task i is executed by node j, and c_{ij} is the cost function under this allocation combination. Constraints ensure unique task allocation and avoid thread redundancy. Through real-time scheduling optimization, the system still maintains low delay and high throughput characteristics in high concurrency scenarios.

To ensure the stable operation and status visualization of the entire data access process, the platform applies the Kafka Streams monitoring module to track the message flow status, delay distribution, and data loss in real-time. Figure 1 shows the flow chart of the power data collection and access architecture, which clearly reflects the logical relationship between various data sources, collection components, middleware, and processing processes.

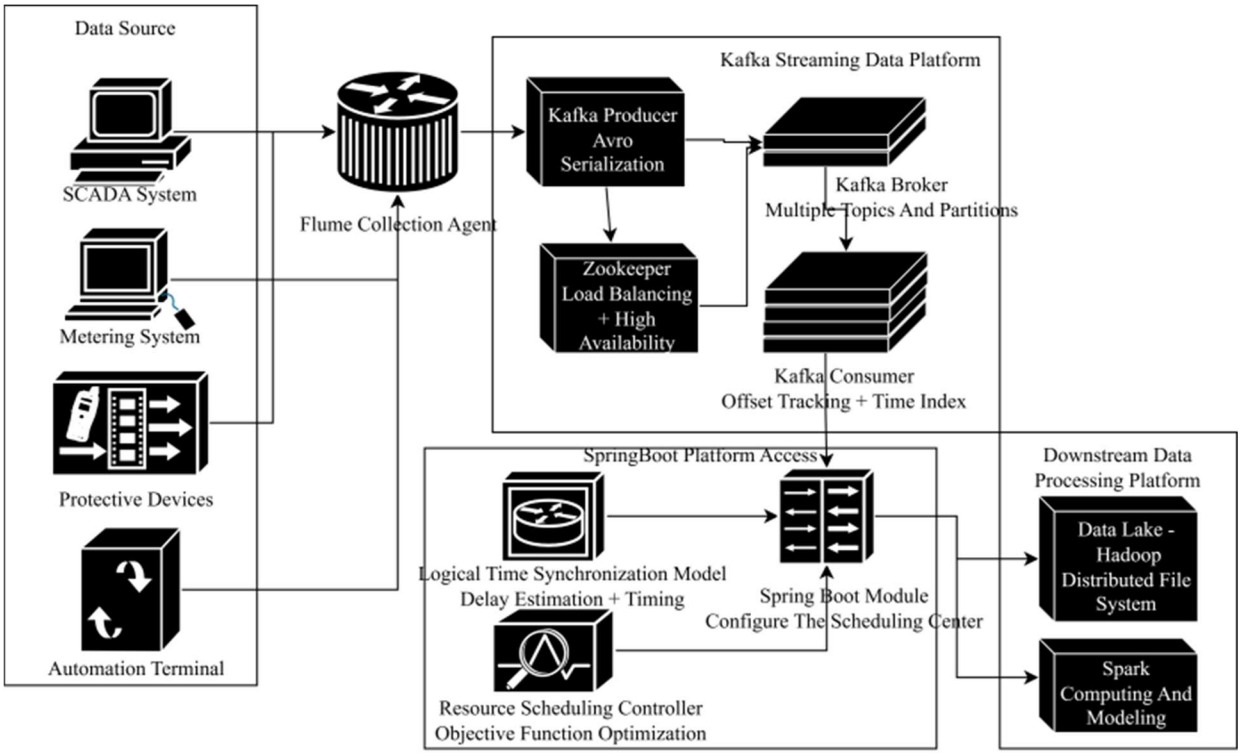


Figure 1: Architecture diagram of multi-source data access in power system

The data packet structure and sampling frequency of various data sources are further sorted out to assist in standardized modeling. Table 1 gives the characteristic parameter information of the current access data source, including data source type, number of fields, sampling frequency, daily data volume, and access delay.

Table 1: Statistical table of characteristic parameters of each data source

Data Source Type	Number of Fields	Average Sampling Frequency (Hz)	Daily Data Volume (GB)	Average Access Delay (ms)
SCADA System	38	1	5.6	83
Metering System	42	0.2	3.1	95
Protection Device	24	0.5	2.4	102
Automation Terminal	31	2	6.8	79

After the data flows into Kafka, the system divides the processing window based on the combination of batch and micro-batch strategies. The sliding window model is used to define the data input batch; the window length is set to W; the sliding step is S; the processing time complexity of each round is $\mathcal{O}(nW\log W)$, where n represents the number of concurrent data channels.

III. B. Data Preprocessing and Storage

To achieve unified management and efficient support for massive multi-source data in the power system, this platform uses the Hadoop ecosystem to build the main data processing framework and uses the distributed file system HDFS (Hadoop Distributed File System) to complete the unified storage of multi-source heterogeneous data such as SCADA systems, metering systems, and device operation records. After the data is connected, the MapReduce task is used to perform preprocessing operations such as field regularization, missing value filling, and outlier removal to ensure that the data quality meets the accuracy requirements of subsequent indicator modeling and benchmarking analysis. The data structure conversion is completed synchronously in the preprocessing stage, and the Parquet column storage format is used to compress redundant information to improve subsequent query performance and I/O (Inpit/Output) efficiency.

The cleaned data is logically divided according to the business dimension and time dimension. A multi-level index mechanism is built and uniformly loaded into the HBase and Hive dual engine architecture. HBase carries high-concurrency, low delay random read and write tasks to support real-time calculation of indicators and online update of models. Hive undertakes batch query and historical data statistics tasks to assist benchmark model training and trend analysis. To solve the time series aggregation and multi-dimensional association problems involved in the calculation of power indicators, the Spark SQL (Structured Query Language) task framework based on window functions is used to perform pre-aggregation operations. The indicator calculation logic is dynamically scheduled by the formula parsing engine. Each indicator definition is converted into a unified expression, and the indicator value generation process is completed through the Spark execution environment.

Considering the significant structural differences between different data sources, different sampling frequencies, and confusing field naming, a cross-source mapping strategy and field standardization mapping table are designed, and an automatic alignment mechanism is established. The configuration center and metadata management module are relied on to perform unified field renaming and data timestamp alignment to ensure the consistency of input data in the indicator warehouse. The system uses the following standardized processing formula to perform normalization operations on various continuity indicators to enhance cross-unit comparison capabilities:

$$X_{ij}^{*} = \frac{X_{ij} - min(X_j)}{max(X_j) - min(X_j)} \tag{3}$$

Among them, X_{ij} represents the original indicator value of the j-th item in the i-th record, and $max(X_j)$ and $min(X_j)$ are the maximum and minimum values of the indicator in the current analysis cycle. The triple standard deviation method is used to detect and eliminate abnormal records, and the formula is as follows:

$$|X_{ij} - \mu_j| > 3\sigma_j \tag{4}$$

Among them, μ_j is the mean of the j-th indicator, and σ_j is its standard deviation. Records that exceed the set range are marked as abnormal data and do not participate in the subsequent indicator calculation.

The data warehouse adopts a snowflake modeling strategy during the design process, constructs standard dimension tables for dimensions such as operating units, voltage levels, and device types, and associates them to fact tables through unique primary keys to support multi-dimensional analysis queries.

