IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131161

Intelligent Data Quality Monitoring in ETL Pipelines Using Machine Learning

Shraddha Narhari Shejul[1], Dhanashree Santosh Suryawanshi[2], Almas Sikander Sayyed[3], Payal S. Ulhare[4]

Department of Computer Applications Engg, Adsuls Technical Campus, Chas Ahilyanagar, India[1-4]

Abstract: ETL (Extract, Transform, and Load) workflows bring together information from different systems so that it can be used for reporting, analytics, and decision-making. Errors such as incomplete fields, repeated records, invalid values, and unexpected changes in data patterns can reduce the trustworthiness of the information delivered to downstream applications. Conventional quality controls generally use predefined rules and thresholds. These controls are useful for known requirements, but they may not identify unusual combinations of values or changes that were not anticipated when the rules were written. This paper outlines a hybrid monitoring approach that combines deterministic validation, statistical data profiling, and unsupervised anomaly detection. The proposed design collects quality indicators at selected ETL stages and uses an Isolation Forest model to identify records or batches that differ from an established reference pattern. A controlled evaluation procedure is also described in which known defects are introduced into test data and detection performance is measured using precision, recall, F1-score, false-positive rate, and processing time. This manuscript describes a research proposal and evaluation plan; it does not report completed experiments or measured outcomes. The framework is intended to provide a practical basis for studying whether machine-learning signals can add useful coverage to conventional data-quality checks.

Keywords: ETL, data quality, anomaly detection, machine learning, Isolation Forest, data profiling, data engineering.

I. INTRODUCTION

Data pipelines support many activities, including business reporting, analytical processing, scientific work, and machine-learning applications. In an ETL workflow, information is collected from one or more source systems, converted into a required structure, and delivered to a destination such as a database or data warehouse. Problems can arise at any point in this sequence. For example, a source application may begin supplying incomplete fields, a transformation may introduce incorrect values, or a source-system update may change the structure or distribution of incoming data.

Data quality has several related dimensions. Completeness describes whether required information is present. Uniqueness concerns repeated records or identifiers. Validity measures whether values follow expected formats and permitted ranges. Consistency considers whether related fields agree with one another. Distribution stability examines whether the current data resembles an established historical pattern. Since these dimensions describe different types of problems, relying on a single validation technique may leave some defects undetected.

This study proposes a monitoring design that combines rule-based checks with statistical profiling and an unsupervised anomaly detector. Explicit rules are retained for requirements that can be stated precisely, while statistical and machine-learning methods provide additional signals about unusual observations. The central research question is whether this combined approach can detect more of the deliberately introduced data defects than rule-based validation alone, without producing an impractical number of false alerts or excessive processing overhead.

II. RELATED WORK

Prior work on data quality in ETL environments describes the difficulty of integrating information from heterogeneous sources and the importance of monitoring quality throughout the integration process [1]. This perspective supports checking data during pipeline execution rather than waiting until records have reached their final destination.

Isolation Forest, introduced by Liu, Ting, and Zhou, identifies unusual observations by repeatedly selecting features and split points to isolate individual records [2]. Because the method does not require labelled examples for every possible defect, it is a candidate for settings where historical fault labels are limited. Its output must nevertheless be interpreted

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131161

carefully: an anomaly score indicates unusual behaviour, not necessarily an incorrect record. Legitimate but uncommon observations can also receive high anomaly scores.

Research on data-quality needs in machine-learning development pipelines discusses quality concerns across different stages of the development lifecycle [3]. Other work on data-pipeline quality considers data-related, processing, infrastructure, and lifecycle factors that can contribute to pipeline failures [4]. Together, these studies support stage-aware monitoring. This paper does not claim that combining validation rules with anomaly detection is inherently new. Its focus is the specification of an implementable workflow and a controlled method for comparing the combined approach with simpler baselines.

III. RESEARCH OBJECTIVES AND EXPECTED CONTRIBUTION

The work is guided by five objectives:

- Define measurable data-quality checks that can be applied at selected stages of an ETL workflow.
- Develop a prototype that combines deterministic validation with machine-learning-assisted anomaly detection.
- Evaluate the different detection approaches using the same controlled test cases.
- Study false alerts and detection performance separately for each injected fault category.
- Document a monitoring workflow that can be implemented and repeated on small or medium-sized structured datasets.

The expected contribution is a clearly specified framework and experimental procedure that can be implemented and assessed. Any statements about improved accuracy, scalability, operational benefit, or research novelty should be made only after the experiments have been completed and the findings have been compared with relevant published work.

IV. PROPOSED FRAMEWORK

The proposed architecture accepts structured data from CSV files or relational tables. The records pass through the normal extract, transform, and load stages. At selected points, a profiling component calculates quality indicators. A detection layer then combines explicit rule checks, statistical reference limits, and an Isolation Forest model. The resulting findings are stored for inspection and may be used to generate alerts or populate a review queue.

