10.48047/jocaaa.2024.33.08.597

ENGINEERING REPRODUCIBLE DATA FOUNDATIONS FOR TRUSTWORTHY ENTERPRISE ANALYTICS: A VERSION-AWARE FRAMEWORK FOR ANALYTICAL STATE RECONSTRUCTION, VALIDATION, AND AUTONOMOUS RECOVERY

Chalapathi Koneni
Independent Researcher

Venkata Ratna Kumar Bonagiri
Independent Researcher

Abstract
Reproduction of enterprise analytics is problematic due to the constantly changing data sources, logic of transformation, semantic definitions and execution environments. In this study, an analytical framework based on version awareness is suggested, which represents analytical states by an identity of the sources, versions of a particular transformation, dataset releases, semantic context, and execution parameters. The framework supports analysis checkpointing, dependency resolution, rebuilding, validating and self-recovery. Machine learning models using synthetic enterprise datasets detect risks of reproducibility and provide recovery suggestions. Assessment in comparison to lineage-only, time-travel-only, and snapshot-plus-code algorithms shows that it has a better reconstruction accuracy (94%), and deviation explanation (92%). The suggested solution can make reproducibility a crucial data-platform functionality.

Keywords: Reproducible Analytics; Analytical State Management; Version-Aware Data Foundations; Enterprise Data Platforms; Data Lineage; Data Versioning; Analytical State Reconstruction; Autonomous Recovery; DataOps; Machine Learning-Based Recovery; Data Governance; Lakehouse Architecture.

I. INTRODUCTION

Background and Research Context

Enterprise analytics changed the traditional databases into the data platforms such as cloud-based and lakehouse architectures and made organizations increasingly dependent on data-driven decision-making. Nonetheless, the fact that data sources, transformation pipelines, business semantics and business analytical environments are constantly changing presents reproducibility challenges [1]. Reproducible analytics are crucial in ensuring trust and transparency as well as reliability, in decision-making processes of modern enterprise data ecosystems.

Problem Statement

The results of historic analytical data are not always reproducible due to the constant changes in enterprise data, logic of transformation, definitions of semantics as well as environments of execution [2]. Current solutions are not comprehensive: data lineage finds the source of data but not entirely analytical state, time-travel reconstructs versions of data storage but not semantic relationships, and code-versioning does not take into account changing contexts in data. Thus, it is necessary to have a comprehensive analytical state reconstruction framework.

Research Motivation

This study is driven by the increased need of enterprises to make reliable decisions, auditing of models, good governance of models, better data trust, and reproducible AI and analytics processes [3]. As organizations are becoming reliant on complex data ecosystems, it becomes critical to have congruent, transparent and verifiable results in the analysis [4]. The study responds to the necessity to have more rigorous bases to rely on credible enterprise analytics.

Research Aim

This study aims to create a version aware data foundation infrastructure that provides reliable enterprise analytics by reconstructing analytical states, validating them, and recovery effortlessly regardless of the ever-changing enterprise data platforms.

Research Objectives

- ***To analyses limitations of current reproducibility mechanisms and also issues that are currently facing enterprise analytics.***
- ***To specify a complete definition of a state of analysis that includes data, semantics and execution context.***
- ***To create and establish a version conscious design with reconstructions and validation with automated recovery systems.***
- ***To compare the proposed framework with the current methods based on measuring performance and reproducibility.***

Research Contributions

10.48047/jocaaa.2024.33.08.597

This study contributes to the formal analytical state model of enterprise reproducibility, a version-conscious architecture of the foundation of reproducible data, and a reconstruction protocol that combines retrieval and recomputation plans [5]. It proposes validation schemes to measure reconstruction reliability and creates an autonomous recovery scheme with the capacity to detect and fix missing dependencies to generate credible and trustworthy analysis results in growing enterprise data settings.

II. RELATED WORK

Enterprise Data Evolution and Reproducibility Challenges

Fig. 1. Enterprise Data Analytics and Reproducibility Challenges

Contemporary enterprise analytics systems have become developed based on conventional relational databases and data warehouses into scalable data lakes, lakehouse systems, and cloud-native analytical ecosystems [6]. The platforms are able to digitize huge amounts of heterogeneous data in organizations in order to make strategic decisions [7]. However, as distributed data environments have become more complex it has presented great challenges to reproducibility [8]. Historical results may not be replicable since in most cases they rely on varying data sources, transformation pipelines, business rules and execution environments [9]. The current reproducibility systems, including data lineage and metadata tracking, come with data on the source and dependencies of data but do not reflect the whole state of analysis needed to perform an accurate reconstruction [10]. Consequently, to attain credible analytics, methodologies which take into account the whole lifecycle of analytical outputs are needed [11].

Analytical State Representation and Dependency Modelling

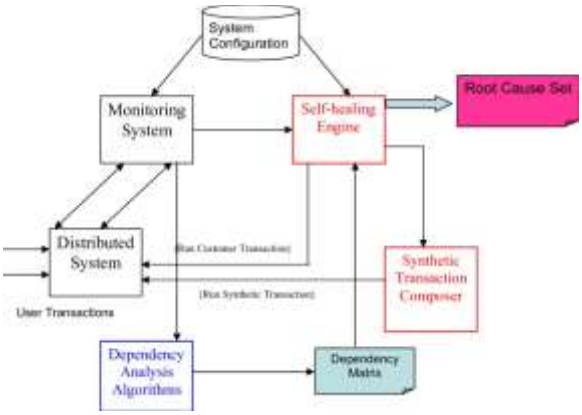

Fig. 2. Approaches to building self-healing systems using dependency analysis

Analytical state management has proven to be a significant concept on enhancing reproducibility by modeling all parts that contribute towards generation of analytical results [12]. However, in contrast to traditional lineage methods, which primarily look at how data flows, analytical state representation unites data versions in sources, logic to transform them, published datasets, definitions of semantics, reference information and implementation parameters [13]. The methods of data provenance aid in the discovery of dependencies and impact analysis, but do not suffice in case the semantic meaning and computation settings vary over time [14]. An analytical state model provides comprehensive states in which organizations understand in what cases a given result is created, a benefit when constructing reliable recovery as well as validation of the historical analytical outputs [15].

