International Journal of Advanced Engineering and Technology Research
ISSN: 3106-8677 (Print), ISSN: 3106-8685 (Online) | Vol. 4, No. 1, 2026

Research on Multi-Source Heterogeneous Data Processing Technology Based on Machine Learning

Ying Dong, Qingze Huo, Hui Wang *

CATARC Information Technology (Tianjin) Co., Ltd., China

* **Corresponding author**: (Email: wanghui2022@catarc.ac.cn)

Abstract: Given that multi-source heterogeneous data is scattered, mixed in format, and full of noise and missing parts, making traditional methods struggle to fully tap into its value, this article explores machine learning-based techniques for handling such data. First, break down the basic characteristics of multi-source heterogeneous data, and categorize the data sources and types; then build a complete data quality management process covering gaps filling, outlier detection and handling, normalization, and feature dimensionality reduction and selection; finally, study data fusion strategies at different levels and compare how machine learning models like logistic regression and XGBoost perform on heterogeneous datasets. The experiment results show that after a complete treatment process, the XGBoost model performs better in accuracy, F1 score, AUC, and other metrics, and has stronger robustness against noise and abnormal data. This article analyzes the limitations of existing technologies and looks ahead to future developments in this field, which can provide technical references for analyzing multi-source heterogeneous data in industrial scenarios.

Keywords: Machine learning, Multi-source heterogeneous data, Data governance, Feature fusion, Ensemble learning.

1. Introduction

Digital transformation has completely changed the way data is generated. Industrial sensors and SCADA systems continuously produce high-frequency time-series measurements, business systems store structured ledgers and records, and the internet and mobile devices contribute unstructured content like text, images, and audio/video. Weather, geography, and environmental monitoring provide external spatial and temporal observations. This data comes from different sources, devices, and protocols, with significant differences in format, meaning, granularity, quality, and units, together forming a typical multi-source heterogeneous data environment.

Multiple heterogeneous data sources can show a full picture that a single data source can't capture. For example, in equipment condition assessment: vibration, temperature, and current reflect the physical state; maintenance work orders reflect operational behavior; environment and load reflect external conditions. Only by cross-verifying these three types can you reliably judge health. Relying on just one source can easily lead to incomplete information, a narrow perspective, and high error rates. Data fusion removes redundancy and complements gaps, making the overall information greater than the sum of its parts. Handling multiple sources is way more complex than dealing with homogeneous ones: data quality varies, and missing, abnormal, duplicate, and inconsistent data are common; heterogeneous features can't be directly compared due to differences in scale, distribution, and meaning, so they need to be standardized; as sources increase, dimensions blow up and redundant noise accumulates; and aligning data across sources in time, space, and granularity is tricky. So, you need a systematic approach covering collection, management, fusion, and modeling instead of just relying on a single algorithm.

Machine learning provides powerful tools for data processing. For example, to address the issue that traditional fault diagnosis relies on a single data source, which leads to high probability calibration bias and low concentration index, reference [1] proposed a software system fault diagnosis method based on machine learning and log data. It uses graph neural networks to integrate structured, semi-structured, and unstructured logs, explicitly modeling their connections; and designs a multi-scale convolution model that takes session-level fused vectors as input to capture long-range dependencies. Experiments with the Spring Cloud distributed e-commerce microservices cluster show that the proposed method has a probability calibration bias closest to 0, a concentration index in the optimal range of 0.5–0.7, and excellent and robust diagnostic performance. For example, as mentioned in reference [2], traditional dam risk assessments rely on expert experience and have low transparency, so they built an evaluation model that combines multi-source monitoring with machine learning. We built a four-dimensional evaluation system covering structure, hydrology, geology, and human factors, combined with relative entropy and climate factor adjustments; tuned parameters using the beluga whale algorithm, integrated satellite and ground sensor data, and used adaptive weights to coordinate multi-source data. In practice, the comprehensive performance for flood season warnings reached 91.5% (14.2%), with an accuracy of 93.8%, recall of 89.2%, false alarm rate of 9.4%, and identification rate for extreme risks at 26.5%, improving both precision and efficiency, offering a new approach for smart water management monitoring. Machine learning enhanced the system's decision-making capability when handling multi-source heterogeneous data.

