DemandClean: A Multi-Objective Learning Framework for Balancing Model Tolerance to Data Authenticity and Diversity

Zekai Qian
qzk010728@gmail.com
Harbin Institute of Technology, China

Xiaoou Ding
dingxiaoou@hit.edu.cn
Harbin Institute of Technology, China

Chen Wang
Tsinghua University, China
wang_chen@tsinghua.edu.cn

Hongzhi Wang*
wangzh@hit.edu.cn
Harbin Institute of Technology, China

ABSTRACT

Real-world datasets often suffer from multiple quality issues, hindering downstream model performance and increasing cleaning costs. To address this, we propose DemandClean, a reinforcement learning-based adaptive data cleaning framework that dynamically balances cleaning effectiveness and operational costs. DemandClean explicitly considers data authenticity (alignment with real-world facts), diversity (richness of feature values), and downstream models' noise tolerance. We categorize data errors as missing (reducing authenticity and diversity), semantic (affecting only authenticity), and syntactic (affecting authenticity but potentially increasing diversity). Based on these errors, DemandClean intelligently selects among Repair, Delete, or No actions, guided by error rates and model robustness. For interpretability, the framework visually distinguishes authenticity, diversity, and tolerance. Extensive experiments confirm that DemandClean achieves near-optimal accuracy at substantially reduced preprocessing costs. Specifically, it reduces repair actions by 80.0% and deletions by 80.7% compared to "Repair All" strategies, while maintaining or even exceeding their predictive performance, thus offering an interpretable, cost-effective, and scalable solution for practical applications.

PVLDB Reference Format:
Zekai Qian, Xiaoou Ding, Chen Wang, and Hongzhi Wang. DemandClean: A Multi-Objective Learning Framework for Balancing Model Tolerance to Data Authenticity and Diversity. PVLDB, 18(12): 5339 - 5342, 2025.
doi:10.14778/3750601.3750666

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/qzkinhit/DemandClean.

1 INTRODUCTION

High-quality data is typically characterized by two essential properties: **authenticity**, referring to the correctness of data values, and **diversity**, reflecting the richness of distinct values across features [8]. Unfortunately, real-world datasets often fall short on both dimensions, and ensuring their quality remains costly [1]. Traditional rule-based methods rely on explicit data quality constraints [2, 3] for error detection and repair, whereas model-based approaches like HoloClean [7] and BoostClean [4] leverage machine learning to automate cleaning processes, sometimes directly optimizing downstream performance.

Despite achieving reasonable accuracy and recall (around 0.6–0.8) for error detection tasks across multiple datasets [6], existing approaches often struggle to produce comprehensive and stable repairs without extensive domain knowledge and substantial human involvement [6]. Moreover, general-purpose cleaning methods may rectify localized errors but cannot always ensure improved downstream model performance [5]. Addressing these limitations, recent frameworks like ActiveClean and GoodCore [1] have emerged, focusing on targeted low-cost repairs by selectively cleaning subsets of data to enhance downstream predictive accuracy.

Building upon these insights, we further explore the intricate relationship between data quality and model performance by categorizing errors into three distinct types: missing errors (reducing both authenticity and diversity), semantic errors (impacting authenticity without affecting diversity), and syntactic errors (compromising authenticity yet inadvertently enriching diversity) [6]. Recognizing the diverse sensitivity of downstream models to these errors is crucial. For example, Random Forest models exhibit robustness against noise but are sensitive to missing data, whereas linear models are generally stable under mild noise but degrade sharply with severe anomalies. Consequently, indiscriminate comprehensive repairs may incur high cleaning costs while yielding marginal improvements when errors are sparse or model tolerance is high. Conversely, aggressive cleaning actions become essential under severe contamination or with sensitive models. Uniform repairs fail to effectively balance costs and predictive gains due to varying model sensitivities. Therefore, incorporating downstream model requirements into data cleaning decisions is essential for achieving a cost-effective balance.

