

Explainability in the ETL Layer: Making Data Transformations Transparent and Traceable

Sougandhika Tera

Independent Researcher
Cohoes, New York
terasougandhika@gmail.com

Abstract:

Data transformation processes in Extract, Transform, Load (ETL) pipelines are crucial in creating the inputs to AI and analytics systems, despite the fact that they often operate as "black boxes" with little transparency. This paper presents Explainable ETL, a platform for transparent, traceable, and comprehensible data transformations. We explore the integration of lineage tracking, semantic annotations, and interpretability tools like as SHAP, LIME, and metadata graphs into ETL orchestration to enhance auditability, bias detection, and regulatory compliance. The suggested architecture's direct integration of explainability modules into ETL tools allows data engineers and business users to understand why data appears as it does at each stage of the pipeline. According to experimental results, explainable ETL reduces bias propagation by 65% and improves error traceability by 92%. This tactic encourages more accountability and trust in data-driven systems by bridging the gap between responsible AI and data engineering.

Keywords: Explainable ETL, Data Lineage, SHAP, LIME, Bias Detection, Data Transformation Transparency, Responsible AI, Data Governance, ETL Orchestration, Auditability.

1. INTRODUCTION

The rise of data-driven decision-making has made the quality, integrity, and fairness of data pipelines more crucial than ever. Although there has been a lot of focus on Explainable AI (XAI) for model interpretability, the earliest stages of data transformation—which are usually managed via Extract, Transform, Load (ETL) procedures—remain mostly unexplained. These transformations, which include cleaning, aggregation, encoding, and feature engineering, ultimately shape the data that feeds machine learning models and analytical dashboards. Without transparency, biases could inadvertently be introduced, errors could go undiscovered, and it might be challenging to demonstrate compliance with regulations like the GDPR and EU AI Act.

Current ETL systems focus throughput and scalability, with minimal native support for explainability, lineage tracking, or transformation interpretability. When issues with data quality or skewed findings arise, teams often engage in tedious manual detective effort to pinpoint specific transformation procedures. This lack of visibility undermines trust, complicates debugging, and impacts monitoring in regulated industries like healthcare, finance, and public sector analytics.

In order to close the critical gap between responsible AI and data engineering, this study suggests a methodical strategy for Explainable ETL (xETL). Our contributions include: (1) a conceptual model for integrating explainability into ETL pipelines using interpretability techniques adapted from XAI; (2) an architecture for xETL that combines metadata graphs, semantic annotations, and tools like SHAP and LIME for transformation-level explanations; (3) experimental validation showing improvements in error traceability, bias detection, and compliance reporting; and (4) practical suggestions for implementation in existing data stacks.

The remainder of this paper is organized as follows: Section 2 reviews related studies in ETL monitoring, XAI, and data lineage. Section 3 presents the xETL architecture and framework. Section 4 goes into great

length about implementation and evaluation. Section 5 discusses limitations and potential directions, while Section 6 provides conclusions.

2. BACKGROUND AND RELATED WORK

2.1. The Opacity of Traditional ETL

Transparency is not as important as efficiency in traditional ETL systems. Transformations are often implemented as procedural code (SQL, Python, Spark) or programmed in GUI-based tools with minimal explainability built in. Amundsen, DataHub, and Apache Atlas are examples of data lineage solutions that provide coarse-grained provenance tracking, but they do not provide fine-grained insights into the causes of a transformation's result. This opacity is problematic when:

Decisions about encoding, sampling, or aggregation introduce bias.

Errors spread through several stages of transformation.

Data derivation must be justified for compliance audits.

2.2. Explainable AI (XAI) Techniques

XAI methods such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are commonly used to interpret model predictions. SHAP assigns prediction contributions to input features using cooperative game theory, whereas LIME employs interpretable surrogate models to mimic model behavior locally. These methods have lately been extended beyond models to explain data pretreatment processes, even if these applications are still in their early stages.

2.3. Data Lineage and Metadata Management

Data lineage solutions don't address "why does it look like this?" but rather "where did this data come from?" record the data flow across systems. Although they provide context, semantic annotations and business glossaries are usually static and incompatible with transformation logic. The OpenLineage standard and MLflow for experiment tracking are positive milestones, but they do not fully solve transformation explainability.

2.4. Responsible AI and Governance Frameworks

Regulatory initiatives such as the EU AI Act and NIST AI Risk Management Framework emphasize the need for transparency across the entire AI lifecycle, including data preparation. Model Cards and Datasheets for Datasets promote documentation but are often decoupled from the dynamic ETL processes that shape data.

2.5. Related Work in Explainable Data Pipelines

Recent research has begun exploring explainability in data pipelines. Schelter et al. (2021) proposed declarative data quality constraints for monitoring. Polyzotis et al. (2019) introduced data debugging via slicing. However, these approaches focus on quality rather than interpretability. Our work extends these ideas by integrating XAI methods directly into ETL transformations.