To improve the update efficiency and consistency of warehouse data, the system configures scheduled batch processing jobs and incremental update mechanisms to perform incremental loading of daily new data, and uses the Lambda architecture to integrate real-time and batch calculation results to build a hybrid processing channel. At the data call layer, the platform establishes an interactive query interface based on Impala, which supports combined retrieval and conditional filtering of any indicator field, significantly improving the response speed of indicator analysis and model debugging.

According to the indicator call frequency and data access mode in the platform operation, the frequency distribution of various query requests in the typical operation cycle is statistically analyzed, as shown in Table 2:

Table 2: Frequency distribution statistics of different types of query requests

Query Type	Average Daily Requests	Peak Requests (Per Day)	Percentage of Total Requests (%)
Real-Time Indicator Value Query	12,430	18,205	34.3
Historical Trend Query	8,017	12,064	22.1
Abnormal Indicator Detection	5,209	9,181	14.4
Comparative Analysis (Multi-Unit)	4,376	7,834	12.1
Aggregated Statistics Query	3,588	6,097	9.9
Indicator Definition Invocation	1,842	2,411	5.1
Auxiliary Queries (Config, Auth)	745	1,002	2.1

The data in Table 2 shows that the query requests for the current value and historical trend of indicators account for 56.4% of the total, which constitutes the core part of the platform access load and places high demands on the system response time and concurrent processing capabilities. Requests for abnormal screening and comparative analysis show obvious peak fluctuations, reflecting the characteristics of periodic centralized execution of analysis tasks, which poses a challenge to the efficiency of batch processing and the stability of data pre-aggregation mechanism. The access volume of dimension aggregation requests is relatively low, but the processing cost is high, indicating that query performance needs to be improved through index optimization and storage structure adjustment. The proportion of indicator definition and auxiliary query is extremely small, indicating that the metadata layer is highly stable; the call demand is sparse; the impact on overall performance is limited.

To optimize the access performance of key indicators, a hot and cold partitioning strategy is adopted to cache the frequently accessed indicator data in Redis for front-end calls to reduce the pressure on the back-end database. The version management and evolution of indicators are controlled by the metadata center, and the indicator definition change history is recorded through the version number to ensure the traceability and consistency of indicators.

III. C. Construction of Power Index System

To achieve standardized management and high-precision benchmarking evaluation of power indicators, the construction of the index system is based on the core principles of “unified definition, hierarchical classification, and quantitative expression”. A combination of business structure modeling and statistical characteristic analysis is adopted to systematically organize multi-level index items under the four core dimensions of operation, quality, safety, and energy efficiency. By analyzing the coupling relationship between management data and operation data of various power companies, an index mapping map is constructed. Based on the comparability and structural dependence of indicators between different business units, the integrated reconstruction of indicators on the time scale and space scale is completed.

The indicator definition adopts a modeling method that integrates attribute constraints and rule reasoning. For the noise and drift in the original data sequence, a multivariate collaborative sliding window algorithm is used to perform dynamic smoothing processing to eliminate nonlinear disturbance terms. Assuming that the input data stream is $X_t \in \mathbb{R}^{n\times m}$, where n represents the number of monitoring points, and m represents the number of sampling moments, the multi-indicator window smoothing function is defined as:

$$\hat{X}_t = \frac{1}{2k+1}\sum_{i=-k}^{k} W_i \cdot X_{t+i} \tag{5}$$

Among them, W_i is the weight factor matrix, which is adaptively adjusted according to historical stability and real-time variability to effectively improve the representativeness of the indicator value. The indicator standardization adopts the interval mapping and Z-Score joint normalization strategy, and the unified standardization formula is defined as:

$$Z_{ij} = \frac{x_{ij}-\mu_j}{\sigma_j} \tag{6}$$

Among them, x_{ij} represents the observed value of the j-th indicator in the i-th sample, and μ_j and σ_j are the mean and standard deviation of the indicator, respectively.

To enhance the logical consistency and dependency expression between indicators, the indicator hierarchical modeling method based on graph structure is adopted to construct the indicator association directed graph G = (V, E). Among them, the node V represents the indicator set, and the edge E represents the functional dependency or

causal path between indicators. According to this structure, the indicators are causally disassembled and grouped to form an indicator cluster structure for functional modules. The optimized indicator system is divided into four first-level indicator dimensions, thirteen second-level subcategories, and sixty-four third-level indicator items. The specific structure distribution is shown in Table 3.

Table 3: Hierarchical structure and coverage statistics of the power indicator system

Primary Dimension	Number of Secondary Categories	Number of Tertiary Indicators	Number of Source Systems for Sample Data
Operation	4	18	5
Quality	3	14	4
Safety	3	17	3
Energy Efficiency	3	15	6

The hierarchical structure of the power indicator system shown in Table 3 reflects the distribution characteristics and structural coverage of multi-dimensional indicators in the systematic construction. The first-level dimension is divided according to the core operating elements of the power system; the second-level category is refined according to the functional attributes; the third-level indicators are formed through statistical screening on the basis of maintaining business representativeness. The number of data source systems reflects the support breadth and coverage depth of various indicators in different platforms.

To quantify the degree of dispersion and outlier risk between indicators, a discriminant function based on information entropy is applied on the standardized dataset to evaluate the stability level of various indicators. It is assumed that the dispersion of the j-th indicator after standardization on the entire sample set is:

$$H_j = -\sum^{n} p_{ij} \log\left(p_{ij}\right) \tag{7}$$

Among them, $p_{ij} = \frac{z_{ij}}{\sum_{i=1}^{n} z_{ij}}$ represents the normalized weight of the i-th sample on the indicator.

Figure 2 shows the key processing flow of multi-source data in the construction of a unified indicator system for the power system. Starting from data sources such as SCADA, metering system, and operation log, after data alignment and fusion, sliding window processing, standardization transformation, graph structure modeling, and function clustering, a multi-level indicator structure system covering four dimensions of operation, quality, safety, and energy efficiency is finally formed.

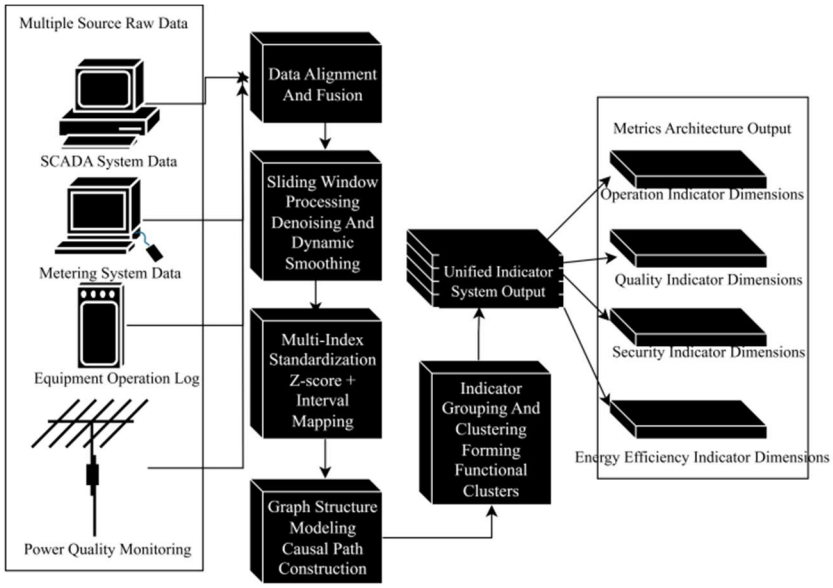


Figure 2: Architecture diagram of the construction of a multi-source data fusion and standardized indicator system for power