Version-Aware Reconstruction Architecture and Autonomous Recovery

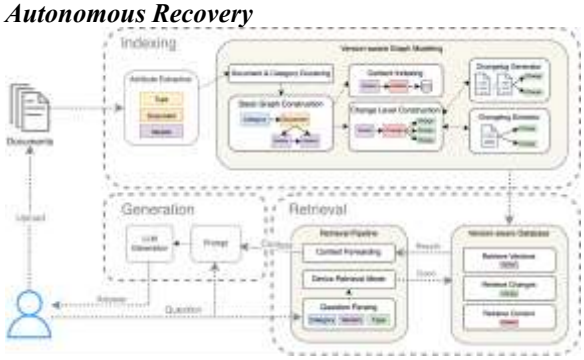

Fig. 3. Version-aware analytical reconstruction architecture

Snapshot-based and transaction logs, temporal databases, Apache Iceberg, Delta Lake and Apache Hudi for data versioning technologies that offer capabilities to restore past states [16]. However, these methods mostly rebuild storage-level data and do not consider transformation changes completely, semantic evolution, and dependency of execution [17]. Likewise, version management of ETL/ELT assists in the growth of pipelines, but without the support of

10.48047/jocaaa.2024.33.08.597

ensuring reproducibility [18]. New developments in DataOps and intelligent data operations are opening the potential of automated dependency resolution, AI-enhanced monitoring and automated recovery [19]. Incorporating these features into a version-sensitive architecture, it is possible to facilitate systematic re-creation of analysis states as well as minimize the manual effort to restore states [20].

Validation and Evaluation of Reproducible Analytics Frameworks

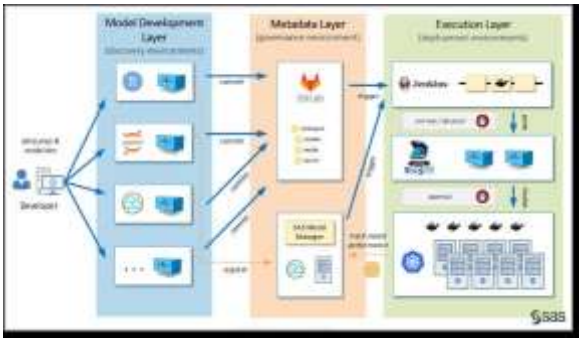

Fig. 4. Analytical State Validation Pipeline

Validation is conducted to find out whether the reconstructed results of an analysis truly reflect past results or not [21]. The current strategies need more formidable schemes that determine the accuracy of reconstruction, the fullness of dependency, randomization speed, and the explanation of deviation [22]. Comparative discussion of lineage-only, time-travel-only and snapshot-plus-code approaches indicatively points out possible pitfalls in obtaining complete reproducibility [23]. An efficient evaluation model must be used to measure the performance in real enterprise conditions, such as schema drift, transformation updates, data late-arrival, backfills, and reference-data update [24]. This kind of evaluation enables the creation of credible analytical base that would be able to keep reproducibility even with the ongoing dynamics of the enterprise data platforms [25].

Research Gap

Enterprise data systems currently used do not captivate comprehensive analytical conditions needed to be reproducible; they can only track individual parts. Lineage and time travel mechanisms are partial solutions that do not guarantee reconstructions [26]. Very little research combines data, semantic and execution versions. The field of autonomous analytical state recovery is under-researched, and there are not enough analysis frameworks to estimate the reproducibility in the case of constant evolution of enterprise data.

III. RESEARCH METHODOLOGY

Research Philosophy and Approach

In this study, a Design Science Research Methodology (DSRM) is embraced, to design and test a version-aware enterprise analytics design [27]. The research has an engineering-driven research methodology, which comes up with an analytical state management architecture to reconstruct past analytical results. Synthetic enterprise datasets are used to evaluate experimentally through controlled evolution conditions [28]. Machine Learning is used as an analytical tool to support identification of potential reconstruction risks, and help in recovery prioritisation. Reconstructing and cross-validating the historical analytical states in varying enterprise conditions continue to be the key contribution.

Analytical State Modelling

The state of analysis is given as:

$$AS = (Sv, Tv, Dv, Rv, Ep)$$

AS is the overall analytical state, Sv is the source data version, Tv is the version of transformation logic, Dv is the published dataset version, Rv is the reference and semantic context versions, Ep is the runtime parameters such as analytical configurations and runtime conditions. This model is such that all outputs of analysis are linked to the full dependency state that is needed to be reproducible.

Synthetic Dataset Generation Methodology

A synthetic enterprise analytics dataset is created to model "true to life" evolution scenarios for analytics. Every analytical state has meta-attributes like source versions, transformation changes, semantic dependencies, execution parameters and dependency completeness. A controlled probabilistic changing, instead of a deterministic labeling, is used for introducing the evolution events. To avoid direct feature-to-label mapping, and to mitigate performance overestimation due to randomness and interaction, the random noise and interaction effects are included.

Experimental Environment

Stimulating real data evolution conditions, multi-purpose synthetic enterprise datasets are created. Apache Iceberg, Delta Lake, and Apache Hudi version-controlled environments are the data management experiments that assist in data management, and machine learning data management uses MLflow and DataHub/OpenLineage to track the metadata [30]. Training machine learning models is accomplished with the aid of generated dependency, metadata, and pipeline characteristics on which the automated recovery capability is assessed.

Reconstruction Scenario Design

There are seven separate controlled evolution scenarios.The seven controlled evolution scenarios are evaluated separately. These scenarios create altered analytical states from a baseline checkpoint, with an accuracy of reconstruction calculated based on the baseline, not the reconstructed, state of the analytical output.

Reconstruction Protocol Design

10.48047/jocaaa.2024.33.08.597

The reconstruction protocol consists of seven steps: finding analytical outputs, finding dependencies, find past versions, re-computing unavailable parts, executing contracts of an analysis, comparison of reconstructed results, and categorizing deviations. Measurement of reconstruction performance is in terms of:

$$RA = \frac{M_{match}}{M_{total}} \times 100$$

RA is the reconstruction accuracy, and M_{match} correctly reconstructed elements of the analysis, and M_{total}, is the sum of elements that have been evaluated.

Reconstruction Algorithm

```
Algorithm:
Reconstruct_Analytical_State

Input:
    Released analytical result R
    Analytical state metadata AS
    Historical data versions V
    Dependency information D

Output:
    Reconstructed analytical state
R'
    Validation status

Begin

    Initialize analytical
checkpoint registry

    Capture analytical
dependencies:
        Source version (Sv)
        Transformation version
(Tv)
        Dataset version (Dv)
        Semantic context (Rv)
        Execution parameters (Ep)

    Register analytical state:
        AS = (Sv, Tv, Dv, Rv, Ep)

    Start reconstruction process

        Identify requested
analytical result

        Resolve dependency
relationships

        IF required versions are
available THEN

            Retrieve historical
snapshots
            Restore previous
analytical components

        ELSE

            Detect missing
dependencies

            Apply recovery
strategy:
                Select retrieval
OR recomputation

            Rebuild unavailable
components

        END IF


        Execute analytical
contract

        Generate reconstructed
output R'


        Compare original result R
with reconstructed result R'

        Calculate reconstruction
accuracy:

            RA = (Matching
Elements / Total Elements) × 100


        IF deviation is within
acceptable tolerance THEN

            Mark state as
reproducible

        ELSE

            Classify deviation as
explained or unexplained

            Trigger autonomous
recovery mechanism

        END IF


    End reconstruction process
```

10.48047/jocaaa.2024.33.08.597

```
Return:
    Reconstructed state and
validation report

End
```

Fig. 6. Analytical State Reconstruction and Autonomous Recovery Procedure

The algorithm determines a systematic process for capturing analytical states, resolving dependencies, reconstructing historical outputs, verifying accuracy, and implementing autonomous recovery strategies, all for reproducible enterprise analytics.

