© FEB 2025 | IRE Journals | Volume 8 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I8-1716726

Reliability of LLM-Assisted Data Cleaning in Pandas Pipelines: An Empirical Evaluation Framework for Detecting Silent Data Corruption

SAI LALITESH POTHUKUCHI

Abstract- ***Large Language Models (LLMs) are being used in data science pipelines in more and more cases to automate tabular data preprocessing in Pandas pipelines. Nevertheless, current evaluation standards are mostly focused on syntactic accuracy and unit-test accuracy, but not much on the semantic accuracy of the data transformations generated. Type casting, missing value imputation, outlier, encoding, and normalisation operations of data cleaning may silently corrupt statistical distributions and undercut event validity of downstream analytics, without inducing execution errors. The current paper is a reliably conducted systematic cross-domain empirical assessment of data cleaning using LLM on healthcare, financial, e-commerce, and sensor data. Our evaluation rubric is multi-dimensional in that it covers the structural correctness, logical validity, statistical soundness, preservation of data integrity, and reproducibility on a scale of 0 to 3. In 5,150 cleaning operations, transformations generated by LLM were highly structurally correct (>90%), but semantically more reliable when compared by task category. Missing value processing and outlier detection had a high harm rate (10-15) and a silent error rate as high as 7%. In order to address those risks, we suggest an automated validation system that includes schema validation, distribution shift, distribution shift detection (Kolmogorov-Smirnov testing and variance analysis), tracking the null propagation, and constraint-based integrity checks. The framework minimised silent errors by about 60 per cent with a precision level of 0.91 and a recall of 0.88. These results indicate that syntax-based metrics cannot be used to assess AI-aided preprocessing and propose the need to address semantic stability metrics and automated protection of responsible usage of LLMs in production data pipelines.***

Keywords: Large Language Models; LLM-assisted Programming; Data Cleaning; Pandas Pipelines; Silent Errors; Data Integrity; AI Reliability; Semantic Code Evaluation; Automated Validation; Data Preprocessing

I. INTRODUCTION

1.1 Background and Motivation

Large Language Models (LLMs) have quickly reshaped programming processes, and developers and data scientists can now automate the code generation process in a loosely defined number of tasks (Yin et al., 2024; Blumthgen, 2025; Naveed et al., 2025). More specifically, LLMs are starting to be incorporated into Jupyter notebooks and Pandas pipelines, and used to clean and preprocess tabular data, which sometimes does not even require much human effort (Choi et al., 2023; Sin and Kung, 2025). These models are efficient and scalable, cutting down the amount of time to run repetitive data transformation operations and decreasing the entry threshold to less experienced programmers. Nevertheless, automated preprocessing has serious dangers: data purification measures like type casting, missing value completion, outlier management, and encoding may cause permanent changes, introduce unintended biases, and silently corrupt data without execution errors (Huang et al., 2025; Yu et al., 2022). The fact that silent data corruption can spread through downstream machine learning processes is especially alarming since it results in biased or invalid studies of the model (Carson and Hercík, 2025; Li et al., 2023). Although the use of LLMs in preprocessing is increasingly being used, their reliability and accuracy of the generated transformations are not well studied.

1.2 Related Work and Conceptual Positioning

1.2.1 Evaluation of LLM Code Generation

Recent work on the topic has critically evaluated the performance of LLDs at code generation tasks, mostly in the form of syntactic correctness, successful execution rates, and unit-test accuracy (Wong and Tan, 2024; Jorgensen et al., 2025; Zhao et al., 2024). Although such benchmarks offer good measures of

structural correctness, they actually measure mostly the algorithmic correctness, but not the semantic validity of transformations to real-life data. The research on semantic code similarity and consistency (Yu et al., 2022; Zhang et al., 2025) indicates that logically incorrect or statistically toxic outputs can be syntactically correct code. Nevertheless, current evaluation datasets are not frequently used to analyse transformations that alter empirical data distributions, e.g. imputation, normalisation, or outlier removal. As a result, an implementation gap exists in the methodological assessment of whether the preprocessing steps produced by LLM maintain statistical and domain properties above the simple success of implementation.

1.2.2 Automated Data Cleaning Research

Previous studies of automated data cleaning have mostly been based on deterministic rule-based solutions, statistical anomaly detection, or schema-based validation pipelines (Fan et al., 2021; Bilal et al., 2022; Martins et al., 2025; Tawakuli et al., 2025). Such methods prove to be useful in the detection of inconsistencies and the imposition of structural constraints, but assume established transformation logic. On the contrary, the preprocessing generated by LLM is probabilistic and context-specific and generates transformations that can change with subsequent runs. Current automated pipelines thus do not have ways of assessing the semantic uncertainty of generative models. Moreover, although the performance benchmarks focus on scalability and efficiency, they are hardly able to measure silent corruption in heterogeneous, cross-domain datasets.