TABLE I MAIN STAGES IN THE PROPOSED DATA-QUALITY MONITORING DESIGN.

Stage	Main activity
Source data	CSV files or relational database tables
ETL processing	Extract, transform, and load records
Quality profiling	Inspect schema, missing values, duplicates, ranges, and distributions
Detection	Apply validation rules, statistical baselines, and Isolation Forest
Monitoring output	Record issues, calculate quality metrics, create alerts, and support review

A. Data ingestion and profiling

The prototype is designed to read structured records from a CSV source or a relational database. During ingestion, it records basic information about the dataset, including its schema, total row count, column data types, null counts, distinct-value counts, and summary statistics for numerical fields. Profiling should be repeated at selected pipeline stages. Comparing these profiles can help identify where a quality problem first appears.

B. Rule-based validation

Deterministic checks are used for requirements that can be expressed as explicit conditions. Examples include checking mandatory fields, verifying primary-key uniqueness, restricting categorical columns to approved values, confirming that dates can be parsed, rejecting negative transaction amounts where they are not permitted, and checking relationships

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131161

between linked tables. Rule violations should be retained separately from machine-learning anomaly scores. A record that breaks a known business rule is different from a record that is unusual but may still be valid.

C. Statistical reference checks

For numerical attributes, the interquartile range (IQR) can provide an initial method for identifying values that are far from the central portion of the observed data. Under the common 1.5 × IQR convention, values below Q1 − 1.5 × IQR or above Q3 + 1.5 × IQR are flagged for review. This multiplier is a starting point rather than a universal rule and should be assessed for the dataset being studied. Batch-level distribution checks may also compare current summaries with a historical reference by using an appropriate distance or drift statistic.

D. Machine-learning detection

The proposed anomaly-detection component uses Isolation Forest, trained on a reference dataset expected to contain predominantly acceptable records. Numerical variables must be prepared consistently before training. Categorical attributes may require encoding, and the treatment of missing values must be specified as part of preprocessing. Training, validation, and test data should remain separate to reduce the risk of information leakage. The contamination setting is a model parameter that requires justification and testing; it should not be treated as the actual proportion of faulty records unless that proportion is independently known. Model flags are intended to guide investigation, not to establish that a record is erroneous.

E. Alerting and traceability

Each detected issue should be associated with enough information to support investigation. Suggested fields include a pipeline run identifier, source name, ETL stage, failed rule or model score, event timestamp, and review status. A monitoring view may summarize the number of records examined, rule violations, anomaly flags, and overall quality measures. The framework should not automatically delete or overwrite source records based only on an anomaly score. Decisions about correction should involve an appropriate data owner or reviewer.

V. RESEARCH METHODOLOGY

A. Research design

The evaluation will use a controlled experiment. A clean reference dataset will be retained, and additional copies will be prepared with known defects introduced according to documented procedures. The source may be a public tabular dataset whose licence permits the intended use, or a synthetic customer or transaction dataset created for the prototype. The study should record the dataset name and source, version, licence, dimensions, and the precise method used to create each defect. If synthetic data is selected, conclusions should be limited to what that test setting can demonstrate.

B. Fault scenarios

TABLE II CANDIDATE FAULT TYPES FOR THE CONTROLLED EVALUATION. EXACT INJECTION RATES AND PROCEDURES MUST BE DOCUMENTED BEFORE TESTING.

Fault category	Example of controlled modification	Ground-truth information
Missing values	Replace selected non-key field values with nulls	Locations of altered cells or affected records
Duplicate data	Repeat selected rows or identifiers	Identifiers of duplicated records
Invalid domain value	Insert an unapproved category or out-of-range value	Records containing the inserted values
Numerical outlier	Change selected numerical values substantially	Records whose values were modified
Distribution shift	Change the value distribution in a selected batch	Affected batch and, where applicable, records

C. Experimental procedure

- Keep an unchanged copy of the selected dataset and document its schema.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131161

- Create reproducible training, validation, and test partitions, using fixed random seeds where applicable.
- Introduce documented defects into the test data. Candidate rates include 1%, 5%, and 10%, where these levels aremeaningful for the selected fault type.
- Run the rule-based checks, IQR method, and Isolation Forest under comparable test conditions.
- Compare each method's predictions with the known labels created during fault injection.
- Repeat experiments where randomness may affect the outcome, and report both average results and variability.
- Measure elapsed processing time and record relevant hardware and software versions.

D. Evaluation measures

For record-level binary classification, precision is calculated as TP/(TP+FP), recall as TP/(TP+FN), and the F1-score as 2 × precision × recall/(precision+recall). The false-positive rate is FP/(FP+TN). In these expressions, TP, FP, FN, and TN represent true positives, false positives, false negatives, and true negatives. The study must explicitly define an injected defective record as the positive class. Some issues, such as a distribution shift affecting an entire batch, are more appropriately assessed at batch level. Confusion matrices and results for individual fault categories should be reported alongside aggregate metrics.