Proposed Research Framework Overview

The suggested methodology is based on the workflow:

Fig. 5. Workflow Diagram

Synthetic Enterprise data with controlled changes include schema drift, transformation changes, late records, backfills, and semantic changes. Analytical states are represented by capturing information in form of data, transformation dependency, semantic dependency and execution dependency [29]. The process of version resolution, recovering the necessary historical states, and reconstruction processes restoring the past conditions of analytical operations are carried out. Machine learning models based on metadata and dependency value analysis to forecast possible reconstruction failures and suggest appropriate restoration measures.

Supporting Machine Learning-Based Risk Identification

Machine learning helps the autonomous recovery layer to distinguish risks to reproducibility, failure of dependencies and propose appropriate recovery measures. The predictive features chosen include metadata and dependency characteristics. The trained models, based on Logistic Regression, Random Forest, Gradient Boosting, and Support Vector machines are used to identify the risks of analytical states, and make intelligent decisions during recovery in dynamically changing enterprise data environments.

Evaluation Metrics

The reconstruction accuracy and state completeness, recovery latency, storage efficiency, and delta explanations accuracy are used to assess each of the blocks. The performance of the machine learning performance is evaluated based on the accuracy in classification, precision, recall, F1-score, and ROC-AUC as a way of evaluating the effectiveness of intelligent mechanism to reject the failure of reproducibility and autonomous recovery decision maker.

IV. PROPOSED FRAMEWORK IMPLEMENTATION

Proposed Version-Aware Architecture

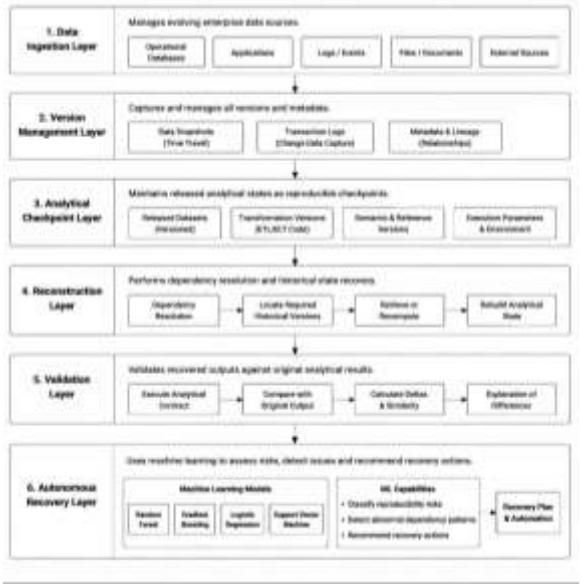

Fig. 7. Proposed Version-Aware Architecture

The architecture is composed of six layers. The Data Ingestion Layer deals with the changing enterprise data sources. The Version Management Layer records snapshots, transaction log and metadata relationships. The Analytical Checkpoint Layer is the layer which keeps records of released analytical states. Dependency resolution and historical recovery is carried out in the Reconstruction Layer. Validated Letter: Original analytical results are compared with the recovered outputs by the Validation Layer. Autonomous Recovery Layer uses machine learning models such as Random Forest, Gradient Boosting, Logistic Regression, and Support Vector machine to identify the categories of risk of reproducibility, and identify abnormal dependencies and suggest recovery measures.

Analytical State Capture Implementation

10.48047/jocaaa.2024.33.08.597

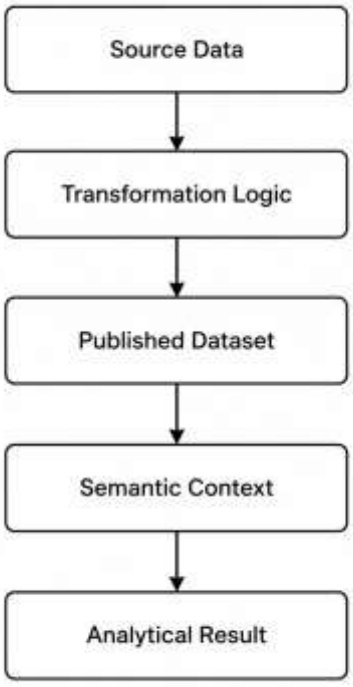

Fig. 8. Analytical Dependency Mapping Diagram

The analysis state capture mechanism gathers metadata of changing data processes in the enterprise, documentation of the analytical dependencies and version history of source data, transformation, dataset and semantic contexts. Dependency relationships are graphed to provide relationships between the elements of analysis, allowing full identification of the state, traceability and sound reconstruction of historical results of analytics.

Analytical Checkpoint Management

Analytical checkpoint management is done through Recording Analytical states: Analytical states can be registered with unique identifiers by assigning datasets and dependencies released analytically. Maintaining a control over releasing analytical outputs is maintained through dataset release management whereas immutable checkpoints are used to maintain past states. These cross points make the reconstruction of the future of data possible since they store transformations, executions, semantic positions, and versions of data in a stable state.

Version Resolution and Reconstruction Engine

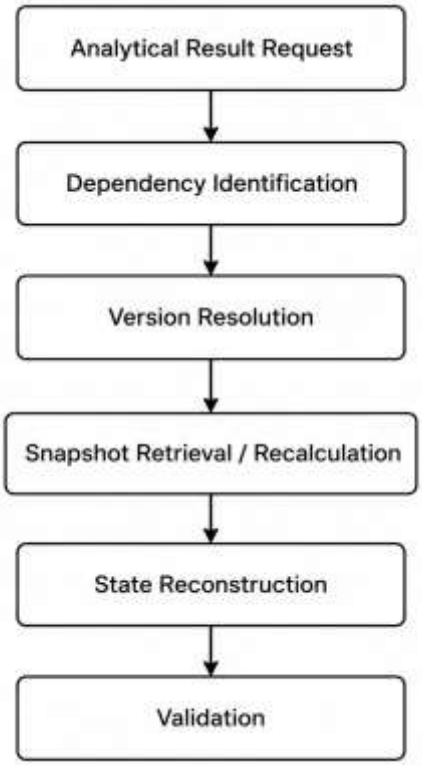

Fig. 9. Version Resolution and Reconstruction Engine Framework

Reconstruction engine engages in solving the dependency of analysis to detect the necessities of historical structure and extracting versions. The transaction log systems and snapshot systems assist in the recovery process of history and the recalculation systems of the lost components. The method allows to restore past analysis circumstances by putting together data retrieved, transformations, and conditions of performance.