Motivated by these challenges, we propose DemandClean, an adaptive reinforcement learning-based data cleaning framework, which addresses the following key aspects:

(1) **Quantifiable interpretability of data quality and model tolerance**: We define a comprehensive five-dimensional state vector $S_{ij} = [E_{ij}, F_j, R_j, i/N, j/M]$ for each data cell, capturing error type (E_{ij}), feature importance (F_j), column-wise error rate (R_j), and normalized positional context ($i/N, j/M$). This facilitates intuitive understanding of the interactions between error types, feature importance, and cleaning decisions. (2) **Adaptive data cleaning**

*Corresponding author.
This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750666

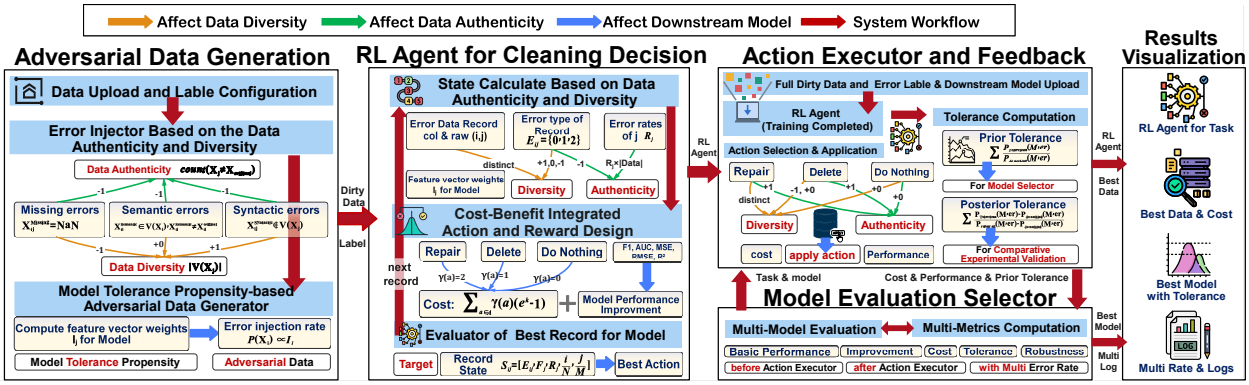

Figure 1: DemandClean system overview.

strategies: Leveraging an integrated multi-objective optimization framework, we dynamically integrate error types (missing, semantic, syntactic) with targeted cleaning actions (*Repair*, *Delete*, *No Action*). The reinforcement learning agent continuously assesses cleaning benefits (improved model performance, restored authenticity) versus associated costs (repair efforts, diversity loss), enabling automatic learning of optimal action sequences tailored to specific error distributions and model sensitivities. (3) **Model-tolerant optimization based on data-task dynamics**: To accommodate broader business scenarios, DemandClean evaluates diverse downstream tasks across at least eight candidate models. It conducts iterative real-time evaluations considering multiple dimensions, including predictive performance, robustness, cleaning costs, performance improvements, and model tolerance. Depending on error severity and model sensitivity, DemandClean adaptively selects optimal combinations of models and cleaning actions, favoring selective repairs or deletions when necessary, and predominantly adopting "*Do Nothing*" strategy when models exhibit higher tolerance, thus preserving data diversity and minimizing redundant cleaning costs.

Overall, compared to "*Repair All*" and "*Delete All*" baseline strategies, DemandClean strategically reduces repair actions by 80.0% and deletions by 80.7%, while achieving comparable or even superior predictive performance. By intelligently aligning data cleaning with downstream model demands, DemandClean offers a highly interpretable, cost-effective, and scalable framework suitable for practical real-world applications.

2 SYSTEM OVERVIEW

Figure 1 presents the architecture of DemandClean with four key modules that balance data authenticity, diversity, and downstream model tolerance to achieve reliable data cleaning solutions.

• *Adversarial Data Generation*: To emulate real-world data quality issues, this module injects three types of errors, i.e., *Missing*, *Semantic*, and *Syntactic*, categorized by their impact on data authenticity and diversity. Configurable parameters such as error rates and injection patterns allow the system to construct diverse and controllable adversarial environments for robust policy learning.

• *RL Agent for Cleaning Decision*: DemandClean formalizes the cleaning process as a reinforcement learning task. Each data cell is encoded as a state vector $S_{ij} = [E_{ij}, F_j, R_j, i/N, j/M]$, capturing its error type, feature importance, error rate, and normalized position. The agent selects actions from {*Repair*, *Delete*, *No Action*}, with rewards based on downstream performance gains minus action costs. A deep Q-learning framework guides the agent to learn a cost-effective cleaning policy through iterative optimization.

• *Action Executor and Feedback*: This module executes the predicted actions, i.e., repair original values (*Repair*), dropping rows (*Delete*), or preserving current data (*No Action*). It updates the data state and returns feedback to the RL agent, facilitating convergence through interaction. Action logs and cost breakdowns are recorded and visualized to assist users in understanding strategy evolution and decision rationale.

• *Model Evaluation Selector*: To quantify model robustness under noisy data, this module evaluates multiple downstream models (e.g., Random Forest, SVM, etc.) using task-specific metrics (Accuracy, F1, MSE, etc.). It also monitors entropy and divergence-based indicators to inform the reward function. By tracking model responses across error types and action frequencies, the system guides users toward robust model-strategy configurations.

