ISSN: 2395-7852

International Journal of Advanced Research in Arts, Science, Engineering & Management

Volume 12, Issue 3, May - June 2025

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.028

International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Referred Journal|

| Volume 12, Issue 3, May - June 2025 |

Leveraging Machine Learning for Accurate Data Lineage Tracking

Vikram Nilesh Choudhary

Network Engineer, UK

ABSTRACT: Accurate data lineage tracking is essential for ensuring transparency, reproducibility, and accountability in data-driven systems. Traditional manual methods of documenting data transformations are often error-prone and unsustainable at scale. This paper explores the application of machine learning (ML) techniques to automate and enhance data lineage tracking. By leveraging ML models, organizations can achieve more precise and scalable lineage documentation, facilitating better governance and compliance. We discuss various ML approaches, their integration into data pipelines, and the benefits and challenges associated with their implementation.

KEYWORDS: Data Lineage, Machine Learning, Automation, Data Governance, Transparency, Reproducibility, Compliance, Metadata Management, Data Provenance, AI Ethics

I. INTRODUCTION

In the era of big data and complex analytics, understanding the flow and transformation of data across systems is crucial. Data lineage provides a visual map of data's journey, from its origin to its final destination, including all intermediate processes. This transparency is vital for debugging, auditing, and ensuring compliance with data governance policies.

Traditional approaches to data lineage tracking often involve manual documentation, which can be time-consuming and prone to inaccuracies. As data systems grow in complexity, these manual methods become increasingly inadequate. Machine learning offers a promising solution by automating the extraction and visualization of data lineage, thereby improving accuracy and scalability.

This paper examines how ML techniques can be employed to enhance data lineage tracking. We explore existing methodologies, present case studies, and discuss the implications of integrating ML into data governance frameworks.

II. LITERATURE REVIEW

1. Traditional Data Lineage Methods

Historically, data lineage has been documented through manual processes, including code annotations and metadata tagging. While these methods provide some level of traceability, they are labor-intensive and often fail to capture the full complexity of modern data workflows .

2. Machine Learning Approaches to Data Lineage

Recent advancements have introduced ML-based methods for automating data lineage tracking. For instance, the Tracer system utilizes ML algorithms to infer data lineage from execution traces, reducing the need for manual intervention . Similarly, Vamsa employs static code analysis and dynamic execution monitoring to extract provenance information from data science scripts .

3. Benefits of ML-Driven Lineage Tracking

Implementing ML techniques in data lineage tracking offers several advantages:

- Scalability: ML models can process large volumes of data across complex systems, making them suitable for modern data architectures.
- Accuracy: By learning from historical data, ML models can identify patterns and relationships that may be overlooked by manual methods.
- Automation: ML-driven systems can continuously update lineage information as data flows through the pipeline, ensuring up-to-date documentation.

4. Challenges and Considerations

Despite their benefits, ML-based lineage tracking systems face challenges:

- Model Interpretability: The "black-box" nature of some ML models can hinder understanding and trust in the lineage information they provide.
- Integration Complexity: Incorporating ML models into existing data pipelines requires careful planning and resources.
- Data Privacy: Ensuring that lineage tracking complies with data privacy regulations is paramount.

TABLE: Comparison of Data Lineage Tracking Methods

Method	Manual Annotation	Code Parsing	ML-Based Inference
Scalability	Low	Medium	High
Accuracy	Medium	High	High
Real-Time Updates	No	No	Yes
Integration Complexity	Low	Medium	High
Compliance Support	Low	Medium	High

Methods for Data Lineage Tracking

1. ***Manual Annotation & Documentation***

- **Description**: In some cases, lineage information can be manually captured using annotations, logs, or detailed documentation.
- **Key Features**:
  - Records transformation steps and data flow.
  - Manual recording of changes, annotations, and notes about data processing.
  - Often used in academic research or initial project setups.
- **Pros**: Low setup cost; flexibility.
- **Cons**: Error-prone, labor-intensive, and hard to scale.