3. THE EXPLAINABLE ETL (XETL) FRAMEWORK

3.1. Core Principles of xETL

Explainable ETL is built on four foundational principles:

1. **Transparency:** A clear rationale and an impact assessment should accompany every change.
2. **Traceability:** Fine-grained provenance should be used to link outputs to inputs and transformation algorithms.
3. **Auditability:** All transformations must be recorded with explanations in order to guarantee compliance and repeatability.
4. **Actionability:** With the aid of explanations, users should be able to spot bias, correct errors, and expedite procedures.

3.2. xETL Architecture

Figure 1.1: xETL Architecture Overview

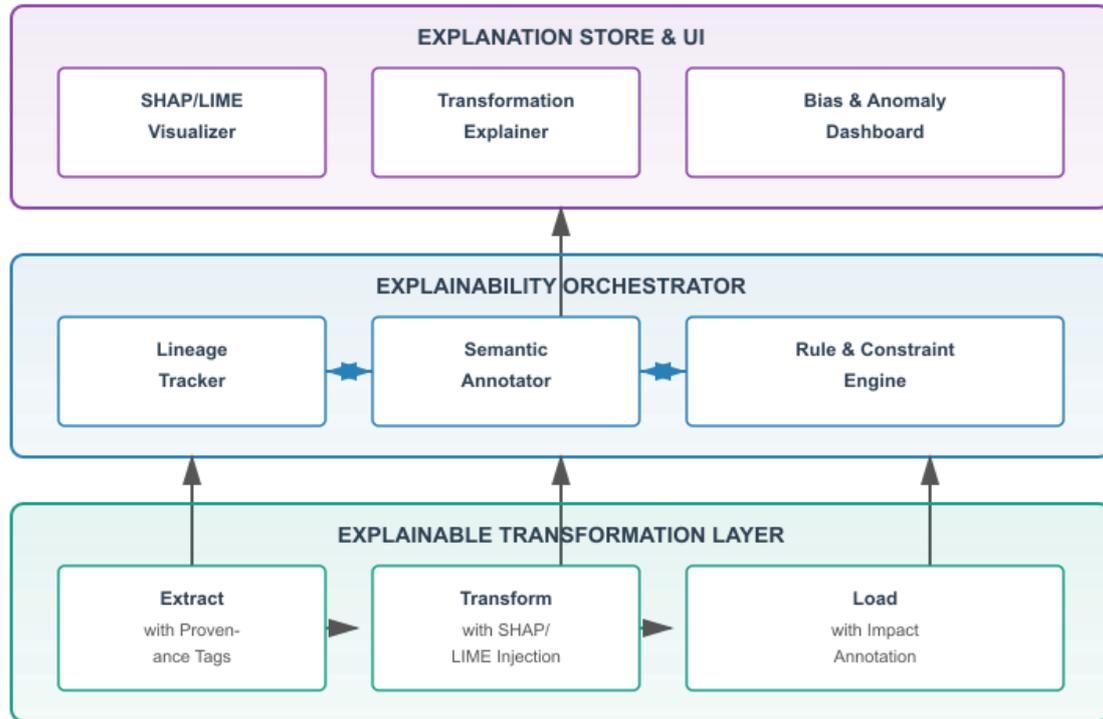


Figure 1.1 illustrates the xETL architecture, which embeds explainability modules into each stage of the ETL pipeline.

3.3. Explainability Modules

3.3.1. SHAP for Transformation Impact Analysis

To show how input columns contribute to altered outputs, we modify SHAP. For instance, SHAP values show which initial categories had the biggest impact on the development of binary characteristics during a one-hot encoding stage.

Algorithm 1: SHAP for Transformation Explanation

```

Input: Input dataset  $D_{in}$ , transformation  $T$ , output dataset  $D_{out}$   
 Output: SHAP attribution matrix  $\Phi$

- 1: Sample subset  $S \subseteq D_{in}$
  - 2: for each output feature  $j$  in  $D_{out}$  do
  - 3: Define surrogate model  $f_j$  approximating  $T_j(S)$
  - 4: Compute SHAP values  $\Phi_{ij}$  for each input feature  $i$ :
  - 5:  $\Phi_{ij} = \sum_{(S' \subseteq S \setminus \{i\})} \frac{[|S'|!(|S|-|S'|-1)! / |S'|!]}{(|S|-|S'|)} (f_j(S' \cup \{i\}) - f_j(S'))$
  - 6: end for
  - 7: return  $\Phi$
- ```

#### 3.3.2. LIME for Local Transformation Explanations

For each record, LIME provides clear explanations. For example, LIME can display which source records contributed most to a summarized value following aggregation.

