FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 12, June, 2026, pp 242 – 251
DOI: https://doi.org/10.33003/fjs-2026-1012-5272

Formalizing Feedback as Reusable Knowledge in Data Preparation Systems: A Persistent Feedback Artefact Model

***[1,2]Sa'adatu Abdulkadir, [1]Philip O. Odion, [1]Darius T. Chinyio, [1]Isah R. Saidu and [3]Muhammad A. Ahmad**

[1]Department of Computer Science, Nigerian Defence Academy, Kaduna State, Nigeria.
[2]Department of Informatics, Kaduna State University, Kaduna State, Nigeria.
[3]Department of Secure Computing, Kaduna State University, Kaduna State, Nigeria.

*Corresponding authors' email: saa.abdul@kasu.edu.ng

Received: 07th July 2026
Accepted: 20th July 2026
Published: 31st July 2026

ABSTRACT
Feedback-driven data preparation systems use user corrections to improve data quality and transformation accuracy. However, existing approaches typically treat feedback as a transient interaction that is consumed during execution and discarded after task completion. This limits the reuse of corrective knowledge across datasets, users, and preparation cycles. This study proposes a Persistent Feedback Artefact Model (PeFAM) as a conceptual and formal representation for treating feedback as a reusable computational knowledge object. The model represents feedback as a structured artefact comprising the original value, corrected value, contextual attributes, metadata, timestamp information, and provenance. A context-aware similarity mechanism is defined to guide how feedback artefacts may be retrieved and assessed for reuse across preparation tasks. The model was assessed through a worked analytical scenario that examined representational completeness, traceability, contextual matching, and threshold-based eligibility for human review. The study contributes a formal representation of feedback knowledge, a similarity-based reuse mechanism, and a conceptual arrangement of functions supporting persistent feedback management in data preparation environments. The analytical demonstration indicates that the model can support representation, traceability, contextual comparison, and threshold-based eligibility within the defined scenario, while empirical evaluation of efficiency, accuracy, usability, and scalability remains necessary.

Keywords: Data Preparation, Persistent Feedback Artefacts, Feedback Reuse, Knowledge Representation, Context-Aware Retrieval, Human-in-the-Loop

INTRODUCTION

Data preparation remains one of the most resource-intensive stages of data analytics and data science. Activities such as data cleaning, transformation, integration, validation, and schema alignment are essential for ensuring data quality and analytical reliability, and preprocessing decisions can materially influence the performance of downstream machine-learning models (Wirth & Hipp, 2000; Azeroual, 2020; Fernandes *et al.*, 2023; Garba *et al.*, 2025). Despite advances in automation, many preparation decisions still depend on human judgement because data sources may be heterogeneous, incomplete, semantically ambiguous, or governed by domain-specific requirements (Konstantinou & Paton, 2020; Somasundaram, 2022).

Human-in-the-Loop and feedback-driven approaches have therefore become increasingly important in data preparation. In these systems, users guide automated processes by correcting erroneous values, validating proposed transformations, resolving ambiguities, or approving system recommendations (Wu *et al.*, 2023; Yin *et al.*, 2024). Feedback may be provided explicitly through corrections and annotations or inferred implicitly from user interactions and behavioural patterns (Narayan *et al.*, 2022; Shome *et al.*, 2024). Such approaches can improvepreparation outcomes by combining computational efficiency with contextual and domain knowledge that may not be captured adequately through predefined rules or statistical inference alone.

However, feedback in many existing systems is used primarily to improve the task in which it is generated. Once the immediate preparation activity has been completed, the correction may remain confined to the current session, workflow, or dataset rather than being retained for subsequent use (Konstantinou & Paton, 2020; Abdulkadir *et al.*, 2026). Consequently, when similar data-quality problems arise in later datasets, users may be required to repeat corrections that have already been performed and validated. This limits the accumulation of transformation knowledge, increases repeated manual effort, and restricts the ability of data preparation systems to improve progressively across preparation cycles.

A major reason for this limitation is the absence of a formal representation through which feedback can be preserved as reusable computational knowledge. Existing research has largely concentrated on how feedback is captured and applied during task execution, with less attention given to how a correction should be structured, stored, interpreted, retrieved, and assessed for use in a different but related context. Retaining only the corrected value is insufficient because the applicability of a correction may depend on the attribute involved, data type, source dataset, transformation purpose, user role, validation status, time of creation, and other contextual or provenance-related information.

This study addresses this gap by proposing the Persistent Feedback Artefact Model (PeFAM), which represents feedback as a structured computational artefact comprising the original value, corrected value, contextual attributes, metadata, timestamp information, and provenance. The model is intended to preserve both the corrective action and

the information required to interpret its relevance beyond the task in which it was created. A context-aware similarity mechanism is also defined to support the retrieval and assessment of stored artefacts when comparable conditions are encountered in later data preparation activities.

The scope of the study is conceptual and formal rather than empirical. It focuses on the representational and logical requirements for persistent feedback management and does not propose a machine-learning algorithm or claim empirical performance improvements. PeFAM instead provides a formal basis upon which persistent repositories and feedback-reuse mechanisms may subsequently be implemented and evaluated. Accordingly, the study contributes: (i) a structured representation of feedback as a persistent computational artefact; (ii) a context-aware similarity mechanism for retrieval and applicability assessment; (iii) an artefact lifecycle and repository-function model supporting controlled reuse; and (iv) an analytical demonstration of representational adequacy, traceability, contextual comparison, and threshold-based eligibility for human review.