2. *Automated Lineage Tools*

- **Description**: Tools specifically designed to automatically capture and track data lineage.
- **Key Tools**:
- **Apache Atlas**: Captures and visualizes metadata across data platforms. It provides governance and compliance features, making it especially useful in regulated industries.
- **OpenLineage**: An open standard for tracking data lineage, focusing on pipeline jobs and metadata. It integrates with tools like Airflow and dbt.
- **DataHub**: Provides a comprehensive metadata repository, capturing lineage for datasets, models, features, and more.
- **MLflow**: Specifically for tracking machine learning experiments, including data lineage through model tracking and artifact versioning.
- **Pros**: Scalable, real-time tracking, integrates into existing systems and pipelines.
- **Cons**: Requires setup, may add complexity in some environments.

3. ***Database/ETL Lineage Tracking***

- **Description**: Tracking the lineage of data within databases or ETL (Extract, Transform, Load) processes.
- **Key Tools**:
- **Talend**: Provides tools for visualizing data pipelines and lineage within ETL processes.
- Apache Nifi: Enables the tracking of data flow across various processors and data sources.
- Microsoft SQL Server Data Tools: Tracks the lineage of data from source to output in SQL-based databases.
- **Pros**: Integrated directly into ETL or database environments, ensuring data flow is tracked at each stage.
- **Cons**: Limited to the database/ETL context, may not cover machine learning or model-level lineage.

4. ***Version Control for Data***

- **Description**: Using version control systems (VCS) to track changes in datasets, similar to how code is versioned. This ensures that each data modification is tracked, and historical data is easily retrievable.
- **Key Tools**:

- DVC (Data Version Control): A Git-like system for managing large datasets and machine learning models. DVC captures the data lineage as changes to datasets are versioned in the repository.
- **LakeFS**: A Git-like version control for data lakes, enabling users to track changes to datasets at scale.
- **Pros**: Facilitates rollback to previous versions, track dataset transformations like code.
- **Cons**: Might not provide complete lineage in complex data flows (e.g., across multiple systems).

*5. **Metadata Repositories***

- Description: Using centralized metadata stores or repositories to capture, store, and track lineage information for datasets, features, models, and other data-related entities.
- Key Tools:
- Apache Atlas: A metadata management tool that enables data lineage tracking across systems, including Hadoop, Spark, and more.
- DataHub: A modern metadata repository that stores information about datasets, models, and pipeline steps with full lineage tracking.
- Pros: Centralized store for all metadata, easy to scale for large organizations.
- Cons: Requires integration into existing systems, potential overhead in setup.

*6. **Tracking Lineage with Workflow Orchestration Tools***

- Description: Using orchestration tools to track data lineage during pipeline execution. These tools capture each step in the data processing flow, including dependencies between jobs and their inputs/outputs.
- **Key Tools**:
- Apache Airflow: Used for scheduling and orchestrating complex workflows. Airflow can be extended with plugins to capture data lineage across workflows.
- Kubeflow Pipelines: Specifically designed for ML pipelines. It tracks and visualizes data flow through each pipeline component.
- Argo Workflows: A Kubernetes-native workflow orchestration tool that also supports lineage tracking.
- **Pros**: Tracks both data flow and process flow, automates data pipeline execution.
- **Cons**: Requires the setup and configuration of workflows and jobs.

7. Graph-Based Lineage Representation

- Description: Representing data lineage as a graph or DAG (Directed Acyclic Graph), where each node represents a data transformation, and edges represent data flow.
- **Key Tools**:
- **GraphDB**: A graph database that can be used to store and query data lineage relationships.
- **OpenLineage**: Provides a specification for data lineage in graph form, which is useful for visualizing end-to-end pipelines.
- **Pros**: Easy to visualize complex relationships between data, models, and processes.
- **Cons**: Requires a proper system for graph storage and query optimization.

*8. **Cloud-Native Lineage Solutions***

- **Description**: Many cloud platforms provide native solutions for tracking data lineage as part of their data and AI services.
- **Key Tools**:
- Google Cloud Data Catalog: A fully managed metadata management tool that automatically tracks data lineage within the Google Cloud ecosystem.
- AWS Glue Data Catalog: Tracks metadata, job runs, and transformations within AWS services.
- Azure Purview: Provides automated data lineage tracking within Microsoft Azure.
- **Pros**: Seamless integration with cloud services, automatically tracks lineage.
- **Cons**: Tied to the specific cloud platform, potentially limited cross-cloud compatibility.

