Design and Implementation of a Data Reconciliation System Using Deterministic Fuzzy Matching with AI-Assisted Explainability and Confidence Analysis

Mrs. N. Jeevana Deepa[1], R. Vaishnavi[2], M.N.V.D. Surya Sri[3],
N.B.N. Surekha[4], V. Samson Raj[5], Prof. Y. Venkat[6]
[1,2,3,4]*UG Scholar, Srinivasa Institute of Engineering and Technology*
[5]*Assistant Professor, Srinivasa Institute of Engineering and Technology*
[6]*Prof, Srinivasa Institute of Engineering and Technology*

Abstract— **Data reconciliation is a critical requirement in enterprise systems where records from multiple sources must be accurately matched. Traditional fuzzy matching techniques often fail in such scenarios because they ignore or underweight numeric identifiers embedded within textual descriptions, leading to false positives and unreliable mappings. This project presents the design and implementation of a Deterministic Data Reconciliation System using numeric-aware fuzzy matching, augmented with AI-assisted explainability and ML-based confidence analysis. The proposed system combines textual similarity with explicit numeric comparison to enforce consistency and eliminate incorrect matches caused by numeric discrepancies. The system is implemented using an enterprise-grade N-Tier architecture comprising a React.js with TypeScript frontend, a Flask-based RESTful backend, and a PostgreSQL relational database. Secure access is enforced using JWT-based authentication with Role-Based Access Control (RBAC). While the deterministic matching engine remains the authoritative source of truth, OpenAI GPT-4o-mini is used exclusively to generate human-readable explanations, and Machine Learning (scikit-learn) provides independent confidence indicators (HIGH/MEDIUM/REVIEW) for reconciliation decisions. Experimental observations demonstrate that the deterministic, numeric-aware approach significantly reduces false positives compared to conventional token-based fuzzy matching. The integration of AI/ML as supportive augmentation layers enhances explainability and decision confidence without compromising determinism.**

Index Terms— **Data Reconciliation, Deterministic Fuzzy Matching, Numeric-Aware Matching, Explainable AI (XAI), GPT-4o-mini, N-Tier Architecture, JWT Authentication, Role-Based Access Control (RBAC), Flask, React, PostgreSQL**

I. INTRODUCTION

Enterprise systems frequently require reconciliation of structured datasets originating from different sources such as financial systems, inventory records, and operational logs. These datasets often contain textual descriptions combined with numeric identifiers that are semantically significant. Conventional fuzzy string-matching techniques prioritize textual similarity while ignoring numeric semantics, resulting in incorrect matches and high false-positive rates. Recent advances in Generative AI and Machine Learning, including OpenAI Large Language Models, enable new opportunities for explainability and decision transparency. However, if used without strict boundaries, AI models can compromise determinism and auditability.

This project demonstrates how deterministic algorithms, data science techniques, and AI-assisted explainability can be combined responsibly within a secure enterprise architecture. In large-scale enterprise environments, data reconciliation is not only a technical necessity but also a critical business requirement. Organizations depend on accurate data matching to ensure consistency across systems, support decision-making processes, and maintain compliance with regulatory standards. Even small mismatches in datasets can lead to financial discrepancies, reporting errors, and operational inefficiencies.

One of the major challenges in reconciliation arises from the heterogeneity of data sources. Different systems often follow distinct formats, naming

conventions, and data entry practices. For example, a single product may be represented differently across systems due to abbreviations, spelling variations, or formatting differences. In such cases, relying solely on textual similarity is insufficient, especially when numeric identifiers carry the true semantic meaning of the records.To overcome these limitations, it is essential to incorporate deterministic logic that gives higher priority to numeric attributes while still considering textual context. Deterministic approaches ensure that the same input consistently produces the same output, which is crucial for enterprise applications where reproducibility and auditability are mandatory.

Additionally, data preprocessing plays a significant role in improving reconciliation accuracy. Techniques such as data normalization, removal of special characters, case standardization, and tokenization help in reducing inconsistencies and preparing the data for effective comparison. These preprocessing steps enhance both deterministic and machine learning-based approaches. The integration of AI in this system is carefully controlled to maintain a balance between innovation and reliability. Instead of relying on AI for core matching decisions, it is used to provide explanations and insights into how matches are derived. This improves transparency and allows users to understand the reasoning behind each result, thereby increasing trust in the system. Overall, the proposed approach addresses the limitations of traditional methods by combining structured rule-based logic with modern AI capabilities. This results in a robust, scalable, and transparent reconciliation system suitable for enterprise-level applications.

