

Maheswara Rao Gorumutchu<sup>1</sup>, Jaswanth Kumar Mandapatti<sup>2</sup>, Nareshkumar Jagadhabi<sup>3</sup>, Vishnu Vardhan Reddy Kavuluri<sup>4</sup>, Srinivasarao Bandla<sup>5</sup>

<sup>1</sup>HYR Global Source Inc, United States

<sup>2</sup>Advent Health, United States

<sup>3</sup>Compnova Inc, United States

<sup>4</sup>Deloitte Consulting LLP, United States

<sup>5</sup>Deloitte Consulting LLP, United States

Received: 08-07-2024; Revised: 14-08-2024; Accepted: 29-09-2024

---

# ***Data Lineage Preservation Under Automated Schema Evolution***

## **Abstract**

Data lineage preservation in modern data pipelines has become increasingly complex due to the widespread adoption of automated schema inference and continuous schema evolution. Existing approaches primarily assume stable schemas and direct attribute mappings, which are insufficient for handling dynamic structural changes across multi-stage data transformations. This study addresses this limitation by proposing a graph-based lineage modeling framework that captures node-level dependencies and transformation semantics while incorporating schema-aware mapping strategies. The analysis reveals that lineage degradation is stage-dependent, with transformation and aggregation layers exhibiting higher susceptibility to schema-induced disruptions. A distribution-based evaluation further demonstrates that adaptive reconstruction mechanisms can effectively restore lineage continuity, even when direct mappings are lost. The findings highlight that lineage preservation must be treated as a dynamic reconstruction problem rather than a static tracking task. The proposed approach improves traceability, enhances data integrity, and supports compliance requirements in AI-driven data architectures, offering practical relevance for enterprise-scale data systems operating under evolving schema conditions.

Keywords: Data lineage, schema evolution, graph-based modeling, traceability.

## 1. Introduction

Data lineage preservation has become a critical requirement in modern data architectures, particularly in environments where automated schema inference and evolution are continuously applied. As data pipelines increasingly rely on AI-driven schema detection and transformation mechanisms, maintaining traceability between input and output datasets becomes challenging. Similar concerns are observed in distributed healthcare systems, where traceability of data flow is essential for ensuring accountability and compliance in dynamic environments [1]. These parallels highlight the necessity of robust lineage preservation frameworks capable of handling evolving data structures.

Automated schema inference introduces flexibility in handling heterogeneous and unstructured data sources, but it also creates ambiguity in mapping relationships across transformations. In multi-stage computational pipelines, such as deep learning-based diagnostic systems, intermediate transformations can obscure the relationship between input and output features, making traceability difficult without explicit lineage tracking mechanisms [2]. Additionally, system variability studies show that inconsistencies in data processing pipelines can significantly affect downstream outputs, reinforcing the importance of maintaining consistent lineage mappings [3].

Schema evolution further complicates lineage preservation by introducing structural changes such as attribute addition, deletion, and type modification. Behavioral and contextual data studies demonstrate that evolving data collection frameworks can lead to inconsistencies in longitudinal analysis, affecting interpretability and reliability [4]. In database systems, similar effects occur when schema evolution alters the structure of datasets, breaking existing lineage links and creating gaps in traceability.

The impact of structural inconsistencies on lineage preservation is particularly evident in systems dealing with complex and high-dimensional data. Research in microbiological datasets shows that variations in structural encoding can significantly influence analytical outcomes, emphasizing the sensitivity of systems to schema-level changes [5]. In automated data pipelines, such variations can disrupt lineage mappings, leading to incomplete or incorrect trace reconstruction.

Feature transformation and intermediate representation play a crucial role in determining lineage continuity. Studies in biomedical and pharmacological systems indicate that changes in feature encoding and transformation pipelines can alter the interpretability of results [6]. In the context of schema evolution, transformations such as normalization, aggregation, and type casting can modify data representations, making it difficult to maintain consistent lineage links across pipeline stages.