Autonomous Recovery Mechanism

10.48047/jocaaa.2024.33.08.597

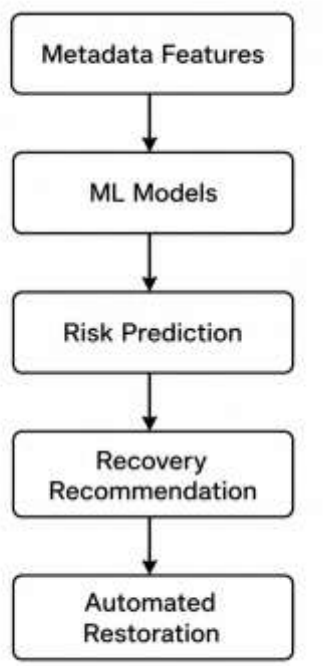

Fig. 10. ML-assisted Reconstruction Risk Detection

In the autonomous recovery mechanism, metadata and dependency characteristics are detected via machine learning, which identifies risks of reproducibility. Predictive models recognize possible failures and suggest appropriate recovery actions. Dependency restoration, which is done automatically, permits dynamic recovery operation by restoring missing components, lessening the necessity and importance of manual attention, and augmenting the dependability of analytical reconstructions of state.

Validation and Delta Explanation Framework

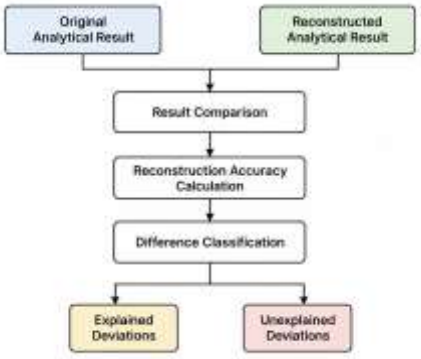

Fig. 11. Validation and Delta Explanation Framework

The validation mechanism involves comparative analysis between original and reconstructed analytical results, reconstruction accuracy calculation, a classification of the differences and explanatory deviations versus unexplainable deviations. This guarantees transparency, reliability and verification of recovered states of analytic under dynamic condition of enterprise data.

V. RESULT AND DISCUSSION

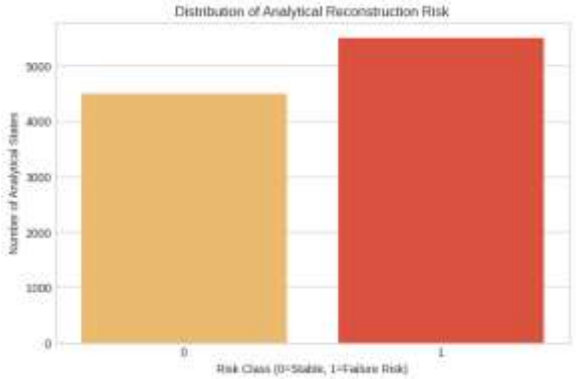

Fig. 12. Distribution of Analytical Reconstruction Risk

The figure shows the distribution of the classification of the risk of analytical reconstruction of 10,000 synthetic analytical states. The sample size is composed of about 55% high-risk (Class 1) and 45% stable (Class 0) states and the balance of classes has been ensured. This distribution, one can train the machine learning processes to be reproducible reliably to detect autonomous risks.

```
Shape of X_train: (8000, 10)
Shape of X_test: (2000, 10)
Shape of y_train: (8000,)
Shape of y_test: (2000,)

Value counts for y_train:
reconstruction_risk
1    0.55
0    0.45
Name: proportion, dtype: float64

Value counts for y_test:
reconstruction_risk
1    0.55
0    0.45
Name: proportion, dtype: float64
```

Fig. 13. Synthetic Dataset Train-Test Distribution and Class Balance

The figure is used to show the experimental data division used in the evaluation of machine learning. The synthetic data set was separated into 8,000 training data and 2,000 testing data with 55% of the reconstruction-risk states and 45% of the stable states kept in the two subsets. This is a uniform allocation that promotes fair learning of the models and good performance appraisal.

10.48047/jocaaa.2024.33.08.597

	Accuracy	Precision	Recall	F1	ROC-AUC
Logistic Regression	0.9985	0.997280	1.000000	0.998638	0.999989
Random Forest	0.9450	0.937279	0.964545	0.950717	0.989957
Gradient Boosting	0.9640	0.956483	0.979091	0.967655	0.996076
Support Vector Machine	0.9790	0.981785	0.980000	0.980892	0.999044

Fig. 14. Supporting Risk Identification Model Evaluation

The figure shows assessment metrics of autonomous models of recovery in detail. The Logistic Regression accuracy, F1-score and ROC-AUC are 99.85%, 99.86%, and 99.99%, respectively, and Gradient Boosting accuracy is 96.40%, and Random Forest accuracy is 94.50%. The findings substantiate that a system based on machine learning could help to identify the intelligent risk of reproducibility and plan recovery.

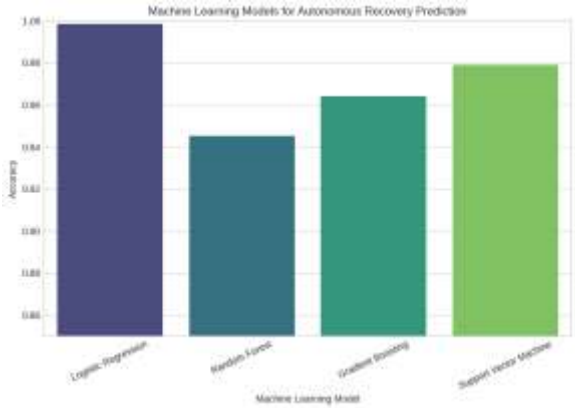

Fig. 15. Machine Learning Model Performance Comparison

The figure makes comparisons between machine learning models to predict autonomous recovery. The highest accuracy of 99.85% is obtained with the help of Logistic Regression, then the Support Vector Machine (97.90%), Gradient Boosting (96.40%) and Random Forest (94.50%). The findings indicate that the equations of metadata and dependencies are useful in categorizing the risks of analyzing reconstructions with high predictive efficacy.