III. D. Design of Intelligent Benchmarking Evaluation Models

In response to the problems of large deviations in the evaluation results of power indicators and lack of dynamic analysis capabilities, this study constructs an intelligent benchmarking evaluation model that integrates

unsupervised clustering and weighted scoring mechanisms to achieve multi-dimensional quantification and dynamic classification of power system operation performance. The model operates on the basis of standardized indicators that are uniformly modeled in the data warehouse. First, based on the K-means algorithm, similar power units are classified. By presetting the number of cluster centers K, the samples in the normalized indicator vector space are iteratively clustered so that each unit belongs to the most similar performance cluster. Assuming that the sample space is $X = \{x_1, x_2, \ldots, x_n\}$, where each $x_i \in \mathbb{R}^d$ represents the d-dimensional standardized indicator vector of the i-th unit, the objective function is to minimize the sum of squared errors within the cluster:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} \| x - \mu_i \|^2 \tag{8}$$

Among them, C_i is the i-th cluster, and μ_i is its centroid. To enhance the robustness of clustering, the silhouette coefficient method is used to evaluate the clustering quality and automatically adjust the optimal cluster number K. After clustering is completed, the power units in each cluster are horizontally compared and ranked according to the weighted scoring method.

The scoring function is constructed based on the weight distribution of each standardized indicator, and the entropy weight method is used to adaptively calculate the indicator weight. Assuming that the standardized value of the j-th indicator in n units is x_{ij}, its entropy value is:

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^{n} p_{ij} \ln p_{ij}, p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \tag{9}$$

After calculating the information entropy of each indicator, its weight w_j is obtained by the following formula:

$$w_j = \frac{1 - e_j}{\sum_{j=1}^{m} (1 - e_j)} \tag{10}$$

This method avoids the bias caused by artificial subjective weighting, so that the evaluation weight of the indicator in different dimensions is determined by its distribution difference. The comprehensive score S_i is the weighted sum of each weight and the unit standardized indicator value:

$$S_i = \sum_{j=1}^{m} w_j \cdot x_{ij} \tag{11}$$

In the implementation of the evaluation model, Spark Machine Learning Library is used to implement the parallel operation of the K-means algorithm, and the data samples are mapped through the RDD (Resilient Distributed Dataset) structure, which significantly reduces the resource overhead of large-scale samples in multi-dimensional space calculation. In the scoring module, Hive SQL and Spark DataFrame are combined to build data extraction logic to ensure seamless switching between weight calculation and scoring functions in batch and stream processing scenarios. A clustering result cache mechanism is also designed in the system architecture to avoid frequent recalculation affecting real-time performance.

To ensure the adaptability of the model to the multi-dimensional indicator structure, the system supports users to customize the evaluation dimension combination at the front end, and automatically adjusts the dimension space and retrains the clustering model through the vector mapping function at the back end, supporting comprehensive classification analysis of power units in different directions such as operating efficiency, quality stability, and energy efficiency level. In addition, a difference analysis module is set at the output end of the model to identify key gap indicators through the significance test of the mean indicators of units between different clusters, providing a decision-making basis for subsequent optimization suggestions.

III. E. Visualization and Intelligent Warning

The platform builds a responsive display and interaction system based on Spring Boot. The backend adopts the RESTful (Representational State Transfer) architecture to design interfaces. MyBatis is used to implement data interaction with HBase and Hive to meet the real-time reading requirements of indicator data. The front end uses Echarts as a graphic display engine to build multi-dimensional chart components, supporting line charts, heat maps, radar charts, etc., and improves response efficiency through asynchronous data binding and dynamic refresh mechanisms. The chart presentation logic is adaptively adjusted based on the indicator dimension and time window, and the user interface integrates filtering, sorting, and focusing operations to ensure the hierarchical and targeted information transmission. The indicator query parameters are embedded in the interface, and the data request and graphic update are automatically triggered after the user inputs, forming a highly coupled feedback closed-loop of query and display.

The visualization module is closely bound to the indicator classification structure, and a hierarchical data view is constructed around operating efficiency, power quality, safety stability, and energy efficiency level. The graphics and numerical tables are synchronized. The platform responds to high-frequency refresh requests through the

front-end rendering cache and lazy loading mechanism to improve the interface response performance of the system in a concurrent environment. The data refresh cycle is uniformly managed by the backend scheduling module. Quartz is used to implement a minute-level update strategy. Combined with the user request-driven mechanism, the component content is asynchronously refreshed under the trigger of manual interaction to reduce unnecessary repeated calculations of the system.

The early warning mechanism is built on the basis of the indicator anomaly detection model. Based on time series analysis, trend prediction and deviation judgment are performed on continuous data. The exponential weighted moving average method is applied when building the baseline model, and the prediction function is defined as:

$$\hat{x}_{t+1} = \alpha x_t + (1-\alpha)\hat{x}_t \tag{12}$$

Among them, $\hat{x}_{t+1}$ is the predicted value; x_t is the current observation value; α is the smoothing coefficient, and the value is obtained by reverse optimization based on the historical error. Deviation detection adopts the statistical discrimination method, and the anomaly judgment formula is:

$$|x_t - \hat{x}_t| > 3\sigma_t \tag{13}$$

Among them, σ_t represents the sample standard deviation in the sliding window.

To enhance the recognition ability of nonlinear mutations, the system integrates the isolation forest algorithm to identify non-periodic abnormal signals in indicators. Isolation forest constructs multiple binary trees based on random partitioning and performs expected modeling on the average path length of samples. The anomaly score is defined as:

$$s(x) = 2^{-\frac{E(h(x))}{c(n)}} \tag{14}$$

Among them, $E(h(x))$ is the average path length of samples in the isolation tree; $c(n)$ is the expected value of the normal path length when the number of samples is n, which is approximately $2\ln(n) + \gamma$; γ is the Euler constant.

The early warning system accesses the indicator abnormal event recording module, marks the abnormal indicator, deviation amplitude, trigger time, and system information for each alarm event, and associates the recent operation log and control instruction record to form a preliminary abnormal attribution path. During the operation of the system, all alarm events are uniformly written into the early warning log database and pushed to the front-end interactive interface in an asynchronous manner to form a closed-loop feedback mechanism. The multi-dimensional indicator correlation analysis constructs a correlation map through the covariance matrix between indicators and combines the principal component analysis method to select the key nodes of the abnormal propagation path to assist managers in the classification of warning levels and disposal strategy decisions.

The platform designs a delay monitoring mechanism to record the time cost of indicator data from access, preprocessing, writing, query to front-end rendering to form a delay distribution dataset. Based on this dataset, the system updates the processing performance status in real-time in the background, dynamically adjusts the concurrent thread resource allocation strategy, and performs load balancing control on the key path nodes to ensure the response stability under abnormally high load conditions. The user interface supports multi-dimensional interactive operations. Indicator switching, chart focus, and time granularity adjustment are all completed in local components, without relying on the overall page refresh, improving operation fluency and information accessibility.

IV. Experiments

IV. A. Experimental Environment Construction

The experimental platform of this study is deployed in a cloud computing environment with elastic expansion capabilities. The overall architecture is divided into data acquisition layer, computing processing layer, data storage layer, and application service layer. The experiment adopts a private cloud deployment solution, builds a virtualized resource pool based on OpenStack, supports multi-node parallel processing tasks, and meets the distributed processing requirements of large-scale power data. The operating system is CentOS 7.9 (Community ENTerprise Operating System), and the underlying hardware configuration uses Intel Xeon Gold series processors with a main frequency of 2.6GHz, 128GB of memory, and a single-node hard disk capacity of 10TB. A storage cluster is established to improve read and write efficiency. The network environment is Gigabit Ethernet interconnection to ensure the stability and low delay of data transmission between nodes.