Related Work

Data preparation involves transforming raw, incomplete, inconsistent, or heterogeneous data into forms suitable for analysis. Common activities include cleaning, transformation, integration, validation, and schema alignment (Azeroual, 2020). Traditional automation relies mainly on deterministic extraction–transformation–loading pipelines, predefined rules, and scripted workflows. These approaches are effective when data structures and transformation requirements remain stable, but they are less adaptable when schemas evolve, data sources are heterogeneous, or preparation decisions require contextual and domain-specific interpretation (Paton, 2019; Konstantinou & Paton, 2020; Somasundaram, 2022).

Advances in machine learning and intelligent data wrangling have expanded the range of preparation tasks that can be automated. Contemporary systems can infer transformationpatterns, recommend cleaning operations, assist with schema matching, and generate preparation logic from examples or natural-language instructions (Narayan *et al.*, 2022; Liu *et al.*, 2024). Nevertheless, these approaches generally concentrate on improving the execution of the current task. Corrective knowledge produced during user interaction is often embedded in temporary execution histories, learned parameters, prompts, or task-specific transformation sequences rather than preserved as an independent and reusable knowledge object.

Human-in-the-Loop approaches address some limitations of fully automated preparation by allowing users to validate outputs, correct errors, resolve ambiguities, and refine proposed transformations (Rezig *et al.*, 2019; Konstantinou *et al.*, 2019; Wu *et al.*, 2023). Feedback may be explicit, through corrections, annotations, approvals, or rejections, or implicit, through interaction patterns and behavioural traces (Narayan *et al.*, 2022; Shome *et al.*, 2024). These mechanisms enable systems to incorporate knowledge that may not be inferred reliably from structural or statistical properties alone. However, much of the existing work uses feedback primarily as an immediate supervisory signal rather than as a persistent resource that remains available after task completion.

Preserving feedback for later reuse requires more than recording the corrected value. The circumstances under which a correction was made must also be retained because a transformation that is valid in one dataset may be inappropriate in another. Relevant context may include the affected attribute, data type, source dataset, transformation objective, domain, workflow stage, and operational conditions. Contextual representation therefore supports the assessment of whether a historical correction is applicable to a current preparation problem (Konstantinou & Paton, 2020; Narayan *et al.*, 2022; Liu *et al.*, 2024).

Metadata and provenance provide additional information required for reliable interpretation and reuse. Metadata can describe the correction type, transformation category, confidence level, validation status, source system, and user role, thereby supporting classification, indexing, filtering, and retrieval. Provenance records the origin and history of a correction, including who generated it, where it originated, and under what conditions it was validated. Such information supports traceability, accountability, and trust when previously captured feedback is considered for use in another task (Herschel *et al.*, 2017; Singh *et al.*, 2019; Sambasivan *et al.*, 2021). Timestamp information is also important because the validity of corrections and organisational rules may change over time.

Existing systems provide only limited support for combining these representational requirements within a reusable feedback structure. Transformation histories, scripts, annotations, interaction logs, and learned models may preserve aspects of previous preparation activity, but they do not necessarily bind the correction to the contextual, descriptive, temporal, and provenance information needed for future applicability assessment. Similarly, storing feedback without a retrieval mechanism does not establish whether it can be reused safely. Effective reuse requires structured representation, persistent storage, contextual matching, and a controlled decision process through which a retrieved correction is assessed before application.

A systematic review of feedback-driven data preparation found that feedback was used predominantly in data cleaning, schema matching, transformation, and validation. Although many reviewed approaches reported improvements in immediate preparation outcomes, support for persistent storage, contextual retrieval, and cross-dataset reuse remained limited (Abdulkadir *et al.*, 2026). Consequently, corrective knowledge often remains confined to individual sessions, datasets, or workflows, requiring users to repeat previously validated interventions when similar data-quality problems recur.

Existing representations preserve different parts of prior data-preparation activity, but they do not necessarily perform the same function as PeFAM. Interaction histories and transformation sequences can record what actions were previously performed, while provenance records support the tracing of data and decision origins (Herschel et al., 2017; Singh et al., 2019). Human-in-the-Loop systems may also retain annotations, corrections, approvals, or interaction traces for immediate learning or task execution (Rezig et al., 2019; Konstantinou et al., 2019; Narayan et al., 2022). However, these representations do not always bind a validated correction to the contextual, descriptive, temporal, and provenance information required to assess its applicability in a later task. Table 1 summarises this distinction.

Table 1: Comparison of Feedback-Related Representations

Representation	Preserves corrective action	Preserves context	Preserves time/provenance	Supports later applicability assessment	Supports controlled reuse
Interaction or correction history	Yes	Limited	Sometimes	Limited	Limited
Transformation sequence or script	Yes	Usually implicit	Limited	Rule-dependent	Yes, where conditions remain valid
Provenance record	Partly	Partly	Yes	Not its principal function	No direct reuse mechanism
HITL annotation or feedback record	Yes	Task-dependent	Task-dependent	Usually task-specific	Limited across cycles
PeFAM	Yes	Yes	Yes	Yes	Yes, subject to review

The comparison does not suggest that PeFAM replaces transformation histories, provenance systems, or Human-in-the-Loop feedback mechanisms. Rather, PeFAM combines the corrective decision with the contextual and traceability information required to support later retrieval and controlled applicability assessment.

The remaining gap is therefore not simply the capture of feedback, but its formalisation as a persistent computational knowledge object. Existing research has not established a widely adopted representation that integrates the original state, validated correction, context, metadata, temporal information, and provenance within a single artefact designed for later retrieval and reuse. PeFAM addresses this gap by structuring these elements as a reusable artefact and defining a context-aware mechanism for assessing stored feedback when comparable preparation conditions arise.