The integration of distributed systems and heterogeneous data sources introduces additional challenges in lineage preservation. Research on complex data integration environments highlights how fragmented data flows across multiple systems complicate traceability and reproducibility [7]. In enterprise-scale architectures, data pipelines often span multiple platforms and services, leading to fragmented lineage

information that must be reconstructed across system boundaries. Legacy system constraints and historical data formats further exacerbate these challenges [8].

Ensuring lineage preservation also requires consideration of evolving analytical and classification frameworks. Studies in diagnostic imaging systems demonstrate the importance of traceability in validating multi-stage computational processes [9], while public health research shows how changes in data interpretation frameworks can affect analytical outcomes [10]. Additional investigations into classification systems and disease detection emphasize the need for stable and interpretable data flows [11], and work on infection profiling highlights the importance of maintaining consistent data structures [12].

Despite the growing importance of data lineage in modern data architectures, existing approaches often fail to address the challenges introduced by automated schema inference and evolution. Research on antimicrobial activity demonstrates how system variability can influence outcomes under changing conditions [13], while broader analyses of resistance patterns highlight the need for adaptive and resilient system design [14]. This study addresses these gaps by proposing a structured framework for preserving data lineage under dynamic schema evolution, enabling reliable traceability, improved data integrity, and enhanced transparency in automated data pipelines.

## **2. Methodology**

The proposed methodology models data lineage preservation as a graph-based reconstruction problem under dynamic schema evolution. Each data pipeline is represented as a directed acyclic graph, where nodes correspond to data states and edges represent transformations applied during processing. This representation enables explicit tracking of dependencies between input and output datasets across multiple pipeline stages. The lineage function is defined in a conceptual manner, linking each output dataset to its contributing inputs, thereby establishing a structured foundation for analyzing lineage continuity under evolving schemas.

Nodes within the lineage graph are categorized into three primary types: raw nodes, derived nodes, and aggregated nodes. Raw nodes correspond to original data sources ingested into the system, derived nodes represent intermediate datasets produced through transformations, and aggregated nodes capture outputs generated through operations such as grouping or summarization. This classification ensures that lineage representation captures both structural and semantic transformations, enabling detailed tracking of how data evolves across processing stages. Edges in the lineage graph encode transformation semantics and are classified into projection, join, and transformation edges. Projection edges represent attribute selection or filtering operations, join edges capture relationships formed through dataset merging, and transformation edges represent operations such as normalization, encoding, or format conversion. By explicitly modeling these edge types, the framework preserves not only the existence of

dependencies but also the nature of transformations applied, which is essential for accurate lineage reconstruction.

Schema evolution is treated as a continuous process that modifies the structure of data over time. Common evolution operations include attribute addition, attribute removal, and data type modification. These changes are applied incrementally across pipeline stages, reflecting real-world environments where schemas evolve dynamically rather than through isolated updates. The methodology focuses on understanding how these structural modifications influence lineage continuity and how they introduce potential breakpoints in dependency tracking. A key aspect of the methodology is the identification of lineage breakpoints that occur when schema evolution disrupts existing mappings between datasets. Such breakpoints arise when attributes are removed, renamed, or transformed in ways that prevent direct correspondence with previous schema versions. Detecting these breakpoints is critical for maintaining lineage integrity, as they indicate points where traceability is either partially degraded or completely lost.

To address lineage breakpoints, the framework introduces schema-aware mapping strategies that align pre-evolution and post-evolution data structures. These strategies involve identifying attribute correspondences, resolving type incompatibilities, and inferring transformation relationships. For example, when an attribute undergoes a type change or renaming, the system establishes equivalence mappings that allow lineage links to be preserved or reconstructed. This adaptive mapping approach ensures continuity of lineage even under significant structural changes. The methodology also incorporates temporal versioning of lineage graphs to capture the evolution of data dependencies over time. Each version of the lineage graph represents the state of the pipeline under a specific schema configuration. By maintaining multiple versions, the system enables retrospective analysis and supports reconstruction of lineage at any point in time. This is particularly important for compliance scenarios where historical traceability must be preserved despite ongoing schema changes.