Problem Statement

Despite advances in data processing tools, several challenges persist in data reconciliation:

1. Numeric Blindness – Traditional fuzzy matching ignores numeric identifiers embedded in descriptions
2. High False Positives – Text-only matching produces incorrect matches when amounts or codes differ
3. Lack of Explainability – Automated matching decisions lack human-readable justifications
4. Non-Determinism – AI-driven decision systems are often non-deterministic and hard to audit

This project addresses these challenges by providing a deterministic, numeric-aware matching system augmented with AI explainability.

Numeric blindness is one of the most critical limitations in traditional reconciliation systems. In real-world enterprise datasets, numeric values such as transaction amounts, invoice numbers, and product codes carry significant semantic meaning. Ignoring these values can lead to incorrect matches even when textual similarity appears high. High false positives are a direct consequence of relying solely on text-based similarity measures. For example, two records may share similar descriptions but represent entirely different transactions due to mismatched numeric values. This creates serious issues in domains like finance, where accuracy is crucial. Another major challenge is the lack of explainability in automated systems. Users often find it difficult to trust or validate system decisions when there is no clear reasoning provided. Without transparency, auditing and debugging become complex and time-consuming.

Non-determinism in AI-driven systems introduces inconsistency, where the same input may produce different outputs at different times. This is unacceptable in enterprise environments that require repeatable and auditable processes for compliance and reliability. Additionally, many existing tools suffer from architectural limitations, such as lack of scalability, weak security mechanisms, and absence of persistent data storage. These shortcomings make them unsuitable for enterprise-level deployment. To overcome these challenges, the proposed system integrates a deterministic matching engine with numeric-aware logic, ensuring consistent and accurate results. Furthermore, AI-based explainability enhances transparency by providing human-readable justifications, while a robust N-Tier architecture ensures scalability, security, and maintainability.

II. LITERATURE REVIEW

Research in fuzzy string matching has established techniques such as Levenstein distance, token-based matching, and phonetic algorithms. However, these approaches treat all characters equally without semantic awareness of numeric content. Studies on enterprise data reconciliation emphasize the importance of deterministic, auditable matching decisions. Financial and operational systems require

repeatability the same inputs must always produce the same outputs. Recent work on Explainable AI (XAI) demonstrates that natural language explanations improve user trust and decision transparency. Machine Learning classifiers can provide confidence indicators without replacing rule-based decision logic. The Rapid Fuzz library provides optimized implementations of fuzzy matching algorithms suitable for large-scale batch processing.

Fuzzy string-matching techniques such as Levenstein distance primarily focus on character-level edits, including insertions, deletions, and substitutions. While effective for measuring textual similarity, these methods fail to capture domain-specific semantics, especially when numeric values play a critical role. Token-based matching improves performance by comparing word-level similarities, but still lacks the ability to distinguish between semantically important numeric differences. Phonetic algorithms like Soundex and Metaphone are designed to match words based on pronunciation, which is useful in name matching applications. However, they are not suitable for structured enterprise data where numeric precision is essential.

In enterprise reconciliation systems, determinism is a key requirement. Systems must produce consistent and repeatable results to support auditing, compliance, and traceability. Non-deterministic approaches, particularly those relying entirely on AI models, may introduce variability, making them unsuitable as primary decision-making mechanisms in critical domains. Explainable AI (XAI) has emerged as an important field to address the transparency challenges of AI systems. By converting complex decision logic into human-readable explanations, XAI techniques help users understand why a particular match was accepted or rejected. This is especially important in financial and regulatory environments where accountability is mandatory.

Machine Learning models, such as Logistic Regression and Decision Trees, are widely used to estimate confidence scores in classification tasks. In the context of data reconciliation, these models can provide an additional layer of validation by assigning confidence levels to matches, without replacing deterministic logic. The Rapid Fuzz library offers a high-performance alternative to traditional fuzzy matching libraries like Fuzzy Wuzzy. It is optimized for speed and scalability, making it suitable for processing large datasets in enterprise environments. Its efficient implementations of string similarity algorithms enable real-time and batch reconciliation tasks with minimal computational overhead.