VI. IMPLEMENTATION PLAN

A prototype suitable for a student research project can be developed using Python, Pandas, scikit-learn, and Streamlit. Pandas can support file loading, transformations, and data profiling. Deterministic validation rules can be implemented in Python, while scikit-learn can be used to train the Isolation Forest model and calculate evaluation metrics. Streamlit may be used to present quality summaries and records that require review. The software environment should be documented through a dependency or requirements file so that the implementation can be reproduced.

TABLE III SUGGESTED SOFTWARE COMPONENTS FOR THE PROTOTYPE.

Technology	Proposed role
Python and Pandas	Load structured data, perform transformations, profile fields, and prepare evaluation data
scikit-learn	Build the Isolation Forest model and calculate evaluation measures
SQLite or PostgreSQL (optional)	Store monitoring runs, issue history, or processed records
Streamlit (optional)	Display monitoring summaries and records flagged for review
Git and fixed random seeds	Track code changes and support repeatable experiments

VII. RESULTS AND REPORTING PLAN

No experimental measurements are presented in this manuscript because the proposed implementation and evaluation have not yet been completed. Once the tests have been run, the table below should be populated with observed values. Estimated or invented figures must not be substituted for measured results. The final report should also identify the test dataset, defect-injection rate, model configuration, decision threshold, and execution environment.

TABLE IV RESULTS TEMPLATE. REPLACE THE PLACEHOLDERS WITH VALUES OBTAINED FROM THE COMPLETED EXPERIMENTS.

Method	Precision	Recall	F1-score	FPR	Time (s)
Rule-based baseline	—	—	—	—	—

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131161

IQR baseline	—	—	—	—	—
Isolation Forest	—	—	—	—	—
Hybrid approach	—	—	—	—	—

The discussion should examine whether the combined approach detects fault types that the deterministic baseline misses, whether the additional detections come with more false positives, and whether the extra processing cost is acceptable. Results that are mixed or negative should be reported as they are. The hybrid system should not be described as better unless the measurements support that conclusion.

VIII. THREATS TO VALIDITY AND ETHICAL CONSIDERATIONS

Artificially introduced defects may not cover the full range of problems that occur in operational data systems. Synthetic datasets can make experiments easier to repeat, but findings from them may not transfer directly to production environments. Isolation Forest may flag valid but uncommon observations, and its output can be sensitive to the selected contamination parameter. Other risks include overlap between training and test data, fitting preprocessing steps on the test set, and allowing injected labels to influence model training. The implementation and report should describe how these risks are controlled.

Datasets made available to the public should be used in accordance with their licences. Personal or sensitive information should be avoided unless its use is justified and appropriate safeguards are in place. Monitoring alerts should normally support human assessment rather than trigger automatic deletion or replacement of source information. Software versions, model parameters, random seeds, and data-preparation steps should be recorded to make the study reproducible.

IX. CONCLUSION

This paper presents a proposed method for monitoring data quality in ETL pipelines by combining explicit validation rules, statistical profiling, and Isolation Forest-based anomaly detection. The framework is intended to examine data at multiple pipeline stages and to retain enough information for detected issues to be reviewed. Its research contribution will depend on implementing the design and comparing it with clearly defined baseline methods using controlled fault scenarios. Possible extensions include identifying schema changes, monitoring batch-level trends over time, and supporting root-cause investigation. Further evaluation on additional datasets and more realistic pipeline workloads is needed before making broad claims about effectiveness in operational environments.

REFERENCES

[1]. “Data quality in ETL process: A preliminary study,” *Procedia Computer Science*, vol. 159, pp. 676–687, 2019. doi:
[2]. 1016/j.procs.2019.09.223.
[3]. F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.
[4]. “A Survey of Data Quality Requirements That Matter in ML Development Pipelines,” *ACM Journal of Data and Information Quality*. doi:
[5]. 1145/3592616.
[6]. “Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers,” *Journal of Systems and Software*. doi: 10.1016/j.jss.2023.111855. SUBMISSION CHECKLIST
[7]. Add the full author names, institutional affiliation, and contact email.
[8]. Confirm each reference against the original publication and apply the required citation style.
[9]. Implement the proposed method and replace the results template with measured experimental findings.
[10]. Record the dataset name, source, licence, sample dimensions, model settings, and code availability.
[11]. Check the target venue's formatting rules, page limits, submission requirements, and publication policies.
[12]. Ensure all listed authors have reviewed and approved the final manuscript. Manuscript status: Revised research draft describing a proposed methodology. It is not yet a completed empirical study or a submission-ready paper. Author details, venue-specific formatting, implementation, measured results, and final reference verification remain outstanding.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License