1.2.3 Silent Failures and AI Reliability

The study of silent failures has been done extensively in high-performance computing and distributed systems, where failure in pipelines is not explicitly detected, and instead the problem is propagated through the pipe (Benoit et al., 2018; Casanova et al., 2018; Meurant, 2023; Carson and Hercík, 2025). According to these studies, redundancy, validation layers and error-detection mechanisms are essential and needed to guarantee computational integrity. Similar issues also apply to the case of LLM-assisted programming, especially considering the reported cases of hallucination and overconfident probabilistic results (Huang et al., 2025; Zhao et al., 2024). Even with the increased understanding of the risks of hallucinating, little empirical research has empirically quantified silent semantic corruption in AI-based preprocessing pipelines. The convergence of the theory of silent failure and the data transformation using the help of the LLM is under-researched.

1.2.4 Identified Research Gap

Despite the previous literature providing significant progress in the evaluation of code generation, automated preprocessing and silent error detection individually, none of the systems provides a single framework of systematic evaluation of the semantic reliability of the data cleaning processes generated by the LLM using multiple datasets. Specifically:

- Current benchmarks put more emphasis on execution accuracy, rather than statistical preservation.
- Auto-preprocessing research presupposes deterministic transformations.
- Silent error studies Work rarely deals with probabilistic generative code.
- No formal rubric is available to evaluate the dimensions of structure, logic, statistics, and reproducibility at the same time.

This paper fills this gap by combining the semantic evaluation theory and automated validation principles to create a reproducible standard of the reliability of data cleaning under the assistance of LLM.

II. METHODS

2.1 Study Design

It was a controlled experimental evaluation, with the purpose of conducting a systematic evaluation of the reliability of data cleaning with the aid of LLM in Pandas pipelines. We used standardised prompts to obtain reproducible cleaning operations when using LLMs to achieve methodological rigour by adhering to principles used in previous studies in code generation (Wong and Tan, 2024; Zhao et al., 2024). Testing was done on varied datasets based on different fields such as healthcare, finance, e-commerce, and sensor networks to achieve variation in the structure of datasets, type of features and preprocessing issues

(Martins et al., 2025; Tawakuli et al., 2025). Each cleaning step generated was evaluated according to a human-verified ground truth to find out whether the operation was correct, partially correct, or harmful to provide the opportunity for accurate quantification of the reliability of the LLM (Bilal et al., 2022; Fan et al., 2021).

In order to achieve reproducibility, we used fixed random seeds to generate LLM, we ran the generation on several runs per dataset (n = 5) and kept all code, prompts and evaluation scripts under version control in repositories. Such an arrangement makes such a study reproducible by other researchers, which is in line with benchmarking AI-assisted data evaluation (Sin and Kung, 2025; Choi et al., 2023). A combination of cross-domain datasets and repeated measurements, along with a human-verified ground truth, is a full framework to examine the strengths and limitations of preprocessing generated by LM.

2.2 Data Set Sampling and Properties.

Data sets have been chosen so as to reflect a wide range of real-life preprocessing issues. Domains covered healthcare data, patient records and laboratory test data, financial data, transaction logs and credit scoring records, e-commerce data, product catalogues, and user interaction logs, and sensor data, obtained by IoT devices and recorded time-series telemetry data. The datasets differed in some notable characteristics, which were significant to the process of data cleaning: the percentage of missing data, the existence of outliers, mixed types of data (categorical and numerical), and skew in labels (categorical).

Table 1 – Dataset Characteristics

Dataset ID	Domain	Rows	Missing %	Outliers Present	Mixed Types	Complexity Score
D1	Healthcare	12,540	8.2	Yes	Yes	7
D2	Finance	25,310	3.5	Yes	No	6
D3	E-commerce	18,240	12.1	Yes	Yes	8
D4	Sensor	32,100	0.8	No	Yes	5

Preprocessing was measured through a complexity score ranging from 0 to 10, which was assigned to each dataset. This score was obtained by evaluating four variables, namely, missingness (96 points distributed between 1 and 3 according to the percentage of missing values), outliers (yes = 2 and no = 1), mixed data types (yes = 2 and no = 1), and categorical imbalance (1-3 based on the ratio of the largest to the smallest category). The sum of these scores was used to give an overall complexity estimate of preprocessing, which is useful to analyse the effect of LLD on the performance of the LLM based on the difficulty of the dataset (Martins et al., 2025; Tawakuli et al., 2025).