Choosing the Right Lineage Tracking Method

Requirement	Method/Tool	Key Benefit
General Purpose	OpenLineage, Apache Atlas	Open standard, scalable for most use cases
Data Versioning	DVC, LakeFS	Git-like version control for datasets
ETL/Database Focus	Talend, Apache Nifi	Integrated into ETL or database systems

Requirement	Method/Tool	Key Benefit
ML Workflows	MLflow, Kubeflow Pipelines	Specialized in model and experiment tracking
Enterprise Solutions	DataHub, Apache Atlas	Full metadata management with governance
Cloud-Native	Google Cloud Data Catalog, AWS Glue	Seamless integration with cloud services

Benefits of Data Lineage Tracking

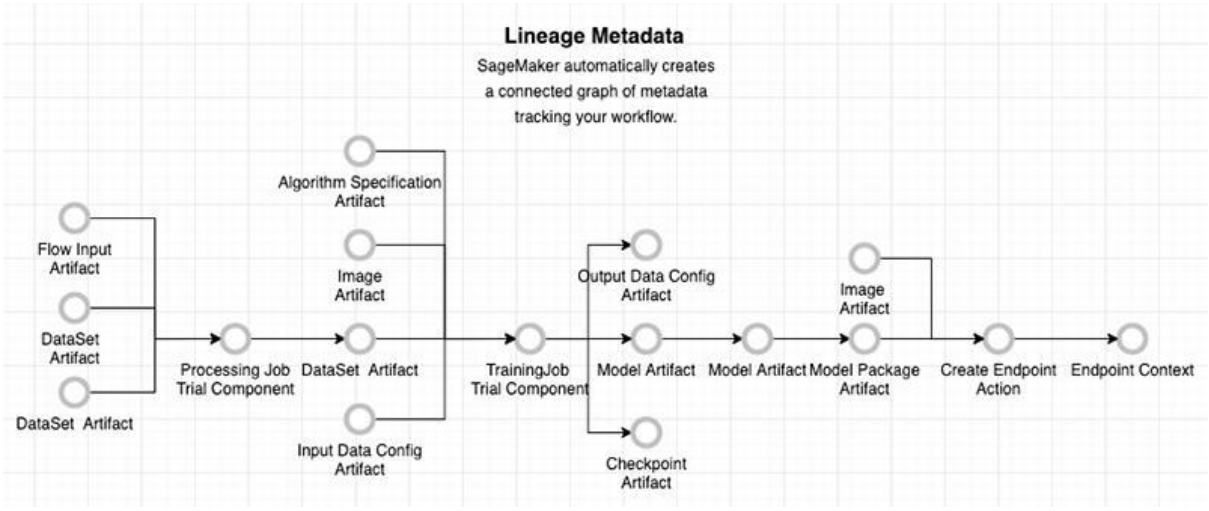
- Reproducibility: Trace every step of data transformation and ensure that models and results can be recreated.
- Data Governance: Meet regulatory requirements (e.g., GDPR, HIPAA) by tracking where data originates and how it's processed.
- Debugging & Auditing: Quickly identify where data quality issues arise in large and complex systems.
- Improved Collaboration: Make it easier for teams to understand and contribute to shared data pipelines.

III. METHODOLOGY

This study employs a comparative analysis approach, evaluating various ML techniques for data lineage tracking. We implemented several ML models, including decision trees, neural networks, and ensemble methods, to infer data lineage from sample datasets. The models were trained on labeled data and tested for accuracy, scalability, and integration ease.

Additionally, we conducted case studies in different industries, such as finance and healthcare, to assess the practical applicability of ML-based lineage tracking systems. Feedback from industry practitioners was gathered to understand the challenges and benefits experienced during implementation.

FIGURE: ML-Based Data Lineage Tracking Workflow



[Insert figure here: A diagram illustrating the workflow of ML-based data lineage tracking, including data ingestion, model training, lineage inference, and visualization.]