From this, we can see that systematically studying machine learning-based multi-source heterogeneous data processing technology is theoretically significant and practically valuable for improving data use efficiency and supporting smart decision-making in industries.

2. Description of Multi-source Heterogeneous Data

2.1. Characteristics of Multi-source Heterogeneous Data

Multi-source heterogeneous data is generated by multiple independent data sources, with differences in structure and semantics, all describing the same object or system [3-4]. Multi-source means having diverse sources; heterogeneity includes structural and semantic heterogeneity: structural heterogeneity refers to different organizational forms such as relational tables, key-value pairs, time series, document trees, and image matrices; semantic heterogeneity means the same concept is inconsistent in naming, coding, units, and values, like different manufacturers coding the same state variable differently. Compared to a single data source, it has four features: diverse origins, spanning perception layers, business layers, and the external environment; heterogeneous formats, with structured, semi-structured, and unstructured data coexisting; variable quality, with notable differences in missing data, noise, anomalies, and timeliness, requiring quality management; and both complementarity and redundancy coexist, with integration aiming to utilize complementarity while suppressing redundancy. Understanding these features is the prerequisite for choosing a processing strategy [3-4].

2.2. Data Sources and Type Classification

Based on how structured they are, multi-source heterogeneous data can be divided into three types: structured data, like relational tables and CSVs, have fixed formats, clear meanings, and are easy to parse—think equipment ledgers or transaction records; semi-structured data, like JSON, XML, or logs, come with self-describing tags but no fixed format, such as IoT messages or operational logs; unstructured data, like text, images, audio, and video, have no standard format and need NLP or computer vision to extract meaning, like work orders, photos, or surveillance videos. Based on where they come from, they can be divided into perception, business, and external sources. Sensing sources come from sensors and measurement terminals, are high-frequency, continuous, and highly real-time, like SCADA measurements and vibration monitoring. Business sources come from enterprise information systems, are low-frequency, discrete, and have strong business semantics, like ERP ledgers and maintenance work orders. External sources come from outside the organization, like weather, geographic information, and public statistics. These three types differ significantly in time scale, update frequency, and reliability, and aligning and integrating them is one of the main challenges in multi-source processing.

3. Data Quality Governance

3.1. Missing Value Handling

Missing values are the most common quality issue in cross-source heterogeneous data, caused by things like sensor failures, communication interruptions, inconsistent collection frequencies, and human oversight. The missing patterns can be completely random, random, or non-random, and the way you handle them should match the missing pattern and how you plan to use the data. There are two methods: deletion and imputation. Deletion removes samples or features with missing values. It works well when the missing rate is low and there's plenty of data, but it loses information and can introduce bias if the missing data is related to the target variable. Imputation methods use existing information to estimate missing values: statistical imputation fills in with the mean, median, or mode; the median is robust to outliers and works well for low missing rates. Interpolation targets time series data, using linear, Lagrange, or spline interpolation to maintain sequence continuity with neighboring points. In engineering, you need to balance accuracy and cost: statistical imputation is simple but ignores feature relationships; model-based imputation is more accurate but computationally heavy and carries the risk of training bias. For time-series data, interpolation methods are preferred to preserve dynamic features; for tabular data, you can compromise between statistical and model-based imputation depending on the scale, and also do sensitivity analysis.

3.2. Outlier Detection and Handling

Outliers are observations that clearly deviate from the normal data distribution, and they can be caused by sensor failures, communication errors, human tampering, or real unusual business events. Anomaly detection is a key part of managing multi-source heterogeneous data. Mixing abnormal and normal data can distort statistical features, mess with model training, and lead to biased decisions. The interquartile range (IQR) is a classic robust outlier detection method. It uses the difference between the upper and lower quartiles to create a decision interval for identifying outliers [5-7]. The formula for calculating the interquartile range (IQR) is shown in formula (1).

$$IQR = Q_3 - Q_1 \tag{1}$$

In formula (1), Q_3 is the upper quartile and Q_1 is the lower quartile. If a sample value falls outside the $[Q_1 - 1.5IQR, Q_3 + 1.5IQR]$ range, it is considered an outlier. The IQR method is based on positional statistics, is not sensitive to the shape of the distribution or extreme values, and is robust, but it can easily produce more false positives for skewed data.