The big data processing framework uses Hadoop 3.3.1 and Spark 3.1.2 to run together. HDFS is used for persistent storage, supporting sharding and replica mechanisms to ensure data redundancy and reliability. The

Spark cluster runs on the YARN (Yet Another Resource Negotiator) resource scheduling system, setting the ratio of Driver and Executor resources to 1:5 to balance task scheduling and computing resource utilization. Kafka 2.8.0 is deployed in the message queue layer to support partitioning mechanism and asynchronous transmission to improve data collection throughput. Flume is used as a data transmission agent to achieve real-time data access and transmission scheduling for data sources such as SCADA systems and metering terminals, and supports multi-channel flow control and fault-tolerant forwarding mechanisms.

In the data storage layer, HBase 2.4.9 is used to build a time series storage structure for power indicators, and the column storage mode is used to optimize the retrieval efficiency of indicator-level data. Hive 3.1.3 is integrated in the Hadoop ecosystem to achieve batch data analysis and structured query, support docking with the Spark SQL engine, and improve query response speed and semantic processing capabilities. Metadata management is uniformly maintained by Hive Metastore to ensure data semantic consistency and access uniformity among modules. The platform control logic is developed based on Spring Boot 2.6.7 to achieve service decoupling, dynamic module deployment, and system load monitoring.

Zookeeper is integrated between nodes for service coordination and status monitoring to ensure operational consistency under the modular deployment of the platform. The system security layer integrates the Kerberos protocol for authentication and authorization to avoid security risks in the data access process and improve the credibility and stability of the platform's overall operation.

IV. B. Dataset Preparation

The experimental data used in this study is derived from the real operation data of a regional power grid, covering multiple business platforms such as SCADA, electricity metering system, energy efficiency management system, and safety monitoring system. The data types cover voltage, current, active power, reactive power, frequency, load rate, device status code, energy efficiency index, and fault information. The data has high-frequency update characteristics and strong heterogeneity. The total amount of data reaches 6.3TB, and the time span covers 120 consecutive days. To ensure the representativeness and integrity of model training and evaluation, all data are systematically integrated, time-series aligned, and null value removed, and uniformly converted into standardized indicator format.

During the preprocessing process, the data is first uniformly stored in the HDFS cluster and structured in Parquet columnar format. When regularizing the original data, a sliding aggregation method based on time window is used to aggregate minute-level data into indicator blocks with a period of 15 minutes. At the same time, a unified dimension and value range are set for each type of indicator, and the dimension effect is eliminated through the Z-score standardization method to improve the comparability of data and the stability of analysis. In the data integrity review, Spark SQL is used to perform batch statistics and screening on the missing rate of each data source. The missing threshold is set to 10% to remove the high missing fields. The sliding mean interpolation strategy is used to repair the low missing fields.

To support the subsequent benchmarking model training and classification analysis, it is necessary to identify and label the data of different power units. The data involves four types of power entities, including power supply stations, substations, switch stations, and main grid dispatching centers in the district. The system sets an independent identification code for each type of entity and completes data mapping. In the data partition design, grouping is carried out according to the dual dimensions of geographical area and voltage level to achieve hierarchical organization of data in the logical structure. After preliminary summary and review, a unified indicator sample set is formed, providing a complete indicator vector for a total of 1,853 independent units. Each unit includes four categories: operation, quality, safety, and energy efficiency, which has a sufficient analysis basis.

In terms of dataset division, the ratio of training set to test set is set to 8:2, and the ratio of each type of unit is kept consistent to ensure the model's generalization ability during training.

IV. C. Parameter Setting and Operation Process

This section elaborates on the parameter configuration and operation process used by the benchmarking evaluation platform during the experiment to ensure that the experiment is reproducible and scientific in terms of technical implementation and data processing. Model training and system operation both rely on reasonable parameter settings, which must take into account data scale, computing efficiency, and indicator stability. The platform's parameter configuration covers multiple aspects such as data collection batch control, preprocessing task scheduling, clustering analysis parameter selection, scoring model weight allocation, and warning threshold setting.

In the data collection stage, the number of Kafka partitions is set to 8 to ensure that the parallel processing capability matches the message transmission stability. The collection interval in the Flume configuration is

uniformly set to 5 seconds, and the buffer size is 128MB to ensure data synchronization performance between different data sources. In the data preprocessing stage, the maximum number of concurrent users of Hadoop's MapReduce task is set to 20. The data cleaning rule uniformly interpolates missing values, uses a fixed value standardization strategy for unified unit conversion, and uses a unified field mapping rule for field specifications. The data batch processing is achieved three times a day by configuring the scheduler.

In the model building stage, the K-Means clustering algorithm is used for unsupervised classification. The number of clusters is set to 6, and the maximum number of iterations is set to 100. The initial center point adopts a random initialization strategy, and the Euclidean distance is used as the similarity metric. The weights of each indicator in the benchmarking scoring model are calculated by the entropy weight method, where the weights of operation indicators, quality indicators, safety indicators, and energy efficiency indicators are 0.30, 0.25, 0.25, and 0.20, respectively. The final score calculation is based on the weighted normalized score, and the score range is 0 to 100. In the early warning module, the abnormal identification is set to trigger a primary warning when the indicator deviates from the standard range by more than 15%, and trigger an advanced warning when it deviates by more than 30%. The time series analysis module sets the sliding window to 10 minutes to identify trend deviations.

In terms of the operation process, the platform is built on the Spring Boot framework as a whole, and the task scheduling and data display are decoupled through the front-end and back-end separation mode. After the operation is started, the data collection module is activated first. Kafka is connected to data sources such as SCADA, metering system, and scheduling platform, and Flume concurrently pulls each data stream and writes it to HDFS. Subsequently, YARN schedules and starts the data cleaning and conversion process. Hadoop formats the raw data, normalizes the indicators and maps the fields, and updates the HBase indicator table synchronously. Hive, as the query middle layer, provides a unified query interface for the model module to access.

After the model scheduling layer loads the training model, it clusters the current batch data, builds a scoring archive based on historical data, and then compares the current unit score with the mean of similar units to output the benchmarking results. The intelligent early warning module automatically identifies abnormal indicators after the score is generated, and pushes the corresponding early warning logo and suggestion content in the front-end display platform. The platform interface displays cluster distribution, scoring trend, and abnormal indicator statistics through ECharts graphic components, and user interaction behavior is synchronously fed back to the recommendation module to form a data closed-loop.

V. Results

V. A. Data Processing Efficiency

This section conducts quantitative experiments on the performance response of the platform in big data processing tasks and multi-user concurrent scenarios, and conducts hierarchical verification of key performance indicators. By setting data scales ranging from 1GB to 100GB, the actual workflow of the platform in the access, preprocessing, and indicator modeling stages is simulated, and the changing trend of its response delay under load growth is analyzed; by constructing concurrent access scenarios of 10 to 1000 users, the resource allocation and throughput performance of the platform under intensive task scheduling are simulated. Figure 3(a) shows the response time performance of the platform under different data volume conditions, and Figure 3(b) reflects the changes in the throughput capacity of the system under different concurrent loads.