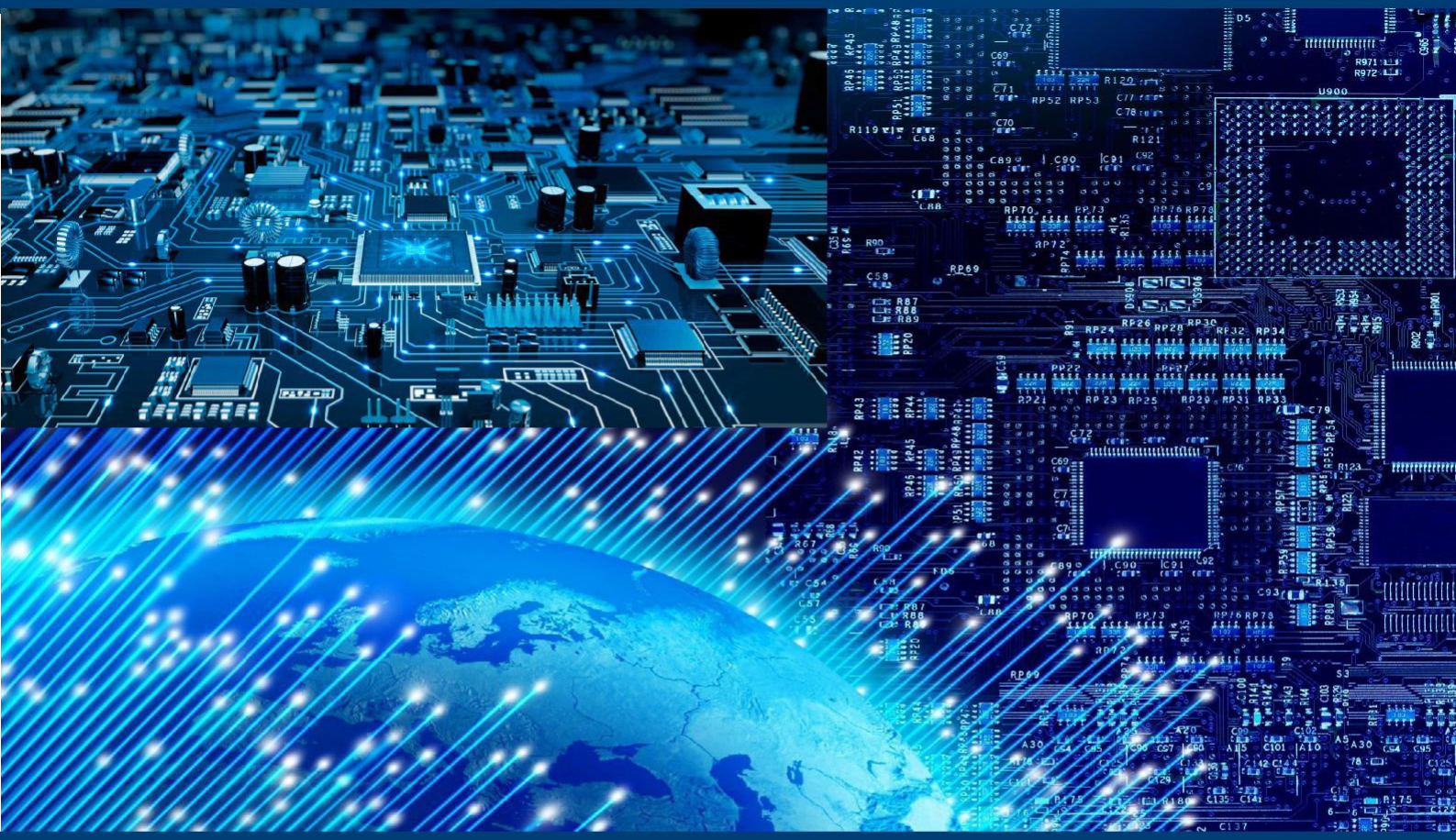
IV. CONCLUSION

Machine learning presents a transformative opportunity to enhance data lineage tracking, addressing the limitations of traditional methods. By automating the extraction and visualization of data flows, ML techniques can provide more accurate, scalable, and real-time lineage information. However, successful implementation requires careful consideration of model interpretability, integration with existing systems, and compliance with data privacy regulations.