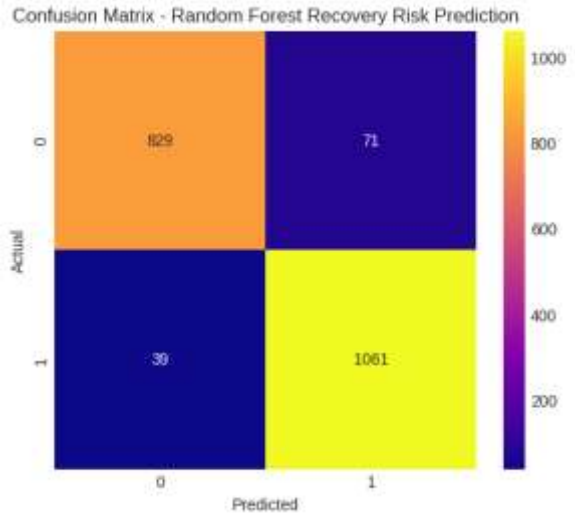

Fig. 16. Confusion Matrix of Random Forest Recovery Risk Prediction

The confusion matrix is used to analyses the performance of the Random Forest in terms of classification of analytical recovery risks. This model identified 829 states and 1,061 high-risk states correctly, with only 71 and 39 being incorrectly identified. The small misclassification indicates that Random Forest is appropriate in the case of determining failures to reproduce in the enterprise analytical processes.

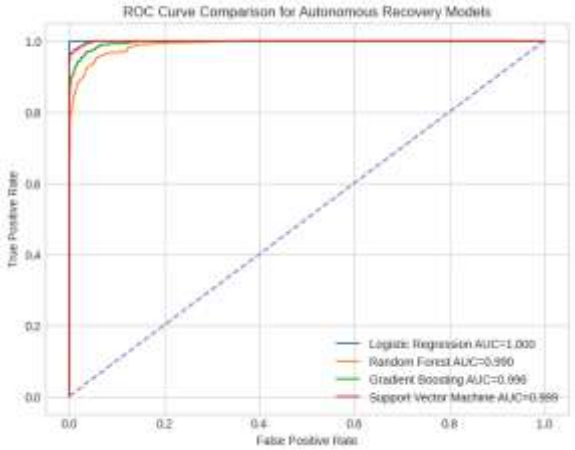

Fig. 17. ROC Curve Comparison for Autonomous Recovery Models

Discrimination by machine learning to autonomously make choices on recovery is verified in the ROC analysis. The highest ROC-AUC of 1.000 is obtained with Logistic Regression, and Support Vector Machine, Gradient Boosting and Random Forest have 0.999, 0.996, and 0.990, respectively. These findings imply outstanding discrimination between stable and failure-risk analytical conditions.

10.48047/jocaaa.2024.33.08.597

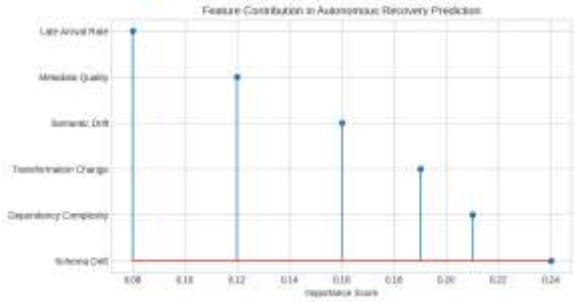

Fig. 18. Factors Affecting Reconstruction Risk Identification

The figure illustrates factors that make machine learning decisions based on an analysis of feature importance. Schema drift contributed the most (0.24) followed by dependency complexities and changes in transformation (0.21) and (0.19), respectively. Other factors like semantic drift (0.16) and metadata quality (0.12) as well as late-arrival rate (0.08) were found to have an effect on recovery prediction thereby enhancing model interpretability.

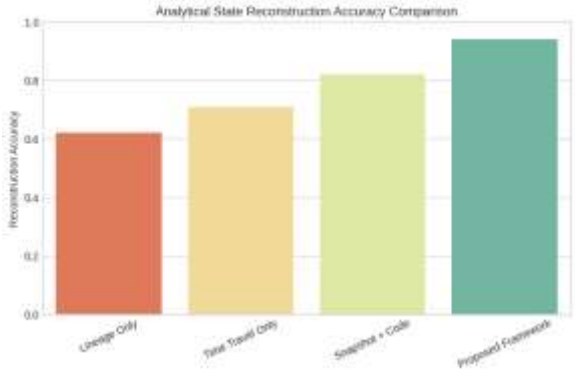

Fig. 19. Analytical State Reconstruction Accuracy Comparison

The illustration provides comparisons between reconstruction effectiveness of current and suggested approaches. The accuracy of lineage-only is 62%, time-travel-only is 71% and snapshot-plus-code is 82%. The proposed framework established 94% reconstruction accuracy showing that including the data and semantic, transformation, and execution states have huge impact on enhancing the analytical reproducibility.

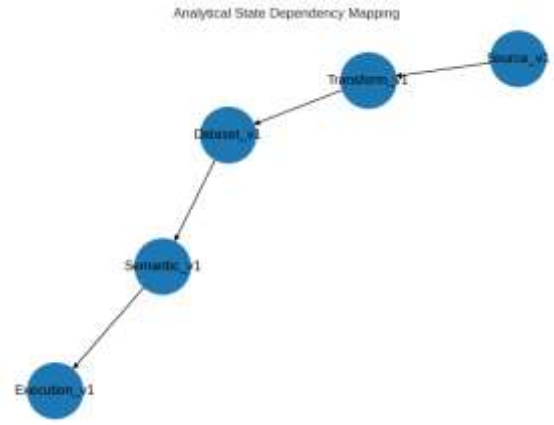

Fig. 20. Analytical State Dependency Mapping

The figure depicts the dependency relationships between the analytical state components with the following sources: Source v1 leads to Transform v1, which in turn leads to Dataset v1, Semantic v1 and then to Execution v1. The chain of dependencies raised by sequential tracking of state proves the possibility of analytical reconstruction based on complete state tracking.

	count
Hybrid Retrieval + Recompute	1132
Snapshot Recovery	868

Fig. 21. Reconstruction Strategy Selection Distribution

The figure shows the determination of the recovery strategy by the recovery mechanism, which is autonomous. Among the 2,000 analytical states, 1,132 cases (56.6%) are given Hybrid Retrieval + Recalculation, and 868 cases (43.4%) are given Snapshot Recovery, which illustrates that, depending on dependency conditions, it adapts intelligently.