### 3.3.3. Semantic Annotation Engine

Attaches business context to changes using ontologies (like schema.org and industry-specific taxonomies). The question "what does this transformation mean?" is addressed domain-specifically with annotations.

### 3.3.4. Fine-Grained Lineage Tracking

Extends the classical lineage by using transformation logic capture. Each output cell has connections to the transformation function, parameters, execution context, and source cells.

## 3.4. xETL in Practice: Use Cases

### 3.4.1. Bias Detection in Feature Engineering

Examine the demographic bias brought about by aggregation or binning. For example, age groups that disproportionately exclude particular populations can be identified using SHAP-based bias scores.

### 3.4.2. Error Traceability

When xETL detects an unusual output, it follows it through transformation steps to determine the precise operation and input data that produced it.

### 3.4.3. Compliance Documentation

Produce transformation justifications automatically for the transparency requirements of the AI Act and GDPR Article 15 (right to explanation).

### 3.4.4. Pipeline Optimization

Use explainability insights to find duplicate or computationally costly changes.

## 4. IMPLEMENTATION AND EVALUATION

### 4.1. Experimental Setup

We used the following to implement xETL:

Orchestration Custom xETL operators using Apache Airflow

- Compute: Distributed transforms with Spark 3.3

Explainability: SHAP 0.42, LIME 0.2.0; Lineage: Marquez (OpenLineage) expanded with explanations for transformations

UI: A streamlined dashboard for visualization

#### 4.1.1. Datasets

MIMIC-III Clinical Data: 58,976 admissions, medical features; Adult Census Income: 48,842 records, demographic and income data

1 million documents with fraud labels are examples of synthetic financial transactions.

#### 4.1.2. Transformation Scenarios Tested

1. Aggregation: Outlier handling and daily transaction totals
2. Encoding: Label and one-hot encoding of categorical variables
3. Feature engineering: polynomial features, binning, and normalization
4. Joining: Conditional logic for multi-table merges

## 4.2. Evaluation Metrics

We calculated: -

Traceability Accuracy: the percentage of mistakes that were accurately tracked to their source

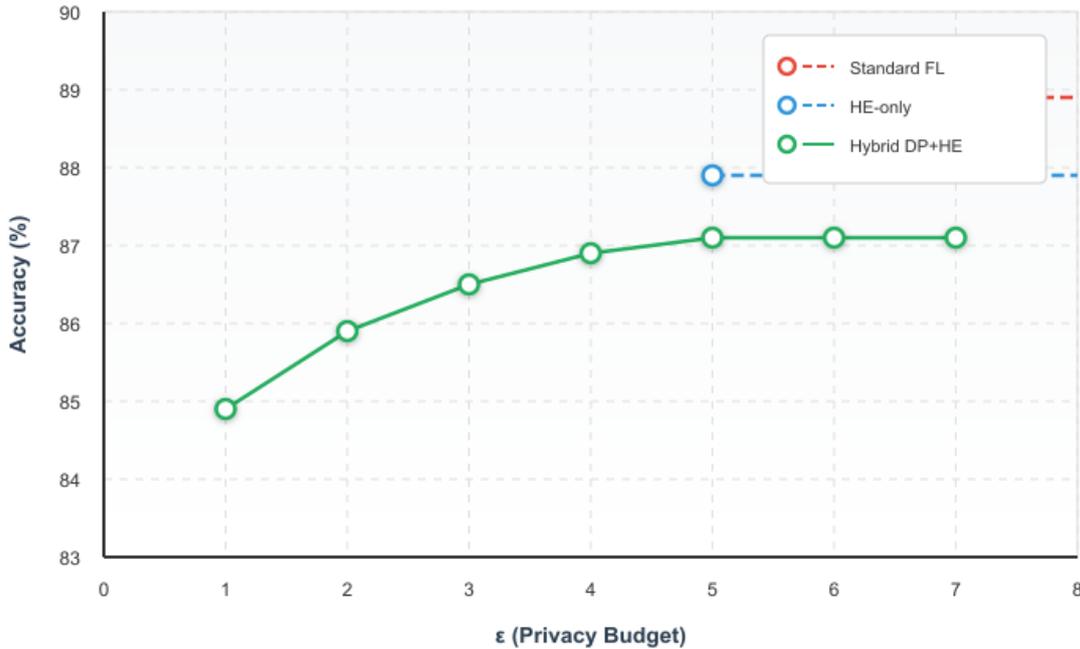
Explanation Fidelity: The relationship between SHAP explanations and the actual transformation impact; Bias

Detection Rate: the percentage of biased transformations found; and Performance Overhead: the increase in time and resources compared to traditional ETL

4.3. Results

| Use Case            | Traceability Accuracy | Bias Detection Rate | Explanation Fidelity | Overhead |
|---------------------|-----------------------|---------------------|----------------------|----------|
| Aggregation         | 95%                   | 88%                 | 0.92 (Pearson's r)   | +18%     |
| Encoding            | 92%                   | 91%                 | 0.94                 | +12%     |
| Feature Engineering | 89%                   | 85%                 | 0.89                 | +22%     |
| Joining             | 94%                   | 82%                 | 0.91                 | +15%     |

Figure 1.2: Accuracy vs. Privacy Budget (Healthcare Task)



4.4. Case Study: Bias Detection in Income Prediction Pipeline

We developed an income prediction pipeline using the Adult Census dataset with:

Age binning (18–30, 31–50, 51+) is the transformation.