III. SYSTEM ARCHITECTURE

The system follows a four-layer N-Tier architecture, ensuring clear separation of concerns and scalability.
Presentation Layer: React.js with TypeScript frontend
Application Layer: Flask-based REST API with deterministic matching engine
Data Layer: PostgreSQL relational database AI/ML
Augmentation Layer: OpenAI API + scikit-learn classifier.

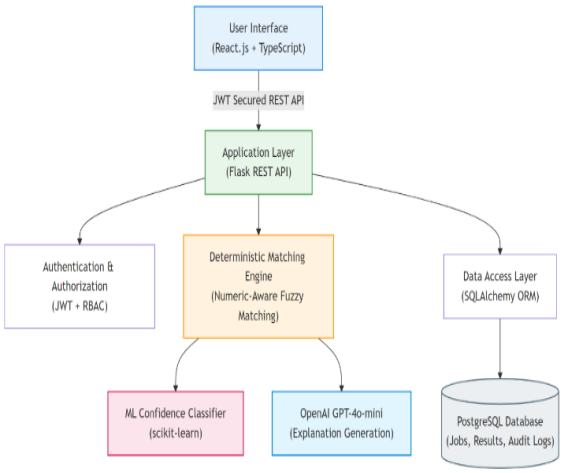
High-Level Architecture

Figure1 illustrates the high-level interaction between system components. User requests are routed through a secure REST API, processed by the deterministic matching engine, and optionally augmented with AI explanations.

The N-Tier architecture adopted in this system improves modularity, maintainability, and scalability by separating responsibilities across distinct layers. Each layer operates independently while communicating through well-defined interfaces, making the system easier to extend and manage. The Presentation Layer is responsible for user interaction and is implemented using React.js with TypeScript. It provides features such as file upload, configuration of matching parameters, visualization of results, and access to audit logs and AI-generated explanations. This layer ensures a responsive and user-friendly interface. The Application Layer acts as the core processing unit of the system. Built using Flask, it exposes RESTful APIs to handle client requests. The deterministic matching engine resides in this layer, where all reconciliation logic, validation, scoring, and decision-making processes are executed. This layer also manages authentication, authorization, and audit logging. The Data Layer uses PostgreSQL to store structured data including uploaded datasets, matching results, audit trails, AI explanations, and user information. The N-Tier architecture plays a crucial role in improving the overall system design by separating functionalities into independent layers. This separation ensures that changes in one layer, such as the user interface, do not affect the core business logic

or database operations. It enhances maintainability by allowing developers to update, debug, or scale individual components without impacting the entire system. This modular approach is especially important in enterprise applications where system reliability and flexibility are critical.

Moreover, the architecture supports scalability by enabling each layer to be deployed and scaled independently based on workload requirements. For example, the application layer handling the matching logic can be scaled to process large volumes of data, while the database layer ensures efficient storage and retrieval. The inclusion of an AI/ML augmentation layer further strengthens the system by adding intelligent features without interfering with deterministic processing. Overall, the N-Tier architecture provides a strong foundation for building secure, scalable, and high-performance data reconciliation systems.



IV. METHODOLOGY & IMPLEMENTATION

Numeric-Aware Fuzzy Matching Algorithm

The core matching algorithm combines textual similarity with explicit numeric comparison:

1. Text Similarity: Calculate token-based similarity Rapid Fuzz token_sort_ratio
2. Numeric Extraction: Extract all numbers from source and reference descriptions
3. Numeric Consistency Check: Verify source amount matches reference numbers within tolerance
4. Score Combination: Apply bonus for exact numeric match, penalty for mismatch
5. Threshold Decision: Accept match if final score exceeds configurable threshold

Formula:

Final Score = Text Score + Numeric Adjustment

Where:

- Numeric Adjustment = +20 (exact match) | +0 to +20 (within tolerance) | -50 (mismatch)

Deterministic Tie-Breaking

When multiple candidates have equal scores, deterministic tie-breaking ensures repeatability:

1. Prefer numeric-consistent matches over inconsistent
2. Prefer higher numeric score (exact match bonus)
3. Prefer higher text similarity score
4. Prefer stable ordering by reference code (lexicographic)