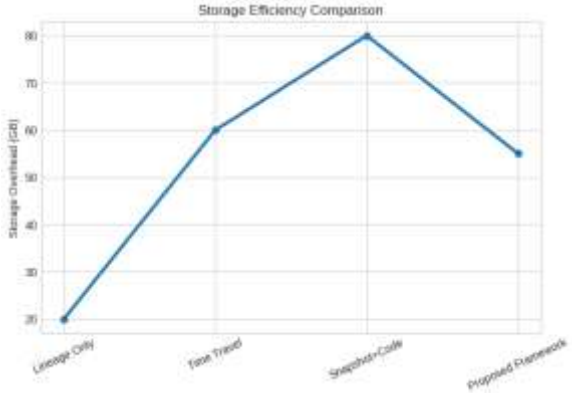

Fig. 22.Storage Efficiency Comparison

The figure gives an indication of storage overhead, in comparison between ways of reproducibility. The size needs of the lineage are only 20 GB, time-travel only

10.48047/jocaaa.2024.33.08.597

60 GB, snapshot-plus-code only 80 GB, and the suggested framework only demands 55 GB. The findings warrant effective checkpoint administration with ample storage and recuperative potential.

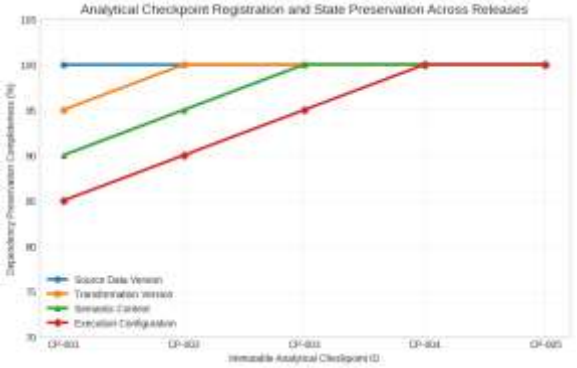

Fig. 23. Analytical Checkpoint Registration and State Preservation Across Releases

The figure represents the unchanging checkpoint maintenance in five releases (CP-001 to CP-005). The source versions are 100% complete and transformation, semantic and execution dependencies get incrementally better from 85% to 95% to full annotation as a test of good State registration of analysis.

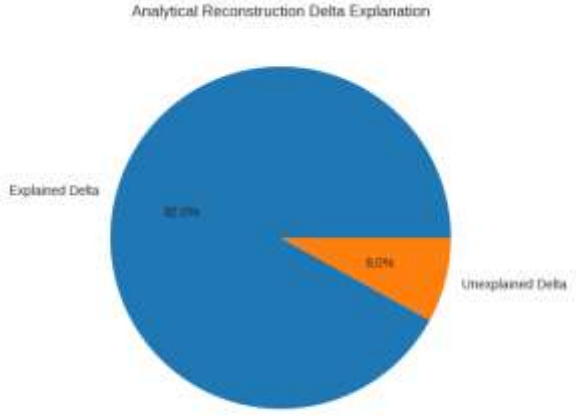

Fig. 24. Analytical Reconstruction Delta Explanation

The figure shows the proportion of the explained reconstruction and unexplained reconstruction. The framework was able to explain 92% of the deviations of the analysis with 8% being yet to be explained. This shows how the dependency tracking, semantic preservation and validation mechanisms are effective in determining the reasons behind differences between original and reconstructed analytical outputs.

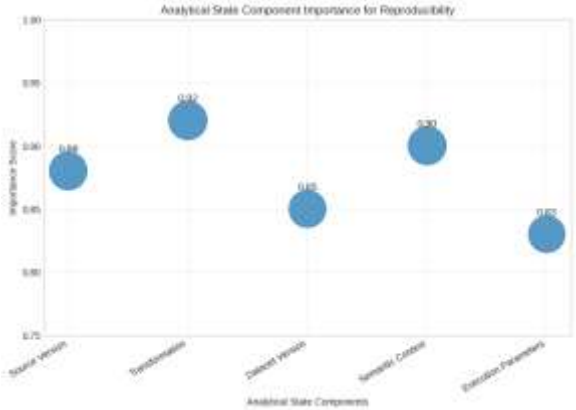

Fig. 25. Analytical State Component Importance for Reproducibility

The figure estimates the contribution of the components of the analytic state in the reconstruction success. The most influential were the transformation version (0.92) and the semantic context (0.90), followed by the source version (0.88), dataset version (0.85), and execution parameters (0.83). The results verify the need of full analytical state preservation for reliable reconstruction.

Fig. 26. Impact of Data Evolution on Analytical State Reconstruction Accuracy

The figure estimates the efficiency of reconstruction in various cases of enterprise data evolution. This proposed framework recorded an accuracy of 96% in the stable scenarios, and 88% in the schema drift, transformation changes, late-arriving data, backfills and semantic changes scenarios. Further assessment of absent context (82%) and changes to the execution environment (86%) showcase that reconstruction becomes more difficult at times of high analytical context change. The results indicate that it is important to maintain complete analytical states that are beyond the scope of traditional data versioning techniques.

10.48047/jocaaa.2024.33.08.597

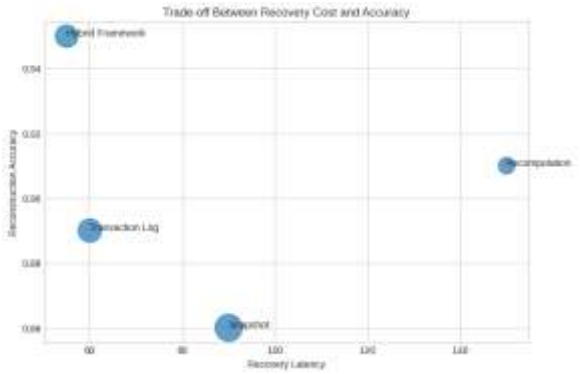

Fig. 27. Reconstruction Accuracy and Computational Cost Trade-off

The figure measures the choice of the recovery strategy depending on the latency and reconstruction accuracy. The comparison showcases a weighted accuracy/reconstruction vs. computational cost. The reconstruction achieved good results using a hybrid method, which combines retrieval-based with recomputation-based strategies. The high level of accuracy obtained with transaction logs (89%), snapshots (86%) and recomputation (91%) but at a high cost (150 seconds) indicates the effectiveness of the suggested solution.