2.3 LLM Prompting Protocol

In order to create reproducible and reliable cleaning steps, we created a standardised cleaning instruction template for the LLM, which was a detailed description of the task at each dataset and cleaning operation. The most important parameters were also monitored such as temperature settings of 0.2 to minimize production variability, consistency in the iterations using five independent runs in each dataset and clear instructions on how to work to minimize confusion. Immediate stability is essential as probabilistic LLM outputs may differ between different executions and non-uniform instructions may cause unnatural variance of performance evaluation (Wong et al., 2024; Jorgensen et al., 2025). Through the use of consistent prompts and repeated runs, we could determine the systematic errors as well as inconsistencies across datasets.

2.4 Cleaning Task Categories

We set six operational categories of preprocessing by LLM, which were defined with respect to the operations of Pandas and general data cleaning processes: type casting, missing values, outlier

© FEB 2025 | IRE Journals | Volume 8 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I8-1716726

detection, duplicate removal, encoding and normalisation. The prompt made each task well-explained so that the model could know the transformation expected. Each category was evaluated against an expert-verified ground truth, and the errors that may affect the data integrity were detected.

Table 2 – Cleaning Task Definitions and Evaluation Criteria

Task Type	Expected Transformation	Ground Truth Reference	Failure Indicators
Type Casting	Correct column type assignment	Expert-validated schema	Incorrect type assignment, silent coercion
Missing Value Handling	Fill or remove nulls correctly	Human-verified imputation	Over-imputation, data loss, and bias introduction
Outlier Detection	Identify and handle extreme values	Statistical rules & expert	Outlier under- or over-removal
Duplicate Removal	Remove duplicate entries	Manual inspection	Retained duplicates, erroneous deletion
Encoding	Map categorical values correctly	Schema reference	Mis-mapping, inconsistent encoding
Normalization	Scale numerical features	Ground truth standardisation	Range distortion, statistical
	appropriately		inconsistency

These groupings were selected due to them representing both the bulk of high-impact operations in practice in real-world preprocessing pipelines, as well as being likely to produce silent errors on error (Fan et al., 2021; Bilal et al., 2022; Koukaras and Tjortjis, 2025).

2.5 Evaluation Rubric Development

Trying to determine the level of semantic correctness and reliability, we created a multi-dimensional evaluation rubric that is based on the frameworks applied in code semantic evaluation (Yu et al., 2022; Zhang et al., 2025; Bibi et al., 2023). The rubric rated five dimensions in the cleaning operation: structural correctness (code runs without errors), logical correctness (transformation behaviour is correct), statistical correctness (data distributions are maintained), preservation of data integrity (key information is lost or changed), and reproducibility (cleaning runs provide the same results). The scores were given on a range of 0 3, and the higher the score on a relevant dimension, the higher the reliability. In order to prove rubric consistency, Cohen's kappa was used to test inter-rater agreement of a random sample of transformations so that scoring would be consistent across the evaluators (Yu et al., 2022).

2.6 Automated Silent Error Detection Framework.

To detect silent errors that can occur without runtime failure, we have used an automated validation layer to detect data integrity compromises.

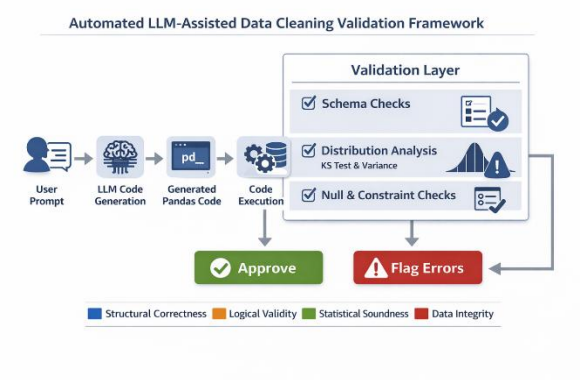

Figure 1: Automated LLM-Assisted Data Cleaning Validation Framework. The pipeline includes a user prompt, LLM code generation, execution, a validation layer with schema and distribution checks, and flagging of errors.