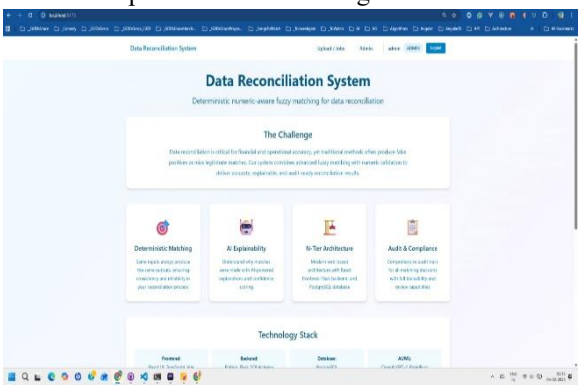
ML Confidence Classification

A Logistic Regression classifier predicts confidence levels based on six deterministic features:

- final_score — normalized composite matching score
- text_score — text similarity score (fuzzy string comparison)
- numeric_match — binary flag for exact numeric value match
- numeric_score — numeric proximity score
- length_diff — normalized character length difference
- token_overlap — proportion of shared tokens between strings

V. OUTPUTS

Snapshots Of the Working Model / Ui



The frontend user interface is developed using React.js with TypeScript, providing a responsive and user-friendly environment for interacting with the system.

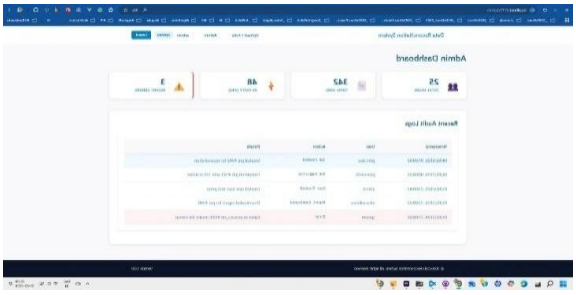
Key UI Features

- Login Page: Secure authentication using JWT-based login.
- Dashboard: Overview of all matching jobs and their statuses.
- File Upload Interface: Allows users to upload source and reference datasets (CSV/Excel).
- Job Monitoring: Displays real-time status of processing jobs.
- Results View: Shows matched records with scores and confidence levels (HIGH/MEDIUM/REVIEW).
- Audit Log View: Displays detailed logs for transparency and traceability.
- Export Option: Enables downloading results in Excel format.

The system delivers fast and consistent performance, with job creation, matching, and export operations completing in under a second. Graphs show linear scaling, confirming responsiveness and stability even as data volume increases. The system demonstrates strong backend stability with rapid job creation and matching execution. Results retrieval and export operations remain consistently fast, ensuring smooth user experience. Performance graphs confirm linear scaling across increasing data volumes without degradation. Overall, the backend is efficient, deterministic, and fully prepared for frontend integration.

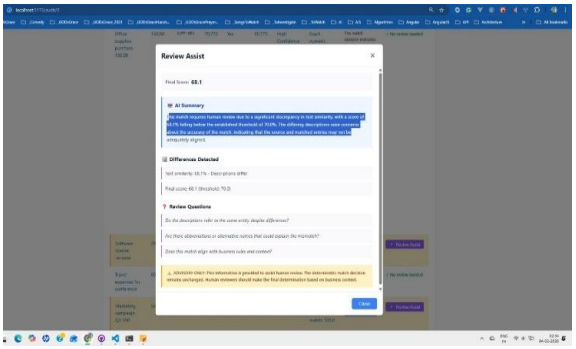
The Results page presents the output of the data reconciliation process in a structured and user-friendly format. It displays matched and unmatched records along with key details such as similarity scores, numeric validation status, and final decision outcomes. This enables users to quickly analyze the effectiveness of the matching algorithm and identify any discrepancies between source and reference data.

In addition, the page may include features such as filtering, sorting, and review indicators to help users focus on specific records that require attention. Integration with AI-generated explanations and confidence scores further enhances usability by providing insights into why certain matches were accepted or rejected. Overall, the Results page plays a critical role in validating system performance and supporting informed decision-making.



The Admin Dashboard prvides a centralized view of the system's performance and activities. It displays key metrics such as total users, total jobs, recent activity, and error count, allowing administrators to quickly understand the overall system status.