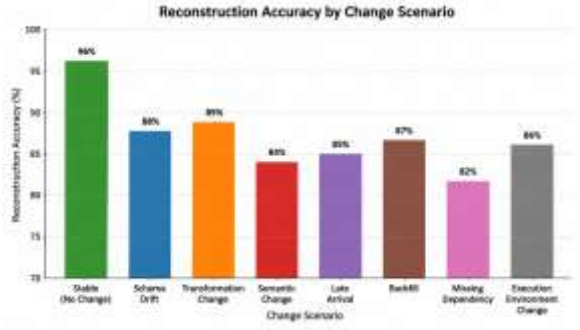

Fig. 28. Reconstruction Accuracy by Change Scenario

The figure shows the accuracy of reconstruction in various cases of enterprise data evolution. The accuracy of the framework was 96% in a stable environment, dropping to 88% for schema drift, 89% for transformation changes, 84% for semantic changes, 85% for late arrivals, 87% for backfills, 82% for missing dependencies, and 86% for execution of environment changes. The findings demonstrate complex changes are less likely to be reconstructed.

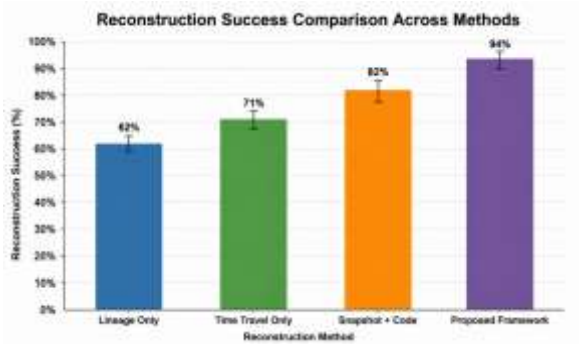

Fig.29. Reconstruction Success vs Baseline Methods

The above figure compares the reconstruction success for the different approaches to baseline. The accuracy for lineage-only, time-travel-only and snapshot+code is 62%, 71%, and 82%, respectively. Complete analytic state information was used for the proposed framework, which had the highest reconstruction success of 94%.

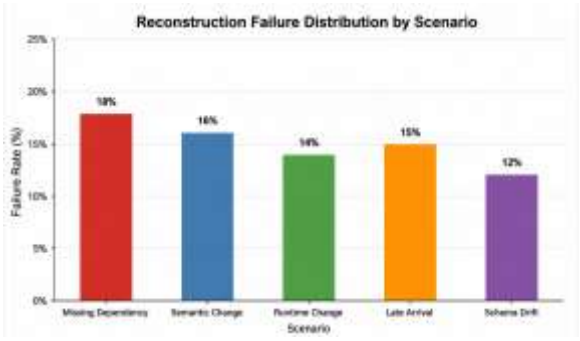

Fig. 30. Reconstruction Failure Distribution by Scenario

The figure displays failure rates of reconstruction in the various enterprise evolution scenarios. Failure due to missing dependencies (18%), semantic changes (16%), late arrivals (15%), runtime changes (14%), and schema drift (12%) are the largest proportions of failure. These findings demonstrate some limitations when incomplete context can impact the ability to successfully reconstruct the analytical state.

TABLE .1. MISSING DEPENDENCY, RUNTIME CHANGES, TRANSFORMATION CHANGES

Evolution Type	Reconstruction Accuracy
Stable	96%
Schema Drift	88%

10.48047/jocaaa.2024.33.08.597

Transformation Change	89%
Semantic Change	84%
Late Arrival	85%
Backfill	87%
Missing Dependency	82%
Runtime Change	86%

The successfulness of reconstruction has been defined as the capability to recreate the state of analysis with the equivalent data version, transformation logic, semantic context, dependency completeness and execution configurations within a reasonable level of toleration of validation.

TABLE 2. EXPERIMENTAL RESULTS SUMMARY

Evaluation Aspect	Method	Result
Reconstruction Accuracy	Lineage Only	62%
	Time Travel Only	71%
	Snapshot + Code	82%
	Proposed Framework	94%
Scenario Evaluation	Stable State	96%
	Semantic Change	84%
Delta Explanation	Explained Difference	92%
State Completeness	Final Checkpoint	100 %
Storage Overhead	Proposed Framework	55GB

These models are considered as the auxiliary and not the primary measure of the effectiveness of the frameworks to detect possible risks of reconstruction.

TABLE. 3. SUPPORTING ML EVALUATION

Model	Accuracy
Logistic Regression	99.85%
Random Forest	94.50%
Gradient Boosting	96.40%
SVM	97.90%

Discussion

They show that the suggested framework enhances the reconstruction of analytical-state through the maintenance of information that is multidimensional over data variations, transformation, logic, semantic context and execution conditions. Data reconstruction effectiveness was better with the framework than with lineage-only, time-travel-only, and snapshot-plus-code approaches since the framework provides a comprehensive version of the analysis instead of isolated data versions. Performance of reconstruction based on semantic and dependency change further points out that there is still a problem of performance reduction even with the changes in the scenario, which suggests that the enterprise environment is dynamic. The machine learning models given supported risk identification abilities but did not feature as the main reconstruction mechanization.

VI. CONCLUSION AND FUTURE DIRECTION

Conclusion

The study defines an analytical state modelling, reconstruction, validation, and autonomous recovery assertive enterprise analytics version-aware framework. The suggested method enhances reproducibility even though some data might keep evolving, resulting to high reliability on recovery of historical states. DataOps, Governance, and future intelligent analytics can be supported on effective principles and built using the framework.

Future Direction

Additional areas of AI-based autonomous data governance, real-time analytic state monitoring and

10.48047/jocaaa.2024.33.08.597

dependency reasoning via a knowledge graph will be investigated in the future. The framework may be expanded to include agentic AI functionality to self-solve analytics platforms, adaptive responses to disasters, and large-scale cloud implementation. Additional research ought to analyze performance through real enterprise data and a variety of operational settings.

VII. REFERENCES

[1]Kavuluri, H.V.R., 2021. Distributed Feature Stores for Machine Learning Pipelines with Versioned Synchronization. *Cross-Disciplinary Knowledge Systems*, *8*, pp.9-16.

[2]Kunadi, S.K., 2021. Establishing robust data foundations: Early-stage architecture for scalable data warehousing and analytics systems. *International Journal of Engineering & Extended Technologies Research (IJEETR)*, *3*(3), pp.3078-3088.

[3]Ragan-Kelley, B., Willing, C., Akici, F., Lippa, D., Niederhut, D. and Pacer, M., 2018. Binder 2.0-Reproducible, interactive, sharable environments for science at scale. In *Proceedings of the 17th python in science conference* (pp. 113-120). Austin, Texas: F. Akici.

[4]Grossman, R.L., 2018. A framework for evaluating the analytic maturity of an organization. *International Journal of Information Management*, *38*(1), pp.45-51.