3.3. Data Normalization and Transformation

Features from multi-source heterogeneous data often have different units and magnitudes. For example, current is measured in amperes, temperature in Celsius, and load in kilowatts. If you feed these directly into models sensitive to scale, features with larger units will dominate distance calculations and gradient updates, hurting model performance. The goal of data normalization is to eliminate the impact of units so that all features are on a comparable scale.

If the data roughly follows a normal distribution, you can use zero-mean normalization (Z-score normalization) to transform the features into a standard normal distribution with zero mean and unit variance. The mathematical expression for zero-mean normalization is shown in formula (2).

$$d^{'} = (\frac{d - d_{\mu}}{d_{\sigma}}) \tag{2}$$

In formula (2), d_{μ} and d_{σ} represent the mean and standard deviation of the feature, respectively. The standardized feature is not affected by the original units and retains the relative structure of the original distribution, making it suitable for data where the distribution is roughly

symmetric and outliers have been handled.

If the data doesn't follow a normal distribution, use Min-Max normalization. Min-Max normalization linearly maps features to the range between 0 and 1. The mathematical expression for Min-Max normalization is shown in formula (3).

$$d' = \frac{d - d_{\min}}{d_{\max} - d_{\min}} \tag{3}$$

In formula (3), $d_{\max}$ and $d_{\min}$ are the minimum and maximum values of the feature. Min-Max normalization is sensitive to the range of values, so when there are extreme values, it can easily get stretched or compressed. Usually, it's used together with outlier handling. For data with outliers, you can use robust scaling based on the median and interquartile range (IQR): transform the data using the median as the center and IQR as the scale to reduce the impact of outliers on the scaling parameters. In practice, you should choose the method based on the data distribution to improve feature scaling stability and model performance.

3.4. Feature Selection and Dimensionality Reduction

After fusing multiple sources, the feature dimensions multiply, increasing redundant and noisy features, which not only adds computational cost but also makes overfitting more likely. Feature selection and dimensionality reduction are key ways to improve the quality of fused data. The Pearson correlation coefficient can describe the linear correlation between two continuous variables and is often used to identify and remove highly correlated features, thereby reducing redundancy, simplifying the model, and improving generalization [8-9]. By combining methods like variance filtering, mutual information, or principal component analysis, you can effectively reduce dimensionality while keeping the key information. Its mathematical expression is defined in formula (4).

$$R = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \cdot \sqrt{\sum_i (y_i - \bar{y})^2}} \tag{4}$$

In formula (4), $\bar{x}$ and $\bar{y}$ are the means of two variables, and R ranges from -1 to 1. The closer the absolute value is to 1, the stronger the linear correlation. Correlation analysis can identify highly redundant feature pairs and also screen features that are strongly related to the target, making it an important basis for feature selection. Feature selection can be divided into three types: filter methods, which are independent of learning algorithms and rank features based on statistical measures like Pearson correlation, chi-square, or mutual information—these are efficient and good for initial screening in high-dimensional data; wrapper methods, which search for the optimal subset based on the performance of a learner, like recursive feature elimination—they are more accurate but costly; and embedded methods, which automatically select features during training, like L1 regularization or tree-based feature importance—these balance accuracy and efficiency.

4. Data Processing Methods Based on Machine Learning

4.1. Integration Levels and Strategies

Data fusion can be divided into data-level, feature-level, and decision-level, with increasing abstraction and decreasing granularity. Data-level fusion aligns and combines raw data, keeping the information complete but requiring precise alignment, which works well for homogeneous sensors. Feature-level methods first extract features and then concatenate, weight, or fuse them as tensors. Early fusion falls into this category, balancing compression and efficiency, and is the most commonly used in engineering. The key is feature alignment and redundancy suppression. Decision-level methods model each source separately and then combine them through voting, weighting, or stacking. They are the most abstract and are more forgiving of single-source quality, but it's hard to leverage deep cross-source relationships. In deep learning, fusion is divided into early, mid, and late stages; mid-level and attention-based dynamic weighting fusion are becoming mainstream.