MATERIALS AND METHODS

Model-Development Approach

This study adopted a conceptual model-development approach to formalise user feedback as reusable computational knowledge in data preparation. The approach was appropriate because the purpose of the study was to define the structure, relationships, and reuse logic required for persistent feedback management rather than to evaluate an implemented platform or measure performance outcomes.

The development of the Persistent Feedback Artefact Model (PeFAM) followed seven related stages:

i. identification of representational limitations in existing feedback-driven data preparation systems;
ii. derivation of design requirements for persistent feedback management;
iii. specification of the components required to represent a feedback artefact;
iv. formalisation of the artefact as a computational structure;
v. definition of a context-similarity and applicability mechanism for retrieval;
vi. specification of the artefact lifecycle, repository functions, and conceptual architecture; and
vii. analytical demonstration of the model using a representative data-preparation scenario.

To improve traceability between the identified research problem and the resulting model, the limitations derived from the literature were mapped to the design requirements and corresponding PeFAM elements. Overlapping observations were consolidated into four minimum requirements: persistence, context awareness, traceability, and reusability. Each requirement was retained only where it addressed a distinct limitation and could be represented through a model component, repository function, or applicability mechanism. Table 2 presents this derivation.

Table 2: Traceability from Identified Limitations to PeFAM design Elements

Identified limitation	Design requirement	PeFAM response
Corrections frequently remain confined to the task or session in which they were generated	Persistence	Repository-ready feedback artefact retained beyond the execution cycle
The same correction may not be appropriate under different operational or semantic conditions	Context awareness	Context component (C) and contextual-similarity mechanism
The origin, timing, authority, and validation history of a correction may be unclear	Traceability	Metadata (M), timestamp (t), and provenance (p)
Stored corrections cannot be safely considered in later tasks without applicability assessment	Reusability	Candidate retrieval, similarity threshold, human review, and outcome recording

This mapping guided the selection of the six artefact components and the definition of the lifecycle, repository, and retrieval mechanisms. It also provides a basis for assessing whether each identified requirement is addressed by the model.

The stages were sequential but interdependent. Limitations identified in the literature informed the design requirements, while the design requirements guided the selection of artefact components and the formulation of the retrieval and reuse mechanism. The resulting artefact structure, lifecycle, repository model, and architecture were subsequently examined through a worked scenario to determine whether the proposed representation could preserve the information required for persistence, traceability, contextual interpretation, retrieval, and controlled reuse.

This process produced a conceptual and formal model rather than a software implementation. Accordingly, the analytical demonstration examines the internal coherence and representational adequacy of PeFAM; it does not constitute empirical validation of system performance.

Design Requirements

The model requirements were derived from limitations identified in prior work on feedback-driven data preparation. Existing approaches commonly capture and apply user corrections during an active task but provide limited support for preserving those corrections together with the information required for later interpretation and reuse. The systematic review underlying this study similarly identified limited support for persistent storage, contextual retrieval, and cross-dataset reuse of corrective knowledge.

Four requirements were therefore defined:

Persistence

Feedback must remain available beyond the execution cycle in which it is generated.

Context Awareness
A stored correction must retain sufficient information about the conditions under which it was made so that its relevance to a later task can be assessed.

Traceability
The representation must preserve temporal and provenance information describing when, where, and by whom the correction was generated.

Reusability
The representation must support retrieval and consideration of a validated correction when sufficiently similar conditions arise in a subsequent preparation task.
These requirements guided the selection of the model components and the definition of the retrieval, repository, and lifecycle mechanisms. They also distinguish PeFAM from a conventional correction log, which may preserve a historical change without retaining sufficient information for later applicability assessment. Reuse control is treated as an analytical dimension of the reusability requirement rather than as a separate design requirement. Reusability therefore includes both the identification of potentially relevant artefacts and the prevention of unconditional application.

Feedback Artefact Specification
Based on the design requirements, a feedback artefact was specified as a structured representation that binds the corrective action to the contextual and traceability information required for future interpretation. PeFAM is formally defined as:
$FA = (v_o, v_c, C, M, t, p)$
Where:

i. **v_o** represents the original data value, attribute value, or transformation rule before correction.
ii. **v_c** represents the corrected value or modified transformation rule.
iii. **C** denotes contextual information describing the operational environment in which the correction occurred.
iv. **M** denotes metadata associated with the feedback instance.
v. **t** denotes the timestamp indicating when the correction was generated.
vi. **p** denotes provenance information describing the origin of the feedback.

The six components are grouped into two categories. The corrective components, v_o and v_c, preserve what was changed. The supporting knowledge components, C, M, t, and p, preserve the information required to explain, classify, trace, retrieve, and assess the correction. The role of each component is as follows:

Original Value (v_o)
This records the data item or transformation state that prompted user intervention. It provides the reference condition from which the corrective action was derived.

Corrected Value (v_c)
This records the validated outcome of the intervention. It represents the corrective knowledge that may be considered for later reuse.

Context (C)
This records the operational and semantic conditions under which the correction occurred. Contextual attributes may include the field name, data type, value pattern, transformation category, source characteristics, domain convention, and workflow stage. Context supplies the principal basis for determining whether an earlier correction is relevant to a later data instance.

Metadata (M)
This describes characteristics of the correction itself, including correction type, validation status, processing category, user role, or confidence information. Metadata supports artefact classification, indexing, filtering, and lifecycle management.

Timestamp (t)
This records when the correction was generated. Temporal information supports auditing, version control, historical analysis, and assessment of whether a stored correction remains current.