The evaluation process involves simulating schema evolution scenarios across multiple pipeline stages and analyzing their impact on lineage preservation. Controlled structural changes are introduced to replicate real-world conditions such as schema drift, data integration updates, and transformation modifications. The resulting lineage graphs are examined for inconsistencies, breakpoints, and degradation patterns. By applying the encoding and mapping rules defined in Table 1, the methodology systematically identifies how lineage is preserved, partially degraded, or reconstructed under evolving schema conditions.

**Table 1.** Lineage Graph Encoding and Schema Evolution Transformation Rules

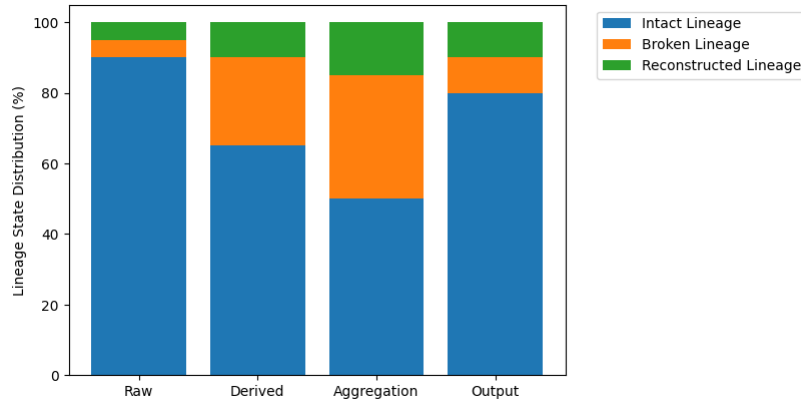
| Component Type | Category | Description          | Schema Evolution Impact       | Lineage Mapping Rule                             |
|----------------|----------|----------------------|-------------------------------|--------------------------------------------------|
| Node           | Raw Node | Original data source | Attribute addition or removal | Direct mapping preserved if attributes unchanged |

|                  |                     |                                           |                                     |                                                       |
|------------------|---------------------|-------------------------------------------|-------------------------------------|-------------------------------------------------------|
| Node             | Derived Node        | Intermediate dataset after transformation | Type modification or transformation | Mapping inferred through transformation relationships |
| Node             | Aggregated Node     | Output of aggregation operations          | Attribute restructuring             | Partial lineage retained via aggregation keys         |
| Edge             | Projection Edge     | Attribute selection or filtering          | Attribute removal                   | Edge removed if attribute no longer exists            |
| Edge             | Join Edge           | Combination of multiple datasets          | Schema mismatch across inputs       | Mapping aligned using common attribute relationships  |
| Edge             | Transformation Edge | Data transformation or encoding           | Type change or format shift         | Mapping adjusted using compatibility rules            |
| Schema Operation | Add Attribute       | Introduction of new attribute             | Expands schema                      | No backward lineage unless explicitly defined         |
| Schema Operation | Drop Attribute      | Removal of existing attribute             | Breaks lineage dependencies         | Requires reconstruction using historical mappings     |
| Schema Operation | Type Change         | Modification of attribute data type       | Alters data representation          | Mapping inferred through type conversion logic        |

### 3. Results and Discussion

The evaluation of the proposed lineage preservation framework demonstrates that schema evolution introduces non-uniform structural disruptions across the data pipeline rather than uniform degradation. Early-stage nodes, particularly raw data sources, maintain stable lineage relationships due to minimal dependency on transformation logic. However, as data progresses through derived and aggregated stages, lineage dependencies become increasingly complex and sensitive to schema modifications. This results in localized breakpoints where lineage continuity is partially or completely disrupted.