4.2. Classical Machine Learning Model

Logistic regression (LR) uses a logistic function to map a linear combination to a probability, estimating parameters with maximum likelihood. It requires a high degree of linear separability, but it's efficient to train and highly interpretable, making it a common baseline for binary classification and risk assessment. Support Vector Machines (SVM) map to high dimensions through kernel functions and find the maximum margin hyperplane, making them suitable for small to medium-scale nonlinear classification, but their training complexity grows quickly with more samples. Random Forests (RF) build multiple decision trees using bootstrap sampling and random feature subsets, then combine them through voting or averaging. The generalization error converges as the number of trees increases, making them robust to noise and overfitting, and they can also output feature importance. XGBoost is an efficient implementation of gradient boosting decision trees, adding a regularization term for model complexity to the loss function [10-11]. The mathematical expression of the objective function is shown in formula (5).

$$L = \sum_i l(y_i, \overline{y_i}) + \sum_k \Omega(f_k) \tag{5}$$

In formula (5), the first term is the empirical loss and the second term is the regularization term. XGBoost approximates the objective function using a second-order Taylor expansion and introduces a sparsity-aware algorithm to automatically learn the optimal split direction for missing values. It works well with data that has missing and sparse features and performs excellently in competitions and industrial scenarios.

Different models adapt differently to multi-source heterogeneous data: linear models are sensitive to feature scales and noise, require careful tuning, but they train fast and are easy to interpret; tree models aren't sensitive to scales and can automatically handle feature interactions, but they tend to overfit in high-dimensional sparse settings; kernel methods are quite accurate on medium-sized data, but their scalability is limited. In practice, multiple model candidate pools are often built in parallel, and then the best ones are selected or combined based on overall evaluation results to balance

accuracy, efficiency, and robustness.

5. Experimental Design and Results Analysis

5.1. Evaluation Metrics

The experiment uses five metrics—accuracy, precision, recall, F1 score, and AUC—to evaluate classification performance. Suppose TP, FP, FN, and TN represent true positives, false positives, false negatives, and true negatives, respectively. Then, accuracy measures overall correctness, precision measures how accurate the positive predictions are, recall measures how well the positive samples are covered, and F1 is the harmonic mean of precision and recall. The mathematical formulas for accuracy, precision, recall, and F1 are given in equations (6)–(9).

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F_1 = \frac{2P \cdot R}{P + R} \tag{9}$$

AUC is the area under the receiver operating characteristic curve and reflects how well the model can rank positive and negative samples—the closer it is to 1, the stronger the discriminative ability. In this experiment, since it's a binary classification with fairly balanced classes, we mainly use F1 and AUC as our evaluation metrics to measure overall precision and recall as well as ranking performance, while accuracy is just a straightforward supplementary reference.

5.2. Analysis of Data Governance Effectiveness

To check how well the governance pipeline works, we ran models with both logistic regression and XGBoost on early fused features under the full governance process. The results are shown in Table 1.

Table 1. Performance Comparison of Data Governance Models

Model	Accuracy	Precision	Recall rate	F1	AUC
Logistic Regression	90%	92%	92%	92%	95%
XGBoost	94%	95%	97%	96%	99%

From the performance comparison results of the data governance models in Table 1, we can see that on multi-source heterogeneous datasets, XGBoost outperforms the logistic regression model across accuracy, precision, recall, F1 score, and AUC. Logistic regression is quite sensitive to feature scale and can be easily affected by data noise and outliers. Its evaluation metrics are generally low, and it has performance limitations with complex heterogeneous data. The XGBoost ensemble tree model is more robust to disturbances and can uncover nonlinear relationships in heterogeneous data. Compared to logistic regression, it improves accuracy by 4 percentage points, F1 score by 4 points, and reaches an AUC of 99%. Experiments show that for multi-source heterogeneous data with noise, XGBoost adapts better and provides more effective modeling, significantly reducing the negative impact of data flaws and offering a solid model foundation for future multi-source data integration analysis.