Provenance (p)
This records the source and history of the artefact, including the originating dataset, user, workflow, organisational unit, or processing environment. Provenance supports accountability and informed assessment of the authority and relevance of stored feedback.
Figure 1 presents the six-component structure of PeFAM and distinguishes the corrective components from the supporting knowledge components.

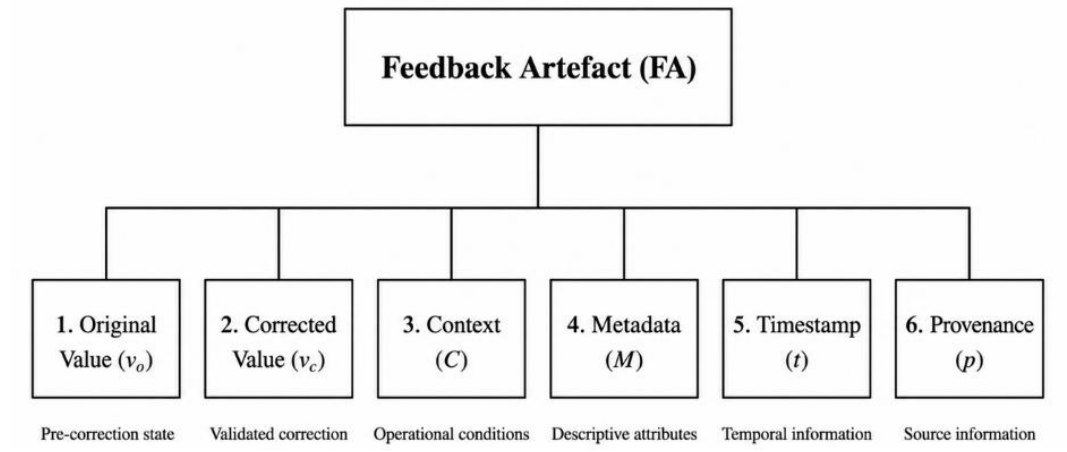

Figure 1: Structure of the PeFAM

Context-Similarity and Applicability Mechanism

A stored correction should not be reused solely because the same or a similar value appears in another dataset. Its applicability depends on the degree of correspondence between the context of the current data instance and the context retained in the feedback artefact.

Let C_n represent the context of a new data instance and C_f represent the context associated with a stored feedback artefact. Their contextual similarity is expressed as:

$$Sim(C_n, C_f) = \sum_{i=1}^{k} w_i \cdot \text{sim}_i(c_{ni}, c_{fi})$$

Subject to:

$$\sum_{i=1}^{k} w_i = 1, w_i \geq 0$$

Where $\text{sim}_i(c_{ni}, c_{fi})$ is the similarity score for the i^{th} contextual attribute and w_i is the weight assigned to that attribute.

The attribute-level function may use exact matching, pattern matching, categorical comparison, or normalised distance depending on the type of contextual information being compared. For example, field names and transformation categories may use exact or semantic matching, while value patterns may require rule-based or distance-based comparison.

A stored artefact is considered eligible for reuse when:

$$Sim(C_n, C_f) \geq \theta$$

Where θ is the minimum contextual-similarity threshold specified for the task.

The threshold determines whether an artefact is eligible for consideration rather than whether it should be applied automatically. Where several artefacts satisfy the threshold, they may be ranked by similarity, but the retrieved artefacts should remain subject to user review because contextual similarity may not capture all domain-specific conditions.

Artefact Lifecycle and Repository Functions

The artefact lifecycle specifies how feedback progresses from a user correction to a reusable knowledge object. It comprises six stages:

Feedback Generation

A user corrects, validates, rejects, or modifies a value or transformation during a preparation task.

Artefact Creation

The corrective action and its associated context, metadata, timestamp, and provenance are encoded according to Equation (1).

Repository Storage

The complete artefact is retained in persistent storage and indexed using relevant contextual and descriptive attributes.

Similarity-based Retrieval

When a later data instance is encountered, its context is compared with the contexts of candidate artefacts using Equation (2).

Review and Selective Reuse

An eligible artefact is presented for assessment and may be accepted, rejected, modified, or selectively reused where appropriate.

Outcome Recording and Repository Update

The review decision and resulting outcome are recorded, and the artefact lifecycle status is updated where necessary.

Figure 2 presents the progression from feedback generation to reviewed reuse, outcome recording, and repository update. The persistent repository supports four principal functions:

i. Artefact storage: Retains the complete artefact beyond the task in which it was generated.
ii. Artefact indexing: Organises artefacts using contextual, metadata, temporal, and provenance attributes.
iii. Similarity-based retrieval: Identifies candidate artefacts whose contexts correspond sufficiently with a new data instance.
iv. Lifecycle management: Supports validation status, versioning, updating, deprecation, and outcome recording as artefacts accumulate.

These functions distinguish the repository from a historical correction log by retaining structured corrective knowledge that can be traced, searched, assessed, and selectively reused.