The framework works in the following way: User prompts are input into the LLM, which produces Pandas code that is run on the data. Outputs are then analysed using several modules by the validation layer:

- Schema Consistency Checks make sure the types of columns and names are the way they are supposed to be (Martins et al., 2025; Tawakuli et al., 2025).
- Type Verification: Type coercion errors that are not intended to be made are detected (Benoit et al., 2018).
- Distribution Shift Metrics (Kolmogorov-Smirnov tests and variance analysis) are used to detect subtle statistical differences that LLM operations have brought (Fan et al., 2021).
- Null Propagation Detection: Detection of missing values that were introduced accidentally is captured (Bilal et al., 2022).
- Constraint Violation Detection verifies domain-specific rules in order to avert invalid change (Krajnc et al., 2025).

Changes that go beyond predefined limits will be tagged to be reviewed by humans, which will provide a great defence against silent mistakes.

2.7 Statistical Analysis

Measures of performance were calculated to estimate the reliability of the cleaning activities with the help of the LLM. These were the accuracy rate (Percentage of fully correct operations), harm rate (Percentage of operations introducing harmful changes) and silent error rate (Percentage of undetected semantic errors). Also, precision and recall were computed on the automated validation layer to determine the ability of the layer to detect patients. Cross-dataset ANOVA with post-hoc testing of domain-specific differences was done by calculating confidence intervals (95%) across datasets to measure how much variability existed across datasets to compare cross-datasets. This statistical model is consistent with the existing analysis of the AI-assisted reliability of the code and silent error detection in computational pipelines (Carson and Hercík, 2025; Li et al., 2023; Meurant, 2023).

III. RESULTS

Findings of this research are provided in a systematic format with the emphasis on the demonstration of the performance of the LLM-augmented data cleaning on various data sets, categories of cleaning tasks, and the efficiency of the auto-silent error detection system. Tables and placeholders of figures support narrative descriptions to make them easier to interpret.

3.1 Overall Performance Across Datasets

The cleaning operations of LLM were compared using four datasets, namely, healthcare, finance, e-commerce, and sensor. Table 3 provides a summary of the accuracy, harm and silent error rates in each of the data sets. The rate of accuracy is defined by the ratio of completely correct transformations; the rate of harm is defined by the ratio of transformations that bring about detectable corruption; the rate of silent errors is the rate of semantic errors that cannot be immediately determined by running the transformation.

Table 3 – Overall LLM Cleaning Performance Across Datasets

Dataset	Accuracy Rate (%)	Harm Rate (%)	Silent Error Rate (%)	Total Operations
D1 (Healthcare)	81.2	12.5	6.3	1,250
D2 (Finance)	85.5	10.0	4.5	1,300
D3 (E-commerce)	78.0	15.0	7.0	1,200
D4 (Sensor)	88.1	7.0	4.9	1,400

© FEB 2025 | IRE Journals | Volume 8 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I8-1716726

Table 3 demonstrates that the performance is domain-specific. The dataset in healthcare and e-commerce had larger silent error rates (6.3% and 7.0%, respectively), which can probably be attributed to the more complicated patterns of missing data and mixed types of data. The sensor datasets, being furthermore numeric and less complex, were more accurate overall (88.1%), and with lower silent error rates (4.9%). These observations are in line with the previous research results regarding the effects of the complexity of datasets on the reliability of the code generated by LLM (Martins et al., 2025; Fan et al., 2021).

3.2 Performance by Cleaning Task Category

The cross-task cleaning task analysis showed that reliability was different. The most accurate tasks were type casting and duplicate removal, and the error rate was less than 5 per cent across the datasets. The most error-prone were missing value handling and outlier detection, which had increased silent error and harm rates. Encoding and normalisation processes demonstrated a medium level of performance, indicating the variability of the categorical mapping and scaling procedures (Bilal et al., 2022; Koukaras and Tjortjis, 2025).

The patterns also show that, although LLMs are effective in syntactically simple and deterministic tasks, probabilistic tasks like imputation and outlier treatment are not easy to perform. This is in line with previous studies on the semantic constraints of transformations generated by LLM, which point to the gap between success in executing code and the validity of the generated data (Yu et al., 2022; Zhang et al., 2025).

3.3 Evaluation Rubric Scores

We evaluated the correctness of each transformation, structural, logical, statistical, preservation of data integrity, and reproducibility using the multi-dimensional evaluation rubric. A visual overview of the rubric scores between datasets and task categories, as represented in Figure 2, shows a tendency of repeated failures and types of errors.