6. Future Development Directions

Future research can go deeper in five directions. First is integrating privacy protection, by introducing federated learning, secure multi-party computation, and homomorphic encryption to enable cross-institution collaborative modeling without moving data, balancing privacy and utility. Second is multi-modal foundational models, drawing from the big model pre-training–fine-tuning approach to build unified representation models for multi-source heterogeneous data, reducing the cost of task customization. This trend is already showing up in medical multi-modal intelligent diagnosis and treatment. Third is enhancing knowledge integration by embedding domain knowledge and physical laws into fusion models to improve cross-source consistency and physical interpretability. Fourth is automated data governance and modeling, leveraging automated machine learning to achieve joint optimization of anomaly detection, feature engineering, and model selection, lowering the engineering barriers. Fifth is a trustworthy evaluation system, establishing a multidimensional evaluation framework covering data quality, model performance, privacy risks, and fairness, supporting the high-quality implementation of fusion systems. These directions support each other and are expected to push multi-source heterogeneous data fusion from just being usable to being trustworthy, interpretable, and scalable.

References

[1] Wang, Y., Ren, Y., Jia, F., et al. (2026). Software system fault diagnosis based on machine learning and log data. Electronic Design Engineering, 34(20), 69–72. https://doi.org/10.14022/j.issn1674-6236.2026.20.012

[2] Yu, Y., Wang, T., & Jiang, X. (2026). Research on model optimization for dam safety risk analysis based on multi-source data fusion and machine learning. Zhihuai, (9), 7–9.

[3] Hao, Q., Dong, Q., Zhi, Y., & Zhang, F. (2025). A global automotive industry multi-source heterogeneous data cleaning platform based on machine learning. In 2025 4th International Conference on Artificial Intelligence and Computer Information Technology (AICIT) (pp. 1–6). Yichang, China. https://doi.org/10.1109/AICIT65974.2025.11282740

[4] Rui, S., Xiaoxia, L., & Fan, Z. (2025). Research and application of digital marketing operation platform based on multi-source heterogeneous data. In 2025 IEEE 3rd International Conference on Sensors, Electronics and Computer Engineering (ICSECE) (pp. 1316–1320). Jinzhou, China. https://doi.org/10.1109/ICSECE65727.2025.11256890

[5] Zhihu. (n.d.). Retrieved from https://zhuanlan.zhihu.com/p/716220715

[6] Liu, Y. (2026). A control method for pipe blank centering and drilling based on quartile filtering and improved least squares method. Automation Application, 67(10), 58–61. https://doi.org/10.19769/j.zdhy.2026.10.016

[7] Jin, Q., Heng, F., Liu, C., et al. (2026). Application of statistical interquartile range in microseismic interpretation. In Proceedings of the 4th China Petroleum Geophysical Prospecting Academic Conference (Volume 2) (pp. 1084–1086). China Petroleum Society, Geophysical Prospecting

Professional Committee. https://doi.org/10.26914/c.cnkihy.2026.009367

[8] Zhu, W., & Liu, Z. (2026). Energy consumption features and prediction based on Pearson algorithm and random forest algorithm: A case study of municipal hospitals in Shanghai. Green Building, 18(5), 70–74+93.

[9] Liu, A., & Feng, X. (2026). Detection of multiplicative false data injection attacks based on Pearson correlation coefficient. Journal of Shanghai Dianji University, 29(4), 200–206.

[10] Zhao, T., Lin, X., Qian, J., et al. (2026). Analysis of the driving mechanisms and spatial heterogeneity of carbon emission intensity in China's crop production based on XGBoost. Environmental Science. Advance online publication. https://doi.org/10.13227/j.hjkx.202605279

[11] Huang, S., Liu, Y., Zhang, Y., et al. (2026). A gas path fault diagnosis method for aircraft engines based on VMD-XGBoost. Sensors & Microsystem, 45(10), 60–65. https://doi.org/10.13873/J.1000-9787(2026)10-0060-06