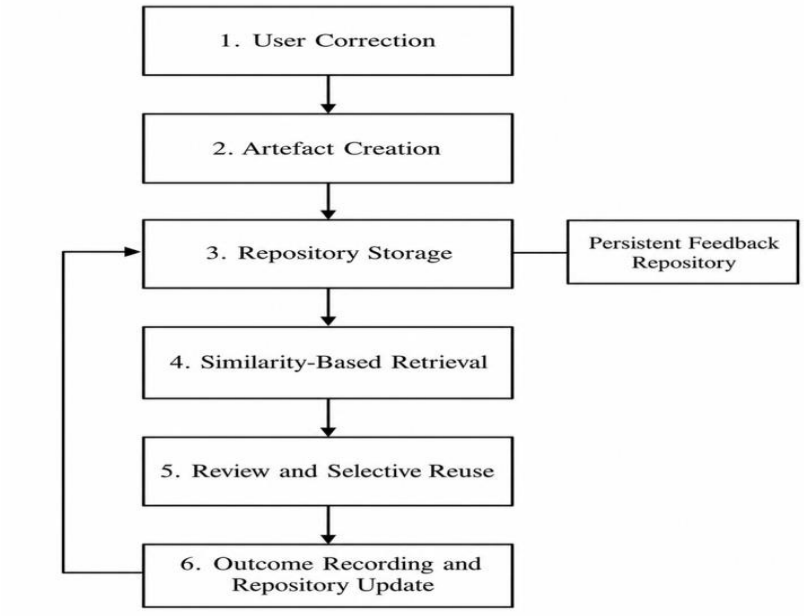

Figure 2: Lifecycle of a feedback artefact from creation to reviewed reuse and outcome recording

Conceptual Functional Layers of PeFAM

The functional arrangement supporting PeFAM groups the artefact specification and reuse logic into four interacting conceptual layers:

Feedback Capture Layer

Captures corrections, annotations, approvals, rejections, and validation actions produced during data preparation.

Artefact Management Layer

Converts captured feedback into the PeFAM structure and manages its context, metadata, temporal information, provenance, and lifecycle status.

Feedback Repository Layer

Provides persistent storage, indexing, search, and candidate retrieval facilities for feedback artefacts.

Reuse and Adaptation Layer

Extracts the context of a new data instance, computes contextual similarity, identifies eligible artefacts, presents relevant corrections for review, and records the reuse outcome.

Together, the four layers identify the principal functions required for a user correction to be captured, formalised, retained, retrieved, reviewed, and selectively reused during a later preparation task. The layers are conceptual and do not constitute an end-to-end software architecture or implementation framework. Figure 3 presents this functional arrangement.

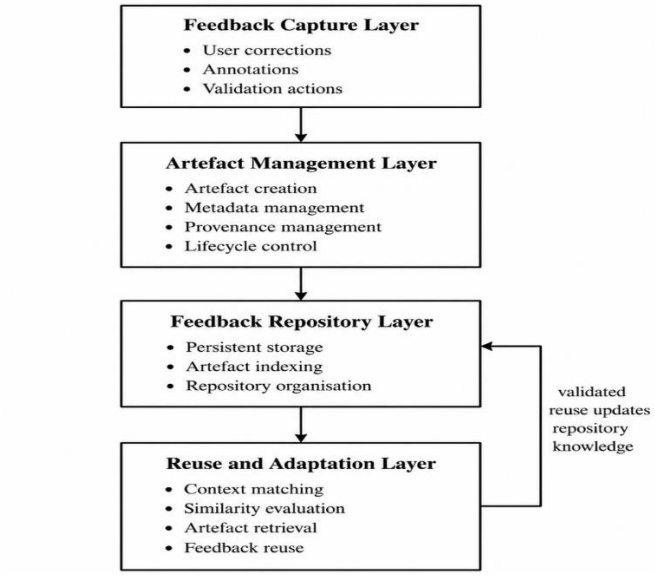

Figure 3: Conceptual functional layers supporting the PeFAM lifecycle

Analytical Demonstration Procedure

Because PeFAM is a conceptual and formal model, its initial assessment was conducted through a worked analytical scenario rather than through implementation-based performance testing. The demonstration was designed to examine whether the proposed structure could:

i. represent both a correction and the original condition that prompted it;
ii. preserve context, metadata, timestamp, and provenance;
iii. support comparison between a stored context and a new data context;
iv. determine eligibility using the similarity threshold;
v. prevent reuse where important contextual conditions differ; and
vi. Retain a traceable record of the reuse decision.

The procedure comprised the following steps:

i. select a representative data-preparation correction;
ii. instantiate the correction as a PeFAM artefact;
iii. define the contextual attributes relevant to its applicability;
iv. construct a later data instance presenting a comparable preparation problem;
v. compute the contextual-similarity score;
vi. compare the score with the predefined threshold;
vii. determine whether the artefact is eligible for user review and possible reuse; and
viii. Examine the artefact against the design requirements of persistence, context awareness, traceability, and reusability.

The worked example and analytical assessment are presented in Section 4. The procedure examines the representational completeness and logical applicability of PeFAM within thedefined scenario. It does not establish efficiency, accuracy, scalability, usability, or generalisable system performance, which require implementation and empirical evaluation.

RESULTS AND DISCUSSION

Model Demonstration and Analytical Validation

This section demonstrates the operation of the Persistent Feedback Artefact Model (PeFAM) using a representative data-preparation scenario. The purpose is to assess whether the proposed model provides the information and logical mechanisms required for feedback persistence, traceability, contextual retrieval, and controlled reuse. The demonstration is analytical rather than empirical: it examines the internal operation and representational adequacy of the model but does not measure system accuracy, efficiency, usability, or scalability.

Validation Criteria

PeFAM was assessed using five analytical criteria derived from the four design requirements established in Section 3.2. Persistence, context awareness, and traceability correspond directly to their respective requirements, while reusability was examined through two related criteria: contextual retrieval and reuse control. Table 3 presents the resulting criteria.