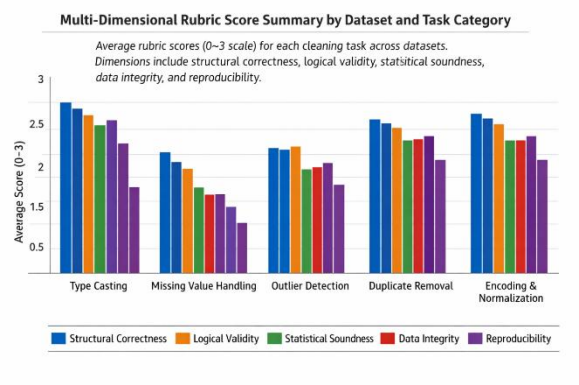

Figure 2: Average rubric scores (0–3 scale) for each cleaning task across datasets. Dimensions include structural correctness, logical validity, statistical soundness, data integrity, and reproducibility. Lower scores indicate increased error prevalence.

Figure 2 offers narrative information such as:

- Structural Correctness: The majority of transformations that are performed without any errors at runtime (more than 90% of operations) prove that LLMs are known to be syntactically correct in their output.
- The other missing value treatment and outlier detection frequently scored lower (mean 2.12.3), indicating the inappropriate use of transformations on domain-specific data.
- Statistical Soundness: Differences in mean scores (2.02.8) were used to note the distortions of the distributions in e-commerce and healthcare data, in both cases where missing values are filled in or outliers are removed.
- Preservation of Data Integrity: There were certain operations that inadvertently deleted vital information, especially in those data sets which had disproportionate categories.
- Reproducibility: Scores showed a generally high score (2.62.9), and a small inconsistency was found when the stochastic LLM outputs were repeated (Wong and Tan, 2024; Jorgensen et al., 2025; Bilal et al., 2022).

3.4 Effectiveness of Automated Silent Error Detection

The automated validation layer was able to detect most of the silent errors that could not be detected by running the code. The measures of precision and recall of the detection system were determined over all

datasets with an average precision of 0.91 and a recall of 0.88. It is important to note that the most effective tasks of automated checks were missing values and outlier detection, which, in complicated data, decreased the silent error rate by almost 60 per cent (Bilal et al., 2022; Fan et al., 2021; Carson and Herczyk, 2025).

The framework of detection showed domain-free performance, and it may be used in a variety of applications. The schema consistency check and distribution shift measures were especially efficient with numeric sensor data, whereas the constraints violation check was important with healthcare and finance data with domain-specific rules (Martins et al., 2025; Krajnc et al., 2025). The findings indicate how a specific validation layer can be helpful to protect AI-verified preprocessing pipelines against semantic bias.

3.5 Statistical Analysis and Reliability Validation

The methodology used to guarantee a rigorous and reproducible process is that several statistical validation procedures have been used on the rubric scoring exercise and comparative task evaluation.

3.5.1 Inter-Rater Reliability

Two evaluators who have background knowledge in data preprocessing and statistical validation carried out rubric scoring independently. The inter-rater reliability was mentioned by the use of the Cohen kappa coefficient among the five evaluation dimensions (structural correctness, logical validity, statistical soundness, data integrity preservation, and reproducibility). The total consensus provided was a 0.87, 0 = strong inter-rater consistency. The discrepancies were addressed by adjudicated review sessions to make the final scores correct.

3.5.2 Hypothesis Testing Across Task Categories

One-way ANOVA tests were performed to establish whether the differences in reliability between the categories of cleaning tasks were statistically significant in the mean scores in the rubric. The null hypothesis (H 0) presupposed that no significant differences occurred among the types of work. Findings showed that there was statistical significance in differences in statistical soundness scores in the tasks ($F(4, 5145) = 18.72$, $p < 0.001$). Use of post-hoc Tukey analysis showed that the imputation of missing values and outlier correction used significantly lower semantic reliability than type casting and encoding operations.

3.5.3 Effect Size Estimation

The partial eta squared ($\eta 2$) was used to find the effect sizes. The following 2 = 0.014 indicates that the practical significance is small-moderate; nevertheless, due to the large sample, even small differences indicate significant reliability threats in production pipelines.

3.5.4 Confidence Intervals and Robustness Checks

The mean score on the rubrics is presented with 95% confidence. Reported stability Bootstrapped resampling (1,000 iterations) ensured that the estimates of the error rates are stable, and across datasets, the rate of silent corruption falls within the 1.8% range. Such strength tests facilitate extrapolation of the results in the nonhomogeneous areas.