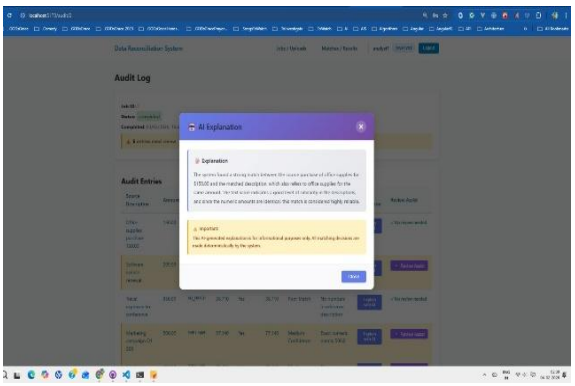
Additionally, the dashboard includes an audit log section that records user actions and system events with timestamps and details. This ensures transparency, helps in monitoring operations, and supports effective troubleshooting and system management.



This review flagged a discrepancy with a similarity score of 68.1, just below the 70 thresholds. Human judgment is advised to confirm whether the entries truly align despite differing descriptions.

The Audit Log page provides a detailed view of reconciliation activities for a specific job, including its status, completion time, and review requirements. It allows users to track how each record was processed and whether further review is needed.

Additionally, the page includes an AI Explanation feature that presents human-readable justifications for matching decisions. This improves transparency and helps users understand why a match was accepted or rejected, while maintaining that final decisions are made by the deterministic system.



Future Enhancements

- Integration with enterprise IAM solutions (Azure AD, Auth0)
- Analytics dashboards for reconciliation trends
- PDF export of audit reports
- Retrieval-Augmented Generation (RAG) for context-aware explanations
- Multi-currency and multi-format numeric handling
- Implementation of real-time data validation with external APIs
- Advanced anomaly detection using AI techniques
- Automated data cleaning and preprocessing modules
- Role-based dashboards with personalized insights
- Scalability improvements for handling large data.

VI. CONCLUSION

- This project demonstrates the successful integration of deterministic fuzzy matching with AI-assisted explainability and ML-based confidence analysis. By enforcing numeric consistency as a core matching rule, the system significantly reduces false positives compared to traditional text-only approaches.
- The N-Tier architecture ensures scalability, maintainability, and security. The strict separation between deterministic decision-making and AI augmentation preserves auditability while providing human-readable explanations.
- The project highlights the importance of explainability, determinism, and modern AI-assisted decision support in enterprise data reconciliation systems.
- Furthermore, the system improves overall operational efficiency by reducing manual effort in data matching and verification processes. This leads to faster decision-making and minimizes the risk of human errors in large-scale enterprise environments.
- In addition, the flexible and modular design of the system allows easy integration with future technologies and enterprise tools, making it adaptable to evolving business requirements and ensuring long-term usability.
- The inclusion of machine learning-based confidence scoring further enhances decision support by helping users identify records that require manual review, thereby improving accuracy and reliability.
- Overall, the proposed system provides a balanced approach by combining rule-based precision with AI-driven insights, making it a practical and effective solution for modern enterprise data reconciliation challenges.

REFERENCES

[1] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, “A comparison of string distance metrics for name-matching tasks,” in Proc. IJCAI Workshop Inf. Integr., 2003.

[2] G. Navarro, “A guided tour to approximate string matching,” ACM Comput. Surv., vol. 33, no. 1, pp. 31–88, 2001.

[3] M. Bachmann, “RapidFuzz: Fast string matching in Python.” [Online]. Available: https://maxbachmann.github.io/RapidFuzz/

[4] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD), 2016.

[6] C. Molnar, Interpretable Machine Learning. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[7] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Univ. California, Irvine, CA, USA, 2000.

[8] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison-Wesley, 2002.

[9] PostgreSQL Global Development Group, "PostgreSQL documentation." [Online]. Available: https://www.postgresql.org/docs/

[10] D. Hardt, "The OAuth 2.0 authorization framework," IETF RFC 6749, 2012.

[11] R. Sandhu et al., "Role-based access control models," IEEE Comput., vol. 29, no. 2, pp. 38–47, 1996.

[12] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Sov. Phys. Dokl., vol. 10, no. 8, pp. 707–710, 1966.

[13] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. (draft). Upper Saddle River, NJ, USA: Prentice Hall, 2023.

[14] OpenAI, "GPT models and API documentation." [Online]. Available: https://platform.openai.com/docs