Table 3: Analytical Validation Criteria for PeFAM

Design requirement	Analytical criterion	Evidence required
Persistence	The correction can be retained independently of the task in which it was generated	A structured artefact containing the correction and supporting information is available for repository storage
Context awareness	The artefact preserves information needed to interpret the correction in a later task	Context includes the affected attribute, data type, domain convention, dataset category, and transformation purpose.
Traceability	The origin and history of the correction can be identified	Timestamp, originating dataset, user or role, validation status, and correction type are retained.
Reusability	A later data instance can be compared with the stored artefact before reuse.	Contextual similarity is calculated and evaluated against an applicability threshold.
Reuse control	An artefact that satisfies the threshold is made eligible for review rather than applied indiscriminately.	The retrieval decision distinguishes eligibility from final application.

The criteria assess whether the model contains the minimum structural and logical elements required to support persistent feedback management. They do not establish that the model improves performance in an implemented environment.

Worked Scenario: Creation of a Feedback Artefact

Consider a student-registration dataset containing a date-of-birth value recorded as 03/04/2006. The dataset follows the UK convention in which dates are interpreted as day/month/year. The intended interpretation is therefore 3 April 2006, not 4 March 2006.

Suppose the user standardises the value from the ambiguous numeric representation to the unambiguous textual form:

03/04/2006 → 3 April 2006

Using Equation (1), the resulting feedback artefact is represented as:

$FA = (v_o, v_c, C, M, t, p)$

Table 4 presents an illustrative instantiation of the feedback artefact for the worked scenario.

Table 4: Instantiated Feedback Artefact for Date-Format Standardisation

PeFAM component	Instantiated value
Original value, v_o	03/04/2006
Corrected value, v_c	3 April 2006
Context, C	Date-of-birth attribute; date data type; UK day-month-year convention; student-registration dataset; date-format standardisation
Metadata, M	Explicit user correction; formatting correction; user validated; approved for repository storage
Timestamp, t	Illustrative time of correction
Provenance, p	Student Registration Dataset A; registration-data preparation task; authorised reviewer

The artefact retains both the corrective action and the information required to interpret it. The original and corrected values record what changed, while the contextual attributes indicate why the correction was appropriate. Metadata records the nature and status of the intervention, the timestamp records when it occurred, and provenance identifies its origin.

The example satisfies the representational requirements for persistence, context awareness, and traceability. However, the artefact should not be reused solely because the same numeric value appears in another dataset, since its interpretation may differ under another date convention.

Contextual Retrieval in a Compatible Case

Assume that a later dataset, Student Registration Dataset B, contains the same value in a date-of-birth attribute. The later instance has the following contextual characteristics:

i. attribute: Date of Birth;
ii. data type: date;
iii. convention: UK day-month-year;
iv. dataset category: student registration;
v. Transformation purpose: date-format standardisation.

For demonstration, four contextual attributes are used in the similarity calculation: field role; data type; date convention; and transformation type.

Equal weights are assigned:

$w_1 = w_2 = w_3 = w_4 = 0.25$

Therefore,

$\sum_{i=1}^{4} w_i = 1$

Because the stored and current contexts match on all four attributes, the attribute-level similarity scores are:

$\text{sim}_1 = \text{sim}_2 = \text{sim}_3 = \text{sim}_4 = 1.0$

Using Equation (2):

$\text{Sim}(C_n, C_f) = (0.25 \times 1.0) + (0.25 \times 1.0) + (0.25 \times 1.0) + (0.25 \times 1.0) = 1.00$

Assume the applicability threshold is $\theta = 0.80$. Because $1.00 \geq 0.80$, the stored artefact is eligible for retrieval and user review.

The outcome does not mean that the correction is automatically applied. It means that the model has identified the stored artefact as contextually relevant enough to be presented as a candidate. The analyst may then confirm, reject, or modify the proposed correction before it is used in the current task.

Contextual Rejection in an Incompatible Case

Consider a second later dataset in which 03/04/2006 occurs in a date-of-birth field, but the dataset uses the US month-day-year convention. In that context, the intended interpretation is: March 4, 2006. The field role, data type, and transformation type remain compatible, but the date convention differs. The contextual similarity scores for the incompatible case are presented in Table 5.

Table 5: Contextual Similarity Scores for the Incompatible Case

Contextual attribute	Weight	Similarity
Field role	0.25	1.00
Data type	0.25	1.00
Date convention	0.25	0.00
Transformation type	0.25	1.00

As shown in Table 5, the field role, data type, and transformation type match the stored artefact, while the date convention does not match. The resulting similarity score is calculated as follows:

$Sim(C_n, C_f) = (0.25 \times 1.0) + (0.25 \times 1.0) + (0.25 \times 0.0) + (0.25 \times 1.0) = 0.75$

Because $0.75 < 0.80$, the stored artefact does not satisfy the applicability threshold and is therefore excluded from reuse consideration.

This case demonstrates why value matching alone is insufficient. The same original value occurs in both datasets, but the contextual difference changes the appropriate interpretation. The similarity mechanism prevents a correction generated under the UK convention from being recommended in a US date-format context.

Artefact Lifecycle Demonstration

The worked scenario can also be traced through the five-stage artefact lifecycle:

Stage 1 - Feedback Generation

The user identifies that 03/04/2006 should be represented unambiguously as 3 April 2006 under the applicable convention.

Stage 2 - Artefact Creation

The original value, corrected value, context, metadata, timestamp, and provenance are encoded as a PeFAM artefact.

Stage 3 - Repository Storage

The complete artefact is retained in the feedback repository and indexed by field role, data type, date convention, dataset category, and transformation type.