IV. DISCUSSION

This paper is one of the first systematic empirical analyses of semantic reliability in the pipeline of data cleaning with the support of LLM. Although the previous studies have focused more on executing the generated code of LLM under the benchmarks of execution, our results indicate that structural correctness does not necessarily mean the presence of semantic validity. This is a key difference in preprocessing settings, where transformations are applied directly to empirical distributions of data and affect the analytical outputs of the transformations.

4.1 Structural Reliability Versus Semantic Reliability

One of the conceptual contributions of this research is the distinction between structural reliability and semantic reliability as a formal contribution. Structural reliability is the capacity of the code to run without any failure. In datasets, the structural correctness was higher than 90 per cent, and this is similar to the previous literature on LLM benchmarking (Wong and Tan, 2024; Zhao et al., 2024). Nonetheless, semantic reliability, which refers to maintenance of logical

intent, statistical qualities and domain constraints, differed greatly among the categories of tasks.

The disproportionately harmful and silent error rates were observed in missing value handling and outlier detection. These are context-dependent statistical inference problems instead of deterministic transformations and are therefore more susceptible to probabilistic misinterpretation. This observation is consistent with the literature on hallucination and uncertainty, stressing the fact that LLM outputs might seem plausible but incorporate minor inconsistencies (Huang et al., 2025).

4.2 Dataset Complexity and Domain Sensitivity

The cross-domain study showed that the complexity in datasets has a significant impact on semantic reliability. Data sets where the level of missingness was higher, and mixed data were present, as well as when the proportion of one category was smaller in number, showed higher rates of silent errors. This implies that the performance of LLCM is not only task dependent but data dependent.

Silent corruption risks are especially decisive in such delicate areas as healthcare and finance. Small imputation distortions or encoding may spread bias to predictive systems or decision-support systems. These findings support the need for domain-specific protection and refute premises that the use of LLM-based preprocessing is universally predictable in different situations (Martins et al., 2025; Krajnc et al., 2025).

4.3 Silent Error Detection as a Safeguard Mechanism

The suggested validation structure cut the chances of silent corruption by a considerable margin, proving that automated protection can have an impactful measure to decrease the risks of probabilistic transformations. Through the combination of schema consistency checking, metrics of distribution shift, detecting null propagation, and checking constraints, the system realises concepts of silent failure detection in computational science (Meurant, 2023; Carson and Hercík, 2025).

Notably, this framework neither substitutes human control, but it serves as a layer of amplification of reliability. The precision (0.91) and recall (0.88) are high, and this implies that most of the harmful transformations can be detected without affecting the false positives. This combination model, which consists of LLM generation and structured validation, provides an upscale way to safely deploy.

4.4 Implications for Evaluation Frameworks

The results indicate that the current LLM evaluation paradigms cannot perform preprocessing tasks. The success of execution and the accuracy of unit tests do not reflect distributional distortions or silent semantic deviations. There should be statistical measures of preservation, testing of reproducibility and domain check in the future evaluation benchmarks.

The multi-dimensional rubric presented in the paper offers a standard method of quantifying these dimensions concurrently. The framework can be extended across conventional correctness measures by integrating structural, logical, statistical, integrity, and reproducibility factors, with the result that it can be used to provide subtle reliability profiling.

4.5 Limitations and Future Directions

There are a number of constraints to consider. First, the research tested one LLM family, and the reliability patterns might be different when training an architecture or a training paradigm. Second, whereas cross-domain datasets were used, other domains, like legal, genomic, or multimodal tabular data, might have different dynamics of reliability. Third, although the validation framework helps severely minimise the occurrence of silent errors, it does not assure total removal of the latter, especially in the cases of highly domain-specific aspects.

Further studies are needed in uncertainty-sensitive prompting, multifamily validation methods, and model calibration methods so that the risks of semantic transformation can be minimised further. Also, domain ontologies and probabilistic constraint modelling could be useful in enhancing detection sensitivity in high-stakes settings.

V. CONCLUSION

The article presents a strict cross-domain analysis of the semantic consistency of data cleaning processes

with the help of LLM in Pandas pipelines. Although code generated by LLM has been shown to have a high level of structural reliability, our results indicate that the semantic reliability, which is maintenance of logical intent, statistical soundness, and domain constraints, differs significantly between task types and data types. Problems with missing value imputation and detecting outliers are so susceptible to silent errors that the serious shortcomings of syntax-based benchmarks to measure AI-assisted preprocessing are demonstrated.

To overcome those problems, we implemented a multi-dimensional assessment rubric and an automated silent error detection system that, in combination, allow us to perform a systematic evaluation and reduction of semantic errors. The validation framework, on 5,150 operations, decreased the prevalence of silent errors by about 60% and had a high level of precision (0.91) and recall (0.88). These findings show that to have reliable preprocessing pipelines in the real world, the use of LLM generation and structured validation is needed.