(a) Response time comparison

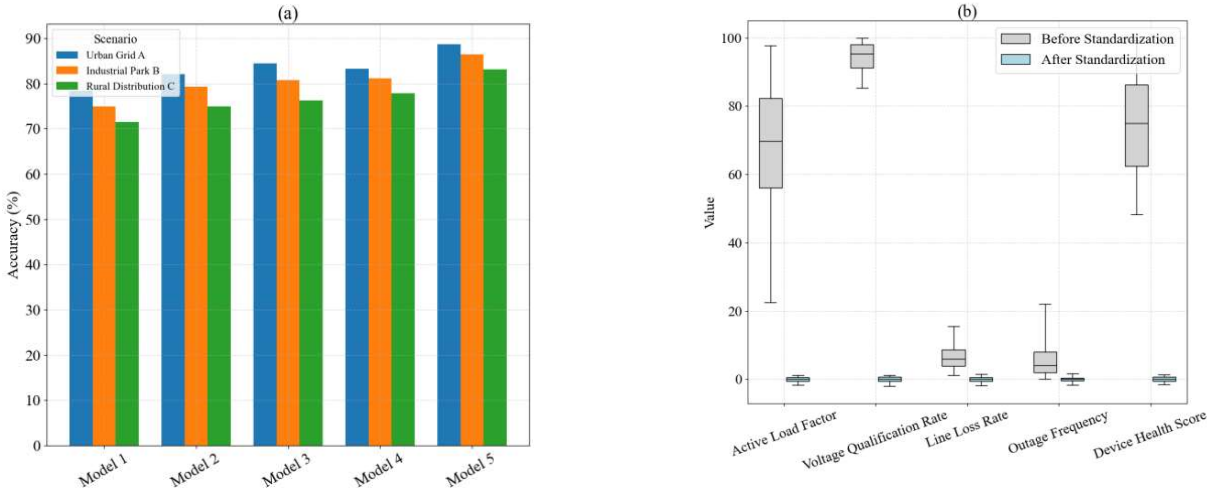
(b) Throughput comparison

Figure 3: Platform processing performance comparison chart: response time and throughput dual perspective evaluation

The optimization effect of the platform on data processing efficiency is quite significant. As the data volume expands from 1GB to 100GB, the platform response time before optimization increases from 4.7 seconds to 329 seconds, while the response time after optimization is controlled within 51.5 seconds, indicating that the system has good stability when facing high-concurrency data processing. The changing trend of the response time growth rate reflects the expansion advantages of Hadoop and Spark distributed architecture in batch task processing, and significantly reduces processing delays through task scheduling and load balancing between nodes. In terms of throughput, the throughput of the platform before optimization decreases after the number of concurrent users reaches 100, showing a resource bottleneck. After optimization, the platform maintains an increasing trend with the increase of concurrent users, up to 48.2MB per second, showing the processing elasticity after the coordinated scheduling of Kafka message distribution mechanism and Spark Streaming real-time computing framework. The experimental results show that the constructed optimized architecture shows stronger processing capabilities and response stability when facing multi-dimensional business pressure.

V. B. Indicator Benchmarking Accuracy

To improve the applicability of benchmarking evaluation models in different scenarios and the effectiveness of standardization strategies, this section conducts an in-depth comparative analysis from two dimensions: model accuracy and indicator data distribution. The comparison objects cover five types of evaluation models. Model 1 is a static evaluation method based on expert rules, which is suitable for business scenarios with clear rules but lack of generalization ability. Model 2 is a benchmarking analysis method based on K-Means clustering, which relies on the distance division of feature space but is sensitive to outliers. Model 3 is an indicator evaluation model based on principal component analysis and weighted scoring, which has certain explanatory power but is easy to weaken local structure. Model 4 is a comprehensive scoring method based on hierarchical clustering analysis, which is suitable for processing complex objects with nested structures. Model 5 is an intelligent benchmarking evaluation method that integrates standardization-clustering analysis-weighted scoring, which realizes dual modeling of business heterogeneity and dynamic correlation of indicators. At the indicator level, the Active Load Factor selected in this study represents the balance of user power load; Voltage Qualification Rate reflects the degree of voltage qualification; Line Loss Rate reflects the efficiency of energy loss; Outage Frequency represents the continuity of power supply; Device Health Score measures the stability of device operation status and early warning needs. The above models and indicators together construct a multi-level information foundation for benchmarking evaluation. Figure 4 (a) shows the accuracy differences of each model in three typical power grid scenarios (Urban Grid A, Industrial Park B, Rural Distribution C), and Figure 4 (b) compares the statistical distribution characteristics of each indicator before and after standardization in the form of a box plot.



(a) Comparison of the accuracy of different models

(b) Comparison of the statistical distribution before and after index standardization

Figure 4: Comparison of the accuracy of different models and the statistical distribution before and after index standardization

Model 5 shows a leading accuracy in all scenarios, reaching 86.4% and 83.1% in industrial parks and rural distribution networks, respectively. This performance is attributed to the fact that its clustering algorithm takes into

account the global distribution structure while maintaining local density characteristics, avoiding the misclassification of marginal samples by traditional methods. In contrast, the accuracy of model 1 in all three scenarios is lower than 79%, mainly limited by the coverage of rules and the lag in strategy updates. Models 2 and 4 perform relatively close. On the other hand, it can be seen from the index distribution diagram that after standardization, the box shape of each index tends to be symmetrical, and the maximum and minimum values converge significantly. Among them, the outlier situation of Outage Frequency is effectively suppressed. This change improves the comparative consistency and generalization stability of the model under different dimensions. The standardized results also enhance the robustness of the weighting mechanism to extreme value indicators, thereby avoiding the problem of a single indicator dominating the overall score in the evaluation results. Overall, the benchmarking system built based on unified standardized processes and structure-aware algorithms shows higher robustness and evaluation accuracy when dealing with heterogeneous scenarios and complex indicator structures.

V. C. Real-time Performance and Response Delay

To evaluate the processing capabilities of the optimized power indicator benchmarking evaluation platform under different data loads, a response time measurement experiment for five core processing stages is designed, covering typical processes such as data collection and access, preprocessing, indicator modeling, warehouse query, and visualization output. There are significant differences in the computing tasks and system call complexity carried by each stage. The data collection and access stage relies on stream processing components such as Kafka and Flume, and the response delay is less affected by network I/O and caching strategies; a large number of ETL (Extract, Transform, Load) operations are required in the data preprocessing stage, which is easy to form a computing bottleneck; indicator calculation and modeling involve clustering and weighted scoring algorithms, which are highly sensitive to memory and CPU usage; the performance of the data query stage is related to the warehouse design and indexing mechanism; the visualization stage mainly examines rendering efficiency and interface responsiveness. In the experimental setting, the input data scale is controlled to 100,000, 500,000, and 1 million records to observe the response time change trend of the platform under different load scenarios. Figure 5 shows the average response delay distribution of each processing stage under three data scales.