Stage 4 - Similarity-Based Retrieval

When a comparable value is encountered in a later dataset, the context of the new instance is compared with the stored artefact.

Stage 5 - Review and Selective Reuse

In the compatible case, the score of 1.00 makes the artefact eligible for review. In the incompatible case, the score of 0.75 excludes it. Eligible artefacts remain subject to acceptance, rejection, or modification by the user.

Stage 6 – Outcome Recording and Repository Update

The final decision is recorded, together with any resulting validation, modification, versioning, or deprecation action.

The demonstration therefore traces a correction from its initial creation to a later applicability decision and outcome record without assuming unconditional reuse.

Analytical Validation Outcome

Table 6 summarizes the outcome of assessing the model against the criteria established in Section 4.1.

Table 6: Analytical Validation Outcome

Criterion	Demonstrated outcome	Assessment
Correction representation	Original and validated values are represented explicitly	Satisfied
Context preservation	Attribute role, data type, convention, dataset category, and transformation purpose are retained	Satisfied
Metadata support	Correction type and validation status are represented	Satisfied
Temporal traceability	A timestamp is included	Satisfied
Provenance traceability	Originating dataset, task, and reviewer role are retained	Satisfied
Persistent storage readiness	The artefact is structurally suitable for repository storage and indexing	Satisfied conceptually
Contextual retrieval	A weighted similarity score is computed for a later instance	Satisfied
Applicability control	The threshold distinguishes compatible and incompatible cases	Satisfied
Human oversight	Threshold satisfaction leads to review eligibility rather than unconditional application	Satisfied conceptually
Empirical performance	Efficiency, accuracy, usability, and scalability are measured	Not assessed in this study

The demonstration indicates that PeFAM can represent a correction together with the supporting information required for persistence, traceability, contextual comparison, and controlled reuse within the defined scenario. It also shows that the similarity threshold can distinguish between two cases containing the same original value but operating under different date conventions.

The analytical outcome is limited to the defined scenario and assumed similarity weights. It indicates logical consistency and representational adequacy within the defined scenario, but it does not establish the optimal choice of contextual attributes, weights, similarity functions, or thresholds. Those parameters require implementation-based testing across multiple datasets and preparation tasks.

Discussion

The analytical demonstration clarifies the contribution of the Persistent Feedback Artefact Model (PeFAM). PeFAM does

not introduce context, metadata, timestamp information, or provenance as new concepts in isolation. Its contribution lies in binding these elements to the original and corrected values within a single artefact designed specifically for later retrieval and reuse in data preparation. This structure addresses a limitation of approaches that retain only a correction, transformation history, interaction log, or task-specific rule without preserving sufficient information to interpret whether the earlier decision remains applicable in a different context.

The worked scenario also shows why persistence alone is insufficient. Storing a correction does not make it safely reusable unless the conditions surrounding that correction are also represented. In the date-format example, the same value could require different interpretations under UK and US conventions. The contextual similarity mechanism therefore serves as an applicability filter rather than as a guarantee of correctness. Its purpose is to identify artefacts that are sufficiently similar to warrant consideration while excluding artefacts whose contexts differ in important respects. This distinguishes PeFAM from simple value matching or unconditional rule replication.

A further implication concerns the role of human oversight. Although the model supports similarity-based retrieval, satisfying the threshold makes an artefact eligible for review rather than automatically applicable. This is important because no formal similarity function can be assumed to capture every semantic, organisational, or domain-specific consideration. The model therefore supports a controlled form of reuse in which computational matching narrows the candidate set while the user retains responsibility for accepting, rejecting, or modifying the proposed correction. This reviewed-reuse principle is consistent with Human-in-the-Loop data preparation, where automated support complements rather than replaces contextual judgement.

PeFAM also provides a conceptual basis for organisational memory in data preparation. When validated corrections are stored as traceable artefacts, knowledge generated during one task need not remain confined to the user, dataset, or session in which it originated. Over time, a repository may accumulate a structured history of corrective decisions that can be searched and considered during related tasks. However, the present study establishes only the representational and logical basis for such accumulation. It does not demonstrate that repository growth will necessarily reduce effort, improve accuracy, or produce cumulative learning in practice. Those outcomes depend on implementation quality, artefact relevance, repository governance, and the suitability of the retrieval mechanism.

The model is complementary to rule-based, machine-learning, and large-language-model-assisted approaches. Rule-based systems may use PeFAM artefacts to preserve the context and provenance of validated rules. Machine-learning systems may use them as structured supervisory records, while LLM-assisted systems may use them in retrieval, prompt construction, correction-history management, or post-generation review. In each case, PeFAM functions as a representation layer rather than as a replacement for the underlying automation technique. Its value lies in making user corrections explicit, traceable, and available for later contextual assessment.

The analytical demonstration nevertheless has important limitations. First, it uses a single illustrative scenario and does not establish that the proposed structure is sufficient for every type of data-preparation task. More complex activities, including schema mapping, multi-column transformations, entity resolution, and domain-specific validation, may require additional contextual attributes or relations among artefacts. Second, the weights and threshold used in the demonstration are illustrative rather than empirically optimised. Their appropriate values may vary across domains, datasets, transformation types, and risk levels. Third, the model does not yet specify comprehensive procedures for resolving conflicting artefacts, assessing artefact quality, managing outdated corrections, or assigning trust to feedback from users with different levels of authority or expertise.