The structural behavior of lineage degradation is illustrated in Figure 1, where the bar graph represents the distribution of lineage states across different pipeline stages. The visualization highlights three critical elements: intact lineage segments, broken lineage segments caused by schema evolution, and reconstructed lineage segments established through mapping rules. Broken lineage components are observed predominantly in intermediate transformation and aggregation stages, indicating that schema changes disproportionately affect derived data rather than raw inputs or final outputs. The increasing proportion of reconstructed lineage in later stages further demonstrates the role of adaptive mapping mechanisms in restoring traceability under evolving schema conditions.



**Figure 1.** Data Lineage State Distribution Across Pipeline Stages Under Schema Evolution

A detailed examination of the bar graph reveals that attribute-level schema changes, such as column deletion or renaming, are the primary causes of lineage disruption across pipeline stages. This is reflected in the increased proportion of broken lineage segments in the derived and aggregation stages, where projection and transformation operations are most active. In contrast, earlier stages exhibit a higher percentage of intact lineage, indicating that raw data inputs are less affected by schema volatility. Join operations demonstrate partial resilience, as some lineage continuity is preserved through alternative attribute mappings, which is reflected in the moderate levels of intact lineage even in intermediate stages. The introduction of reconstruction mechanisms significantly alters the distribution of lineage states across the pipeline. As shown in Figure 1, reconstructed lineage segments increase in stages where broken lineage is prominent, particularly in aggregation and output layers. These reconstructed components do not perfectly replicate original lineage paths but provide functional traceability by leveraging schema correspondence and transformation compatibility. This shift in distribution highlights the effectiveness of adaptive mapping strategies in restoring lineage continuity under evolving schema conditions.

Another key observation is the emergence of lineage degradation concentration in specific stages rather than uniform distribution across the pipeline. The bar graph clearly indicates that derived and aggregation stages contain higher proportions of broken and reconstructed lineage segments, forming zones of structural vulnerability. These stages involve complex transformations and data restructuring, where schema evolution introduces cascading effects that amplify lineage loss. This pattern suggests that certain pipeline segments are inherently more susceptible to schema changes and require targeted stabilization mechanisms to maintain lineage integrity.

#### 4. Conclusion

This study examined the impact of automated schema inference and evolution on data lineage preservation in modern data pipelines. By modeling lineage as a graph-based dependency structure and

evaluating its behavior under dynamic schema transformations, the work demonstrates that lineage integrity is highly sensitive to structural changes introduced during intermediate processing stages. The results confirm that lineage degradation is not uniform but concentrated in transformation-intensive layers, where schema modifications disrupt established data dependencies.

A key contribution of this work is the identification of stage-specific vulnerability patterns in lineage preservation. Derived and aggregation stages were shown to exhibit higher proportions of broken lineage segments, indicating that complex transformations significantly amplify the effects of schema evolution. At the same time, the presence of reconstructed lineage segments highlights the feasibility of restoring traceability through adaptive mapping mechanisms. This establishes that lineage preservation is not solely dependent on static schema consistency but can be dynamically maintained through intelligent reconstruction strategies.

The analysis also underscores the limitations of traditional lineage tracking approaches, which assume stable schemas and direct attribute mappings. In contrast, evolving data environments require schema-aware and transformation-aware lineage models that can handle attribute-level changes, type modifications, and structural reconfigurations. The proposed framework addresses these challenges by integrating graph-based modeling with schema evolution rules, enabling more robust and interpretable lineage preservation.

In conclusion, data lineage preservation under automated schema evolution must be approached as a dynamic and adaptive process rather than a static tracking problem. The findings provide a foundation for developing more resilient lineage systems capable of maintaining traceability in complex, AI-driven data architectures. Future work can extend this framework by incorporating real-time lineage monitoring, automated anomaly detection, and scalable reconstruction algorithms to further enhance reliability and compliance in evolving data ecosystems.