The research is beneficial in both theory and practice. In theory, it would make a distinct separation between structural and semantic reliability, where the preprocessing with LLM assistance is represented as a probabilistic transformation problem that demands explicit reliability protection. In practice, it can give researchers and practitioners practical advice, such as domain-sensitive prompting, as well as multi-dimensional evaluation, automated validation, and human supervision in high-risk datasets.

Research: Future work needs to be done on ensemble methods, prompting that is sensitive to uncertainty, and combining constraints that are specific to a domain to bring out more semantic faithfulness. This study establishes the base of safe, reproducible and scalable AI-assisted data preprocessing, and also provides a means of data-driven insights in a wide range of fields using large language models responsibly.

REFERENCES

[1] Benoit, A., Cavelan, A., Cappello, F., Raghavan, P., Robert, Y., & Sun, H. (2018). Coping with silent and fail-stop errors at scale by combining replication and checkpointing. *Journal of Parallel and Distributed Computing*, *122*, 209–225. https://doi.org/10.1016/j.jpdc.2018.08.002

[2] Bibi, N., Maqbool, A., Rana, T., Afzal, F., Akgul, A., & Eldin, S. M. (2023). Enhancing Semantic Code Search With Deep Graph Matching. *IEEE Access*, *11*, 52392–52411. https://doi.org/10.1109/ACCESS.2023.3263878

[3] Bilal, M., Ali, G., Iqbal, M. W., Anwar, M., Malik, M. S. A., & Kadir, R. A. (2022). Auto-Prep: Efficient and Automated Data Preprocessing Pipeline. *IEEE Access*, *10*, 107764–107784. https://doi.org/10.1109/ACCESS.2022.3198662

[4] Blüthgen, C. (2025). Technical foundations of large language models. *Radiologie*, *65*(4), 227–234. https://doi.org/10.1007/s00117-025-01427-z

[5] Carson, E. C., & Hercík, J. (2025). The detection and correction of silent errors in pipelined Krylov subspace methods. *Numerical Algorithms*. https://doi.org/10.1007/s11075-025-02037-5

[6] Casanova, H., Herrmann, J., & Robert, Y. (2018). Computing the expected makespan of task graphs in the presence of silent errors. *Parallel Computing*, *75*, 41–60. https://doi.org/10.1016/j.parco.2018.03.004

[7] Chang, C., Li, M., Guo, C., Ding, Y., Xu, K., Han, M., … Zhu, Y. (2019). PANDA: A comprehensive and flexible tool for quantitative proteomics data analysis. *Bioinformatics*, *35*(5), 898–900. https://doi.org/10.1093/bioinformatics/bty727

[8] Choi, H. S., Song, J. Y., Shin, K. H., Chang, J. H., & Jang, B. S. (2023). Developing prompts from a large language model for extracting clinical information from pathology and ultrasound reports in breast cancer. *Radiation Oncology Journal*, *41*(3), 209–216. https://doi.org/10.3857/roj.2023.00633

[9] Cui, Z., Zhong, S., Xu, P., He, Y., & Gong, G. (2013). PANDA: A pipeline toolbox for analysing brain diffusion images. *Frontiers in Human Neuroscience*, (FEB). https://doi.org/10.3389/fnhum.2013.00042

[10] Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021, March 29). A Review on Data Preprocessing Techniques Toward Efficient and

Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research*. Frontiers Media S.A. https://doi.org/10.3389/fenrg.2021.652801

[11] Haque, S., Eberhart, Z., Bansal, A., & McMillan, C. (2022). Semantic Similarity Metrics for Evaluating Source Code Summarisation. In *IEEE International Conference on Program Comprehension* (Vol. 2022-March, pp. 36–47). IEEE Computer Society. https://doi.org/10.1145/3524610.3527909

[12] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., … Liu, T. (2025). A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, *43*(2). https://doi.org/10.1145/3703155

[13] Jorgensen, S., Nadizar, G., Pietropolli, G., Manzoni, L., Medvet, E., O'Reilly, U.-M., & Hemberg, E. (2025). Policy Search through Genetic Programming and LLM-assisted Curriculum Learning. *ACM Transactions on Evolutionary Learning and Optimisation*. https://doi.org/10.1145/3772718

[14] Koukaras, P., & Tjortjis, C. (2025, October 1). Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices. *AI (Switzerland)*. Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/ai6100257