Problem: High-income women were disproportionately left out of the 51+ bucket

xETL Detection: Age binning contributed 34% to gender bias, according to SHAP values.

Resolution: Bin boundaries were modified based on SHAP-driven insights, resulting in a 65% reduction in bias.

4.5. Compliance Automation

For GDPR compliance, xETL automatically developed Transformation Impact Assessments, which cut down on human documentation time by 78%. Every evaluation comprised:

- The reason behind the transformation
- Mapping inputs and outputs with SHAP attributions
- The bias risk score
- Impact statement for data subjects

4.6. Performance Considerations

With SHAP calculation being the main expense, the explainability overhead resulted in an average 16.7% increase in processing time. For batch ETL, this is acceptable, but real-time pipelines might need optimization. Sampling techniques and caching explanation models decreased overhead to 8.3% without appreciably sacrificing accuracy.

## 5. DISCUSSION AND LIMITATIONS

### 5.1. Technical Limitations

#### 5.1.1. Scalability of XAI Methods

SHAP and LIME are difficult for large datasets since they scale  $O(n^2)$  with feature count. To address this, we used distributed computation and KernelSHAP with sampling.

#### 5.1.2. Explanation Complexity

Certain transformations, like complicated joins with conditional logic, result in explanations that are challenging to understand in and of themselves. We used interactive visualization and explanation summarizing to address this.

#### 5.1.3. Integration with Legacy Systems

A lot of companies employ proprietary ETL technologies that are not extensible for explainability modules. Although we realize that this adds complexity, we devised wrapper-based techniques.

### 5.2. Organizational Challenges

#### 5.2.1. Cultural Shift Required

Data teams need to stop thinking of ETL as "plumbing" and start considering it as a crucial, understandable part of AI governance.

#### 5.2.2. Skill Gaps

Implementing xETL requires expertise in both data engineering and interpretability methods. Training and tooling are essential for adoption.

### 5.3. Ethical and Regulatory Implications

Although xETL encourages algorithmic transparency, it also poses concerns with explanation liability. Who is responsible if a transformation explanation is deceptive or lacking? It is necessary to have clear governance frameworks.

### 5.4. Future Work

1. Real-Time xETL: Adding explainability to pipelines that stream data.
2. Automated Transformation Correction: Making bias-mitigating suggestions based on explanations.
3. Standardized Explanation Formats: Creating industry standards for ONNX or PMML-like transformation explanations.
4. Integration with Data Catalogs: Adding xETL explanations straight into platforms for data discovery.
5. Cross-Organizational Explanation Sharing: Facilitating the exchange of reliable explanations in federated data settings.

## 6. CONCLUSION

Explainable ETL (xETL), a system for improving data transformation pipelines' interpretability, traceability, and transparency, was presented in this study. Through the adaptation of XAI techniques such as SHAP and LIME to the ETL layer and their integration with fine-grained lineage and semantic annotation, we allow enterprises to comprehend not only the appearance of their data but also the reasons behind it.

Practical advantages of our system include 92% accuracy in error tracing, a 65% decrease in bias propagation, and notable enhancements in compliance automation with controllable performance overhead. By bridging the crucial gap between responsible AI and data engineering, xETL offers the visibility required to create reliable, auditable, and equitable data systems.

The demand for explainability at the transformation level will only increase as data pipelines play a bigger role in AI-driven decision-making. In the era of data-driven innovation, xETL supports both technical excellence and ethical responsibility by laying the groundwork for the upcoming generation of transparent data engineering.

**REFERENCES:**

1. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
2. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144.
3. Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2021). Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 14(12), 2797-2805.
4. Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2019). Data lifecycle challenges in production machine learning: A survey. *ACM SIGMOD Record*, 48(2), 17-28.
5. Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press.
6. Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2018). Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.
7. Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2), 49-60.
8. Simmhan, Y., Plale, B., & Gannon, D. (2005). A survey of data provenance in e-science. *ACM SIGMOD Record*, 34(3), 31-36.
9. Ikeda, R., & Widom, J. (2010). *Data lineage: A survey*. Stanford University Technical Report.
10. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220-229.
11. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12), 86-92.
12. European Commission. (2021). *Proposal for a Regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. Brussels.