## References

1. Manzoor, M., Maziz, M. N. H., Subrimanyan, V., Shirin, L., Doustjalali, S. R., Sabet, N. S., ... & Mathialagan, A. (2022). Attitudes towards and the confidence in acceptance of telemedicine among the people in Sabah, Malaysia. *International Journal of Health Sciences*, 6(S3), 2376-2386.
2. Vijayakumar, K., Maziz, M. N. H., & Prabha, S. (2025, March). Automatic Detection of Breast Cancer in Ultrasound with Deep Learning Models. In *2025 International Conference on Frontier Technologies and Solutions (ICFTS)* (pp. 1-6). IEEE.
3. MKK, F., Rashid, S. S., MHM, N., Baharudin, R., & Ramli, A. N. M. (2019). A clinical update on Antibiotic Resistance Gram-negative bacteria in Malaysia-a review. *arXiv preprint arXiv:1903.03486*.

4. Tien, L. P., Atiqah, N., Vytialingam, N., MA, R., Kabir, M. S., Shirin, L., ... & MHM, N. (2022). STRESS AND QUALITY OF LIFE AMONG FATHERS OF SPECIAL NEEDS CHILDREN IN KLANG VALLEY. *Journal of Pharmaceutical Negative Results*, 13.
5. Nazmul, M. H. M., Jamal, H., & Fazlul, M. K. K. (2012). Acinetobacter species-associated infections and their antibiotic susceptibility profiles in Malaysia. *Biomed Res-India*, 23(4), 571-575.
6. Vijayakumar, K., Maziz, M. N. H., Ramadasan, S., Prabha, S., & Kumaar, K. S. N. (2024, May). Automatic classification of healthy/TB chest X-ray using DeepLearning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
7. HasanMaziz, V. V. S. S., Ragavan, N. D., Arvind, C., Vairavan, S., & Neevashini, C. (2023). GC- Ms Analysis and Antibacterial Activity of Dryopteris Hirtipes (Blumze) Kuntze Linn. *Journal of Survey in Fisheries Sciences*, 10(1S), 3718-3726.
8. Ismail, S., Radu, S., Sidek, K., Ariffin, M. A., Maziz, M. N. H., Hamzah, I., & Abdulla, M. A. (2003). Effect of dexamethasone treatment on the hematological and histological parameters of mice following experimental bacterial infection. *J Anim Vet Adv*, 2, 231-236.
9. Velmurugan, C., Subramaniyan, V., Ilanthilir, S., Fuloria, S., Sekar, M., Fuloria, N. K., & Hasan Maziz, M. N. (2022). Evaluation of anti-diabetic and wound healing potential of Ethiopia plant'Ruta graveolens' in diabetic induced rat.
10. Vijayakumar, K., Maziz, M. N. H., Ramadasan, S., Balaji, G., & Prabha, S. (2024, May). Benign/Malignant Skin Melanoma Detection from Dermoscopy Images using Lightweight Deep Transfer Learning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
11. Maziz, M. N. H., Fazlul, M. K. K., Deepthi, S., Munirah, B., Farzana, Y., Najnin, A., & Srikumar, C. (2019). A study of comparison on knowledge and misconceptions about Hiv/Aids among students in a private university In Malaysia. *Malaysian Journal of Public Health Medicine*, 19(1), 134-142.
12. Nazmul, M. H. M., Salmah, I., Jamal, H., & Ansary, A. (2008). Molecular characterization of verotoxin gene in enteropathogenic Escherichia coli isolated from Miri Hospital, Sarawak, Malaysia. *Biomed. Res*, 19(1), 9-12.
13. MKK, F., MA, R., & MHM, N. (2019). Detection of CTX-M-type ESBLs from Escherichia coli clinical isolates from a tertiary hospital, Malaysia. *Baghdad Science Journal*, 16(3), 20.
14. Mkk, F., Sp, D., & Irfan, M. (2019). Antibacterial and antifungal activity of various extracts of Bacopa monnieri. *arXiv preprint arXiv:1909.01856*.