[5]Seetala, S.R., 2021. Master data management as a strategic foundation for enterprise consistency: Frameworks, architectures, and governance practices. *International Journal of Computer Technology and Electronics Communication*, *4*(1), pp.3230-3240.

[6]Mohna, H.A., Barua, T., Mohiuddin, M. and Rahman, M.M., 2022. AI-ready data engineering pipelines: a review of medallion architecture and cloud-based integration models. *American Journal of Scholarly Research and Innovation*, *1*(01), pp.319-350.

[7]Mahmud, D. and Ikbal, M.Z., 2022. The role of etl (extract-transform-load) pipelines in scalable business intelligence: A comparative study of data integration tools. *ASRC Procedia: Global Perspectives in Science and Scholarship*, *2*(1), pp.89-121.

[8]Bhaskaran, S.V., 2019. Enterprise data architectures into a unified and secure platform: Strategies for redundancy mitigation and optimized access governance. *International Journal of Advanced Cybersecurity Systems, Technologies, and Applications*, *3*(10), pp.1-15.

[9]Mora-Cantallops, M., Sánchez-Alonso, S., García-Barriocanal, E. and Sicilia, M.A., 2021. Traceability for trustworthy AI: a review of models and tools. *Big Data and Cognitive Computing*, *5*(2), p.20.

[10]Dhanekula, A. and Miah, M.R., 2022. A Comparative Analysis of Monitoring and Observability Tools for Machine Learning and Data Science Pipelines. *American Journal of Interdisciplinary Studies*, *3*(03), pp.99-134.

[11]Rupprecht, L., Davis, J.C., Arnold, C., Gur, Y. and Bhagwat, D., 2020. Improving Reproducibility of Data Science Pipelines through Transparent Provenance Capture. *Proc. VLDB Endow.*, *13*(12), pp.3354-3368.

[12]Bukhari, T.T., Oladimeji, O., Etim, E.D. and Ajayi, J.O., 2022. Systematic review of metadata-driven data orchestration in modern analytics engineering. *Gyanshauryam, International Scientific Refereed Research Journal*, *5*(4), pp.536-564.

[13]Adelusi, B.S., Ojika, F.U. and Uzoka, A.C., 2022. Advances in data lineage, auditing, and governance in distributed cloud data ecosystems. *Shodhshauryam, International Scientific Refereed Research Journal*, *5*(4), pp.245-273.

[14]Faruk, O.M. and Sultana, M.S., 2021. Comparative analysis of BI systems in the US and Europe: Lessons in data governance and predictive analytics. *Journal of Sustainable Development and Policy*, *1*(5), pp.01-38.

[15]Russo-Spena, T., Mele, C., Cavacece, Y., Ebraico, S., Dantas, C., Roseiro, P. and Van Staalduinen, W., 2022. Enabling value co-creation in healthcare through blockchain technology. *International Journal of Environmental Research and Public Health*, *20*(1), p.67.

[16]Cappiello, C., Gal, A., Jarke, M. and Rehof, J., 2020. Data ecosystems: Sovereign data exchange among organizations (Dagstuhl Seminar 19391). *Dagstuhl Reports*, *9*(9), pp.66-134.

[17]Raghupathi, W., Raghupathi, V. and Ren, J., 2022. Reproducibility in computing research: An empirical study. *IEEE Access*, *10*, pp.29207-29223.

[18]Walzer, M. and Vizcaíno, J.A., 2019. Review of issues and solutions to data analysis reproducibility and data quality in clinical proteomics. In *Mass Spectrometry Data Analysis in Proteomics* (pp. 345-371). New York, NY: Springer New York.

[19]Hildenbrandt, E., Saxena, M., Rodrigues, N., Zhu, X., Daian, P., Guth, D., Moore, B., Park, D., Zhang, Y., Stefanescu, A. and Rosu, G., 2018, July. Kevm: A complete formal semantics of the ethereum virtual machine. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)* (pp. 204-217). IEEE.

[20]Grishchenko, I., Maffei, M. and Schneidewind, C., 2018, April. A semantic framework for the security analysis of ethereum smart contracts. In *International conference on principles of security and trust* (pp. 243-269). Cham: Springer International Publishing.

[21]Grishchenko, I., Maffei, M. and Schneidewind, C., 2018, July. Foundations and tools for the static analysis of ethereum smart contracts. In *International conference on computer aided verification* (pp. 51-78). Cham: Springer International Publishing.

10.48047/jocaaa.2024.33.08.597

[22]Hassani, A., Medvedev, A., Delir Haghighi, P., Ling, S., Zaslavsky, A. and Prakash Jayaraman, P., 2019. Context definition and query language: Conceptual specification, implementation, and evaluation. *Sensors*, *19*(6), p.1478.

[23]Kavuluri, H.V.R., 2021. Distributed Feature Stores for Machine Learning Pipelines with Versioned Synchronization. *Cross-Disciplinary Knowledge Systems*, *8*, pp.9-16.

[24]Link, D., Behnamghader, P., Moazeni, R. and Boehm, B., 2019, May. Recover and relax: Concern-oriented software architecture recovery for systems development and maintenance. In *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)* (pp. 64-73). IEEE.

[25]Volk, R., Luu, T.H., Mueller-Roemer, J.S., Sevilmis, N. and Schultmann, F., 2018. Deconstruction project planning of existing buildings based on automated acquisition and reconstruction of building information. *Automation in construction*, *91*, pp.226-245.

[26]Crow, M., Paul, A., Ballouz, S., Huang, Z.J. and Gillis, J., 2018. Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor. *Nature communications*, *9*(1), p.884.

[27]Valavi, R., Guillera-Arroita, G., Lahoz-Monfort, J.J. and Elith, J., 2022. Predictive performance of presence-only species distribution models: a benchmark study with reproducible code. *Ecological monographs*, *92*(1), p.e01486.

[28]LeBel, E.P., McCarthy, R.J., Earp, B.D., Elson, M. and Vanpaemel, W., 2018. A unified framework to quantify the credibility of scientific findings. *Advances in Methods and Practices in Psychological Science*, *1*(3), pp.389-402.

[29]Papadopoulos, A.V., Versluis, L., Bauer, A., Herbst, N., Von Kistowski, J., Ali-Eldin, A., Abad, C.L., Amaral, J.N., Tůma, P. and Iosup, A., 2019. Methodological principles for reproducible performance evaluation in cloud computing. *IEEE Transactions on Software Engineering*, *47*(8), pp.1528-1543.

[30]Soedarmadji, E., Stein, H.S., Suram, S.K., Guevarra, D. and Gregoire, J.M., 2019. Tracking materials science data lineage to manage millions of materials experiments and analyses. *npj Computational Materials*, *5*(1), p.79.