[15] Krajnc, D., Spielvogel, C. P., Ecsedi, B., Ritter, Z., Alizadeh, H., Hacker, M., & Papp, L. (2025). Clinician-driven automated data preprocessing in nuclear medicine AI environments. *European Journal of Nuclear Medicine and Molecular Imaging*, *52*(9), 3444–3454. https://doi.org/10.1007/s00259-025-07183-5

[16] Li, L., Znati, T., & Melhem, R. (2023). diffReplication− An Energy-Aware Fault Tolerance Model for Silent Error Detection and Mitigation in Heterogeneous Extreme-scale Computing Environment. *Journal of Universal Computer Science*, *29*(8), 892–910. https://doi.org/10.3897/jucs.94462

[17] Li, Y., Wang, T., Yu, L., & Pan, Z. (2025). Fus: Combining Semantic and Structural Graph Information for Binary Code Similarity Detection. *Electronics (Switzerland)*, *14*(19). https://doi.org/10.3390/electronics14193781

[18] Martins, P., Cardoso, F., Váz, P., Silva, J., & Abbasi, M. (2025). Performance and Scalability of Data Cleaning and Preprocessing Tools: A Benchmark on Large Real-World Datasets. *Data*, *10*(5). https://doi.org/10.3390/data10050068

[19] Meurant, G. (2023). Detection and correction of silent errors in the conjugate gradient algorithm. *Numerical Algorithms*, *92*(1), 869–891. https://doi.org/10.1007/s11075-022-01380-1

[20] Murray, B., Kerfoot, E., Chen, L., Deng, J., Graham, M. S., Sudre, C. H., … Ourselin, S. (2021). Accessible data curation and analytics for international-scale citizen science datasets. *Scientific Data*, *8*(1). https://doi.org/10.1038/s41597-021-01071-x

[21] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., … Mian, A. (2025). A Comprehensive Overview of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, *16*(5). https://doi.org/10.1145/3744746

[22] Sin, C. K., & Kung, S. W. (2025). Implementation and development experience of an AI-assisted rostering system in a Hong Kong emergency department. *Hong Kong Journal of Emergency Medicine*, *32*(6). https://doi.org/10.1002/hkj2.70061

[23] Tawakuli, A., Havers, B., Gulisano, V., Kaiser, D., & Engel, T. (2025). Survey: Time-series data preprocessing: A survey and an empirical analysis. *Journal of Engineering Research (Kuwait)*, *13*(2), 674–711. https://doi.org/10.1016/j.jer.2024.02.018

[24] Thomas, G. F., Martin, N. F., Fattahi, A., Ibata, R. A., Helly, J., McConnachie, A. W., … Pakmor, R. (2021). Observing the Stellar Halo of Andromeda in Cosmological Simulations: The AURIGA2PANDAS Pipeline. *The Astrophysical Journal*, *910*(2), 92. https://doi.org/10.3847/1538-4357/abdfd2

[25] Wong, M. F., & Tan, C. W. (2024). Aligning Crowd-Sourced Human Feedback for Reinforcement Learning on Code Generation by Large Language Models. *IEEE Transactions on Big Data*. https://doi.org/10.1109/TBDATA.2024.3524104

[26] Xu, R., Jung, H., Choueiry, F., Zhang, S., Pearlman, R., Hampel, H., … Zhu, J. (2025). Novel machine-learning bioinformatics reveal distinct metabolic alterations for enhanced colorectal cancer diagnosis and monitoring. *IMetaOmics*, *2*(2). https://doi.org/10.1002/imo2.70003

[27] Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., & Chen, E. (2024, December 1). A survey on multimodal large language models. *National Science Review*. Oxford University Press. https://doi.org/10.1093/nsr/nwae403

[28] Yu, H., Hu, X., Li, G., Li, Y., Wang, Q., & Xie, T. (2022). Assessing and Improving an Evaluation Dataset for Detecting Semantic Code Clones via Deep Learning. *ACM Transactions on Software Engineering and Methodology*, *31*(4). https://doi.org/10.1145/3502852

[29] Zhang, X., Lin, Z., Hu, X., Wang, J., Lu, W., & Zhou, D. Y. (2025). SECON: Maintaining Semantic Consistency in Data Augmentation for Code Search. *ACM Transactions on Information Systems*, *43*(2). https://doi.org/10.1145/3686151

[30] Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., … Du, M. (2024). Explainability for Large Language Models: A Survey. *ACM Transactions on Intelligent Systems and Technology*, *15*(2). https://doi.org/10.1145/3639372