Further work should therefore implement PeFAM within an operational data-preparation environment and evaluate it using multiple datasets, users, task types, and reuse conditions. Such evaluation should examine retrieval precision, inappropriate-reuse rates, completion time, residual errors, manual interventions, usability, scalability, repository growth, and the effect of different similarity functions and thresholds. It should also investigate artefact versioning, conflict resolution, validation status, access control, and deprecation policies. These studies are necessary before claims can be made about measurable efficiency, data-quality improvement, or generalisable feedback-reuse performance.

CONCLUSION

This study proposed the Persistent Feedback Artefact Model (PeFAM) as a formal representation for preserving user corrections as reusable computational knowledge in data preparation. The model binds the original value, corrected value, context, metadata, timestamp information, and provenance within a single structured artefact and defines a context-aware similarity mechanism for identifying potentially relevant artefacts in later preparation tasks. The worked demonstration showed that the model can distinguish between contextually compatible and incompatible cases and can support retrieval for human review rather than unconditional application. PeFAM therefore provides a formal representational foundation and a conceptual arrangement of functions for persistent feedback repositories and controlled reuse across preparation cycles. However, the study establishes representational adequacy and logical applicability only; it does not demonstrate empirical improvements in efficiency, accuracy, usability, scalability, or cumulative learning. These outcomes require implementation and controlled evaluation across multiple datasets, users, domains, and task conditions.

REFERENCES

Abdulkadir, S., Odion, P. O., Chinyio, D. T., Saidu, I. R., & Ahmad, M. A. (2026). Feedback-driven automation in data preparation: A systematic literature review. *Science World Journal (SWJ), 21*(1), 290-303. https://doi.org/10.4314/swj.v21i1.39

Azeroual, O. (2020). Data wrangling in database systems: Purging of dirty data. *Data*, *5*(2), Article 50. https://doi.org/10.3390/data5020050

Fernandes, A. A. A., Koehler, M., Konstantinou, N., Pankin, P., Paton, N. W., & Sakellariou, R. (2023). Data preparation: A technological perspective and review. *SN Computer Science*, *4*(4), Article 425. https://doi.org/10.1007/s42979-023-01828-8

Garba, M., Usman, M., & Saidu, M. (2025). Enhancing employee attrition prediction: The impact of data preprocessing on machine learning model performance. *FUDMA Journal of Sciences*, *9*(1), 205–210. https://doi.org/10.33003/fjs-2025-0901-3030

Herschel, M., Diestelkämper, R., & Lahmar, H. B. (2017). A survey on provenance: What for? What form? What from? *The VLDB Journal*, *26*(6), 881–906. https://doi.org/10.1007/s00778-017-0486-1

Konstantinou, N., Abel, E., Bellomarini, L., Bogatu, A., Civili, C., Irfanie, E., Koehler, M., Mazilu, L., Sallinger, E., Fernandes, A. A. A., Gottlob, G., Keane, J. A., & Paton, N. W. (2019). VADA: An architecture for end user informed data preparation. *Journal of Big Data*, *6*(74), 1-32. https://doi.org/10.1186/s40537-019-0237-9

Konstantinou, N., & Paton, N. W. (2020). Feedback driven improvement of data preparation pipelines. *Information Systems*, *92*, Article 101480. https://doi.org/10.1016/j.is.2019.101480

Liu, L., Hasegawa, S., Sampat, S. K., Xenochristou, M., Chen, W., Kato, T., Kakibuchi, T., & Asai, T. (2024). AutoDW: Automatic data wrangling leveraging large language models. In *ASE '24: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering* (pp. 2041–2052). ACM. https://doi.org/10.1145/3691620.3695267

Narayan, A., Chami, I., Orr, L., & Ré, C. (2022). Can Foundation Models Wrangle Your Data? *Proceedings of the VLDB Endowment*, *16*(4), 738-746. https://doi.org/10.14778/3574245.357425

Paton, N. (2019). *Automating data preparation: Can we? Should we? Must we?* Research Explorer, the University of Manchester. https://research.manchester.ac.uk/en/publications/automating-data-preparation-can-we-should-we-must-we/

Rezig, E. K., Ouzzani, M., Elmagarmid, A. K., Aref, W. G., & Stonebraker, M. (2019). Towards an End-to-End Human-Centric data cleaning framework. In *HILDA '19: Proceedings of the Workshop on Human-In-The-Loop Data Analytics* (pp. 1–7). ACM. https://doi.org/10.1145/3328519.3329133

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). "Everyone wants to do the model work, not the data work": Data cascades in High-Stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1–15). ACM. https://doi.org/10.1145/3411764.3445518

Shome, A., Cruz, L., Spinellis, D., & Arie, V. D. (2024). Understanding feedback mechanisms in Machine Learning JuPyter Notebooks. *arXiv*, 1-37. https://doi.org/10.48550/arxiv.2408.00153

Singh, J., Cobbe, J., & Norval, C. (2019). Decision Provenance: Harnessing Data Flow for Accountable Systems. *IEEE Access*, *7*, 6562-6574. https://doi.org/10.1109/ACCESS.2018.2887201

Somasundaram, P. (2022). Streamlining Data Wrangling Processes through Automation and Tooling. *International Journal of Science and Research (IJSR)*, *11*(2), 1323–1325. https://doi.org/10.21275/sr24418101747

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (pp. 29–39).

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2023). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, *135*, 364–381. https://doi.org/10.1016/j.future.2022.05.014

Yin, Y., Wang, B., Zuo, H., & Childs, P. (2024). Effects of different Human-In-The-Loop approaches on Human-AI Co-Design: A comparison between Human-Learning HITL approach and Machine-Learning HITL Approach. In *Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 2A, Article V02AT02A050). ASME. https://doi.org/10.1115/detc2024-143295

s