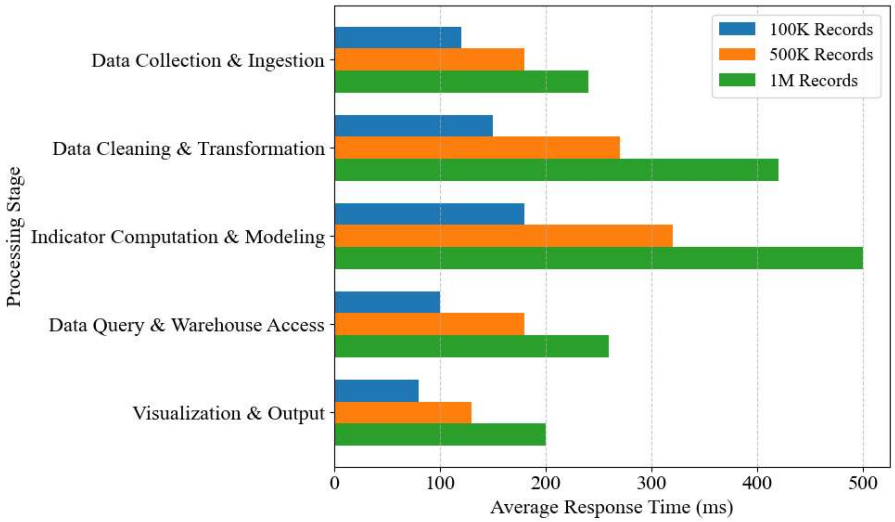


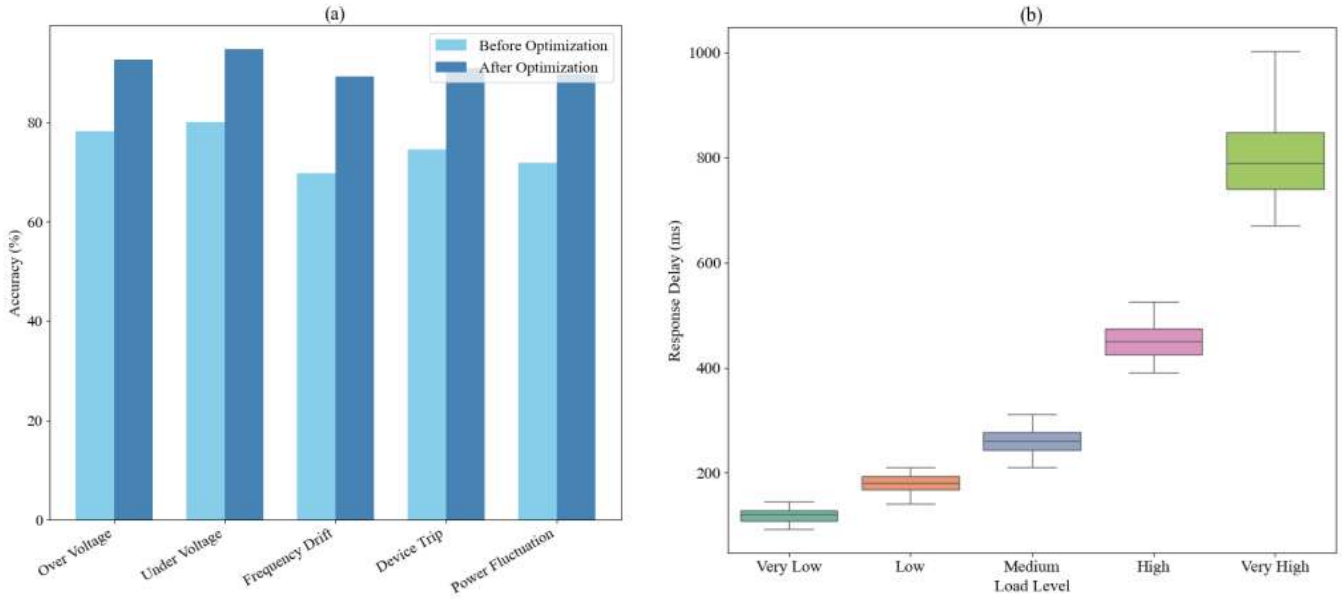
Figure 5: Response time distribution of each processing stage of the platform under different data scales

In terms of the overall trend, data preprocessing and indicator modeling stages always dominate the response time under all data scales. Among them, when 1 million data are input, the average time consumed by the preprocessing stage reaches 420 milliseconds, while the indicator modeling stage is even higher, reaching 500 milliseconds, reflecting the high dependence of these two stages on computing resources in the big data environment. In contrast, the data collection stage benefits from Kafka's asynchronous buffering mechanism and Flume's batch processing optimization, and the response time increases more slowly, only increasing by 120 milliseconds between 100,000 and 1 million data scales, showing good scalability. The data query stage is affected by HBase column storage and Hive query optimization, and the delay is controlled within 260 milliseconds, showing the effectiveness of the platform in query path compression and index loading strategies. The delay of the

visualization stage is always kept within 200 milliseconds, mainly due to the low computing density and cache mechanism support of front-end rendering and data aggregation tasks. Overall, the platform still maintains good stage response stability and distribution balance under high load, indicating that it has good adaptability in multi-stage collaborative processing and load stratification optimization.

V. D. Evaluation of Intelligent Early Warning Effect

To verify the overall optimization effect of the platform in terms of abnormal identification accuracy and response performance, this paper designs a set of experimental schemes covering typical fault types and multi-level load scenarios to systematically evaluate its robustness performance in complex operating environments. The experiment selects five common anomalies: overvoltage, undervoltage, frequency drift, device tripping, and power fluctuation, corresponding to different risk dimensions such as power quality, frequency stability, and device safety. This selection takes into account both steady-state disturbances and sudden faults, and has good representativeness and identification differences. Among them, voltage anomalies directly affect the power supply quality on the user side; frequency drift reflects the source-load power imbalance; device tripping is usually triggered by short-term shocks or protection strategies; power fluctuations reflect the uncertainty of load-side operation. In addition, to simulate the performance of the platform under different load pressures, the experiment sets five load levels (from extremely low to extremely high), covering typical business scenarios from no-load debugging to peak operation. The data flow density increases significantly during the high-load stage, which often causes processing delays and resource conflicts, and is a key test point for the system's responsiveness. Figure 6(a) shows the difference in the accuracy of various anomaly recognition before and after optimization, and Figure 6(b) describes the distribution and stability boundary of the platform response delay under different load levels.



(a) Anomaly detection accuracy (b) Load response delay performance

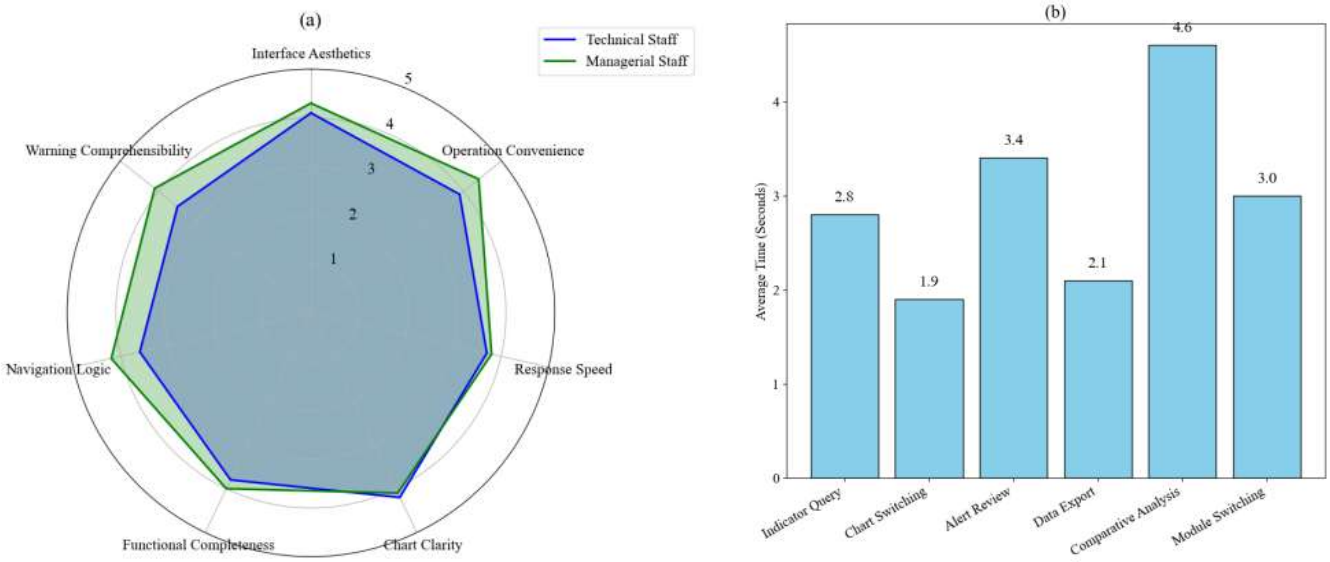
Figure 6: Comparative analysis of anomaly detection accuracy and load response delay performance