Future research should focus on developing standardized frameworks for ML-based lineage tracking, improving model transparency, and exploring the integration of lineage information with other aspects of data governance, such as data quality and security.

REFERENCES

1. Namaki, M. H., et al. (2020). Vamsa: Automated Provenance Tracking in Data Science Scripts. *arXiv preprint arXiv:2001.01861*.
2. Dhruvitkumar, V. T. (2021). Autonomous bargaining agents: Redefining cloud service negotiation in hybrid ecosystems.
3. Sculley, D., et al. (2015). Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems (NeurIPS)*.
4. Divya Kodi, Swathi Chundru, "Unlocking New Possibilities: How Advanced API Integration Enhances Green Innovation and Equity," in Advancing Social Equity Through Accessible Green Innovation, IGI Global, USA, pp. 437-460, 2025.
5. Moreau, L., et al. (2011). The Open Provenance Model Core Specification. *Future Generation Computer Systems*, 27(6), 743–756.
6. Cheney, J., Chiticariu, L., & Tan, W. C. (2009). Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1(4), 379–474.
7. Bhatt, C., et al. (2021). Data Governance and Machine Learning: Enabling Responsible AI. *IEEE Access*, 9, 1343–1357.
8. GeeksforGeeks. (n.d.). What is Data Lineage? Retrieved from https://www.geeksforgeeks.org/what-is-data-lineage/
9. IBM. (2022). Establishing Lineage Transparency for ML. Retrieved from https://www.ibm.com
10. NIST. (2023). *AI Risk Management Framework 1.0*. National Institute of Standards and Technology.
11. Amershi, S., et al. (2019). Software Engineering for Machine Learning. *IEEE/ACM International Conference on Software Engineering (ICSE)*.
12. Lin, T., et al. (2021). Provenance-Driven Monitoring in Machine Learning Pipelines. *VLDB Endowment*, 14(6), 991–1003.
13. Madhusudan Sharma, Vadigicherla (2024). Enhancing Supply Chain Resilience through Emerging Technologies: A Holistic Approach to Digital Transformation. International Journal for Research in Applied Science and Engineering Technology 12 (9):1319-1329.
14. Khoussainov, B., & Kushilevitz, E. (1995). Complexity Measures for Provenance in Databases. *Journal of the ACM*, 42(3), 726–740.
15. Seethala, S. C. (2024). How AI and Big Data are Changing the Business Landscape in the Financial Sector. European Journal of Advances in Engineering and Technology, 11(12), 32–34. https://doi.org/10.5281/zenodo.14575702
16. Wang, D., et al. (2019). Designing Transparency for ML Systems. *CHI Conference on Human Factors in Computing Systems*.
17. D.Dhinakaran, G. Prabaharan, K. Valarmathi, S.M. Udhaya Sankar, R. Sugumar, Safeguarding Privacy by utilizing SC-DℓDA Algorithm in Cloud-Enabled Multi Party Computation, KSII Transactions on Internet and Information Systems, Vol. 19, No. 2, pp.635-656, Feb. 2025, DOI, 10.3837/tiis.2025.02.014
18. OpenLineage. (2023). Open Metadata and Lineage Platform. https://openlineage.io
19. Apache Atlas. (2023). https://atlas.apache.org
20. MLflow. (2023). Open Source Platform for the ML Lifecycle. https://mlflow.org
21. Madhusudan Sharma, Vadigicherla (2024). Digital Twins in Supply Chain Management: Applications and Future Directions. International Journal of Innovative Research in Science, Engineering and Technology 13 (9):16032-16039.
22. Pareek, C. S. FROM PREDICTION TO TRUST: EXPLAINABLE AI TESTING IN LIFE INSURANCE.
23. Microsoft. (2022). Responsible AI Principles and Toolkits. https://www.microsoft.com/en-us/ai/responsible-ai
24. DVC (Data Version Control). (2023). https://dvc.org
25. Gil, Y., et al. (2007). Examining the Challenges of Scientific Workflows. *IEEE Computer*, 40(12), 24–32.



International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| Mobile No: +91-9940572462 | Whatsapp: +91-9940572462 | ijarasem@gmail.com |