From the analysis of detection accuracy, the accuracy of voltage anomalies has increased to more than 92% after optimization. Among them, the accuracy of voltage under-detection reaches 94.8%, an increase of 14.7 percentage points compared with before optimization, indicating that with the support of unified indicator definition and feature modeling, the system is more sensitive to the extraction of steady-state anomaly features. The detection effect of frequency drift and power fluctuation has been improved more significantly, with the accuracy rates of 89.2% and 89.6%, respectively after optimization, an increase of nearly 19.5 percentage points and 17.7 percentage points. This proves that after the application of the sliding window trend analysis mechanism and the isolation forest algorithm, the system has significantly enhanced its ability to extract short-period nonlinear disturbance features, thereby improving the recognition accuracy of power quality and frequency anomalies. It is worth noting that the accuracy of device tripping detection has increased by 16.4% simultaneously, which is closely related to the expansion of the impulse current waveform sample library and the mining of protection action

association rules. The response delay results show that the median response time of the system under extremely low load is 120.8 milliseconds, and the median values under higher load and extremely high load increase to 450.5 milliseconds and 790.2 milliseconds, respectively, with significant differences, indicating that data concurrency pressure poses a challenge to the response rate of the early warning module. However, after optimization, a reasonable delay distribution range is still maintained, and the abnormal points are controlled within the statistical normal range. The platform has a certain real-time processing elasticity and performance stability. The comprehensive results show that the platform has strong operational adaptability while ensuring accuracy. The system applies a sliding window mechanism to dynamically adjust the warning threshold, adaptively correct the judgment boundary according to real-time statistical characteristics, and improve the response accuracy to changes in indicator time series.

V. E. User Interaction Experience

To comprehensively evaluate and optimize the performance of the platform in terms of interactive design, this section applies a data analysis framework that integrates subjective cognition and objective measurement, covering two levels: user satisfaction survey and operation behavior monitoring. The satisfaction evaluation focuses on seven interactive dimensions, such as interface aesthetics, chart clarity, and navigation logic, and collects technical and management personnel's evaluation of system functions and interactive fluency; the operational efficiency evaluation selects six typical task operations and reflects the system's responsiveness and interface fluency by recording the task completion time. This design covers both the user's perception level and the performance in the operation chain, which helps to judge the platform's interactive capabilities from multiple angles. Figure 7 contains a radar chart of the distribution of two types of user satisfaction on multi-dimensional attributes and a bar chart of the average time spent on operation tasks, showing the performance differences of the platform's interactive mechanism in different usage scenarios.



(a) Subjective ratings of interactive experience (b) Comparison of task time consumption

Figure 7: Comparative analysis of subjective ratings of platform interactive experience and task time consumption

From the perspective of satisfaction, managers score significantly higher than technicians in the two dimensions of operational convenience and navigation logic, 4.4 and 4.2, respectively, while the corresponding scores of technicians are only 3.9 and 3.6. This difference reflects that managers have a more positive perception of the simplicity of use paths, probably because their typical tasks are more fixed, and the operation process is more familiar. The technicians score 4.2 on the clarity of the chart, higher than the 4.1 of the managers. Combined with their attention to graphic details and data accuracy, it can be seen that they have higher requirements for the quality of chart expression. In terms of warning comprehensibility, the scores of both types of users are at a relatively medium level, 3.5 and 4.1, respectively, indicating that there is still room for improvement in the semantic expression and prompt strategy of this function. The task time consumption data shows that the average time consumption of the comparative analysis task is 4.6 seconds, which is much higher than the 1.9 seconds for chart switching and the 2.1 seconds for data export. This shows that the task involves data integration and chart

generation; the logical chain is complex, and the system calculation is large, which is the bottleneck link affecting the overall response experience. In contrast, the operation time of indicator query and module switching is 2.8 seconds and 3.0 seconds, respectively, indicating that the system maintains a relatively stable response efficiency under medium-complexity tasks. The comprehensive analysis results show that the platform's improvements in task structure clarity and response optimization strategies have a direct driving effect on the improvement of the interactive experience of different user groups.

VI. Conclusions

This paper proposes an optimization method for the power index benchmarking evaluation platform based on cloud computing and big data technology. By building a system architecture that integrates a standardized index system and an intelligent benchmarking algorithm, Kafka and Flume are integrated to achieve real-time access to multi-source heterogeneous data. Hadoop and Spark are used to complete data preprocessing and modeling. A unified index warehouse is established by combining HBase and Hive, and a dynamic scoring model based on unsupervised clustering and entropy weight method is designed. The experimental results show that the response time of the optimized platform is reduced from 329 seconds to 51.5 seconds under the scale of 100GB data. The accuracy of index benchmarking for industrial park scenarios reaches 86.4%, and the accuracy of voltage undervoltage anomaly detection is increased by 14.7 percentage points to 94.8%. The system has a balanced response delay distribution at each stage under the scale of 1 million data, and the average time consumption of the preprocessing and modeling stages is 420 milliseconds and 500 milliseconds, respectively. The shortcomings of the research are that the warning response delay still fluctuates in some extreme load scenarios, and the dynamic update mechanism of the index system has not yet fully adapted to high-frequency business changes. Future work needs to further optimize the resource scheduling strategy of the model under high concurrent load, explore incremental learning mechanisms to improve the dynamic adaptability of the indicator system, and expand multimodal data fusion capabilities to meet the complex analysis needs of new power systems.

References

[1] Omitaomu, O. A., & Niu, H. (2021). Artificial intelligence techniques in smart grid: A survey. Smart Cities, 4(2), 548-568.

[2] Gupta, R., & Chaturvedi, K. T. (2023). Adaptive energy management of big data analytics in smart grids. Energies, 16(16), 6016-6032.

[3] Tong, W., Liu, X., Wang, G., Chen, Z., & Peng, Z. (2025). Optimization of Power Indicator Benchmarking Assessment Based on Cloud Computing and Big Data Technology. J. COMBIN. MATH. COMBIN. COMPUT, 127(1), 6449-6469.

[4] Kwilinski, A., Lyulyov, O., Dzwigol, H., Vakulenko, I., & Pimonenko, T. (2022). Integrative smart grids' assessment system. Energies, 15(2), 545-563.

[5] Fragkos, G., Johnson, J., & Tsiropoulou, E. E. (2022). Dynamic role-based access control policy for smart grid applications: an offline deep reinforcement learning approach. IEEE Transactions on Human-Machine Systems, 52(4), 761-773.

[6] Maizana, D., & Putri, S. M. (2022). Appropriateness analysis of implementing a smart grid system in campus buildings using the fuzzy method. International Journal of Power Electronics and Drive Systems, 13(2), 873-883.

[7] Butt, O. M., Zulqarnain, M., & Butt, T. M. (2021). Recent advancement in smart grid technology: Future prospects in the electrical power network. Ain Shams Engineering Journal, 12(1), 687-695.

[8] Dicorato, M., Tricarico, G., Forte, G., & Marasciuolo, F. (2021). Technical indicators for the comparison of power network development in scenario evaluations. Energies, 14(14), 4179-4204.

[9] Wang, H., Wang, B., Luo, P., Ma, F., Zhou, Y., & Mohamed, M. A. (2021). State evaluation based on feature identification of measurement data: for resilient power system. CSEE Journal of Power and Energy Systems, 8(4), 983-992.

[10] Zhao, Z., Chen, Y., Liu, J., Cheng, Y., Tang, C., & Yao, C. (2022). Evaluation of operating state for smart electricity meters based on transformer–encoder–BiLSTM. IEEE Transactions on Industrial Informatics, 19(3), 2409-2420.

[11] Liu, W., Lei, P., Xu, D., & Zhu, X. (2023). Anomaly recognition, diagnosis and prediction of massive data flow based on time-GAN and DBSCAN for power dispatching automation system. Processes, 11(9), 2782-2794.

[12] Bin Mofidul, R., Alam, M. M., Rahman, M. H., & Jang, Y. M. (2022). Real-time energy data acquisition, anomaly detection, and monitoring system: Implementation of a secured, robust, and integrated global IIoT infrastructure with edge and cloud AI. Sensors, 22(22), 8980-9003.

[13] Arcas, G. I., Cioara, T., Anghel, I., Lazea, D., & Hangan, A. (2024). Edge offloading in smart grid. Smart Cities, 7(1), 680-711.

[14] El-Balka, R. M., Saleh, A. I., Abdullah, A. A., & Sakr, N. (2022). Enhancing the performance of smart electrical grids using data mining and fuzzy inference engine. Multimedia Tools and Applications, 81(23), 33017-33049.

[15] Chen, B., & Ge, W. (2024). Design and optimization strategy of electricity marketing information system supported by cloud computing platform. Energy Informatics, 7(1), 67-87.

[16] Gao, R., Xia, Y., Dai, L., Sun, Z., & Zhan, Y. (2022). Design and implementation of data-driven predictive cloud control system. Journal of Systems Engineering and Electronics, 33(6), 1258-1268.

[17] Neranjan Thilakarathne, N., Mohan, K. K., Surekha, L., & Hussain, A. (2021). Smart Grid: A Survey of Architectural Elements. Machine Learning and Deep Learning Applications and Future Directions. J. Intell. Syst. Internet Things, 3(1), 32-42.

[18] Shwe, T., & Aritsugi, M. (2024). Optimizing data processing: a comparative study of big data platforms in edge, fog, and cloud layers. Applied Sciences, 14(1), 452-475.

[19] Yıldırım, F., Yalman, Y., Bayındır, K. Ç., & Terciyanlı, E. (2025). Comprehensive Review of Edge Computing for Power Systems: State of the Art, Architecture, and Applications. Applied Sciences, 15(8), 2076-3417.

[20] Liu, S., Teng, Y., Cheng, S., Xu, N., Sun, P., Zhang, K., & Chen, Z. (2024). A cloud-edge cooperative scheduling model and its optimization method for regional multi-energy systems. Frontiers in Energy Research, 12(a), 1372612-1372628.
[21] Liang, Y., Dong, H., Gao, Y., Hu, L., Gao, Y., Lin, Z., et al. (2024). A comprehensive evaluation index system for low-carbon development of power systems in a load-intensive city. Frontiers in Energy Research, 12(1), 1453754-1453770.
[22] Deng, D., Li, C., Zu, Y., Liu, L. Y. J., Zhang, J., & Wen, S. (2022). A systematic literature review on performance evaluation of power system from the perspective of sustainability. Frontiers in Environmental Science, 10(1), 925332-925352.
[23] Liu, M., Zhao, Y., Zhu, L., Chen, Q., & Chang, D. (2024). Co-evaluation of power system frequency performance and operational reliability considering the frequency regulation capability of wind power. Frontiers in Energy Research, 11(1), 1334565-1334579.
[24] Wang, J., Ouyang, R., Wen, W., Wan, X., Wang, W., Tolba, A., & Zhang, X. (2023). A post-evaluation system for smart grids based on microservice framework and big data analysis. Electronics, 12(7), 1647-1666.
[25] Singhal, S., Athithan, S., Alomar, M. A., Kumar, R., Sharma, B., Srivastava, G., & Lin, J. C. W. (2023). Energy aware load balancing framework for smart grid using cloud and fog computing. Sensors, 23(7), 3488-3509.
[26] Mostafa, N., Ramadan, H. S. M., & Elfarouk, O. (2022). Renewable energy management in smart grids by using big data analytics and machine learning. Machine Learning with Applications, 9(1), 100363-100375.
[27] Xu, S., Xue, Y., & Chang, L. (2021). Review of power system support functions for inverter-based distributed energy resources-standards, control algorithms, and trends. IEEE open journal of Power electronics, 2(1), 88-105.
[28] Zhang, C., Lasaulce, S., Hennebel, M., Saludjian, L., Panciatici, P., & Poor, H. V. (2021). Decision-making oriented clustering: Application to pricing and power consumption scheduling. Applied Energy, 297(1), 117106-117119.
[29] Zubiria, A., Menéndez, Á., Grande, H. J., Meneses, P., & Fernández, G. (2022). Multi-criteria decision-making problem for energy storage technology selection for different grid applications. Energies, 15(20), 7612-7637.
[30] Al-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Paw, J. K. S., & Singh, M. J. (2023). Big data analytics using cloud computing based frameworks for power management systems: Status, constraints, and future recommendations. Sensors, 23(6), 2952-2989.
[31] AL-Jumaili, A. H. A., Mashhadany, Y. I. A., Sulaiman, R., & Alyasseri, Z. A. A. (2021). A conceptual and systematics for intelligent power management system-based cloud computing: Prospects, and challenges. Applied Sciences, 11(21), 9820-9861.
[32] Zhang, S., Pandey, A., Luo, X., Powell, M., Banerji, R., Fan, L., et al. (2022). Practical adoption of cloud computing in power systems—Drivers, challenges, guidance, and real-world use cases. IEEE Transactions on Smart Grid, 13(3), 2390-2411.
[33] Chuang, X., Li, L., Zhu, L., Wei, M., Qiu, Y., & Xin, Y. (2025). The design of a real-time monitoring and intelligent optimization data analysis framework for power plant production systems by 5G networks. Energy Informatics, 8(1), 1-27.
[34] Qi, D., Xi, X., Tang, Y., Zheng, Y., & Guo, Z. (2024). Real-time scheduling of power grid digital twin tasks in cloud via deep reinforcement learning. Journal of Cloud Computing, 13(1), 121-133.
[35] Dancheva, T., Alonso, U., & Barton, M. (2024). Cloud benchmarking and performance analysis of an HPC application in Amazon EC2. Cluster Computing, 27(2), 2273-2290.