In summary, DemandClean orchestrates adversarial data simulation, learning-based action selection, cost-sensitive execution, and multi-model evaluation in a unified pipeline. It enables fine-grained, interpretable, and scalable cleaning decisions tailored to model-specific tolerance and performance, offering a practical solution for real-world data preparation.

3 IMPLEMENTATION DETAILS

Given a large-scale dataset often plagued by diverse quality issues, DemandClean improves data quality by addressing two critical properties: ***authenticity*, measured by the number of correct data cells, and *diversity*, quantified by the number of distinct values per feature column.** To this end, it integrates adversarial data generation, RL-based decision-making, cleaning execution, and model tolerance evaluation into a unified pipeline (Figure 1). We introduce the core technologies in DemandClean below.

3.1 Evaluating Model Tolerance to Data

Note again acquiring fully error-free, high-quality data is often prohibitively expensive in real scenarios. As a practical compromise, existing systems typically rely on expert verification or core-set selection to extract a reliable subset from large-scale datasets for downstream cleaning. DemandClean adopts a randomized sampling strategy, or integrates core-set-based methods [1], to obtain a small subset with a distribution aligned to the full dataset, which serves as the initialization basis for error injection and model tolerance evaluation.

(1) Error injection aligned with authenticity and diversity. We categorize three types of errors based on their impact on authenticity and diversity: **Missing errors**: Null values degrade both authenticity and diversity. These disproportionately affect models relying on complete input (e.g., linear regression), while tree-based models may tolerate moderate missingness. **Semantic errors**: Values are randomly replaced within existing domains, impairing authenticity but preserving diversity. These often shift decision boundaries and require model-specific handling. **Syntactic errors**: Introduced by adding noise or unseen categories, these errors reduce authenticity but may increase diversity. Moderate syntactic noise can improve generalization; excessive noise harms learning.

(2) Directed perturbation via feature importance. Rather than applying uniform corruption, we propose a feature-aware perturbation mechanism. Feature importance is computed based on impurity reduction, i.e., a feature's contribution to minimizing classification or regression errors, using a Random Forest model. Features with higher importance scores are assigned greater corruption probabilities, encouraging the RL agent to better learn the relationship between data manipulations and downstream task performance.

(3) Cost-benefit cleaning action decision model: We formalize the RL framework into states, actions, and rewards, enabling adaptive exploration and convergence:

(a) State definition (State): For each data cell (i, j), we define the state vector $S_{ij} = [E_{ij}, F_j, R_j, i/N, j/M]$, where E_{ij} denotes the error type classified by diversity, F_j represents feature importance computed previously, R_j indicates the column-wise error rate, reflecting authenticity, and i/N, j/M are normalized positional indices capturing data relational context and diversity.

(b) Action and reward function: Actions A are discretized as $\{0, 1, 2\}$, corresponding to *No Action*, *Repair*, and *Delete*, respectively. The reward function integrates downstream model performance gains minus cleaning costs. In the part where performance differences drive the positive rewards, the cumulative action cost (with an upper-bound normalization and exponential penalty) restricts excessive cleaning, where the cleaning cost associated with the action set A is defined as:

$$\text{Cost}(A) = \frac{\Delta P_{max}}{\text{count}_{max}} \sum_{a' \in A} \gamma(a') \left(e^{\frac{\ln(\text{count}_{max}+1)}{\text{count}_{max}} \cdot \text{count}(a')} - 1 \right).$$

(c) Action decision for optimal data diversity and authenticity: Ultimately, `DemandClean` learns a cost-aware decision policy: for each cell at position $(i/N, j/M)$ with error type E_{ij}, the agent selects the high-cost *Repair* action if both the column error rate R_j and feature importance F_j exceed thresholds; otherwise, it chooses between *Delete* and *No Action* to balance performance improvement and cleaning cost.

3.2 Cleaning Action and Multi-Model Selection

To accurately assess the impact of preprocessing on downstream tasks and comprehensively analyze model tolerance, `DemandClean` jointly optimizes both data and model aspects. It includes two key modules: *Data Cleaning* and *Model Selection*.

(1) Cleaning action based on model requirements. In practical applications, users upload complete dirty datasets with error labels identified by an error detector. As detailed in Section 3.1, `DemandClean` computes state vectors for each data cell (i, j), and then predicts the optimal action via the trained RL agent: $a_{ij} = \arg\max_{a \in \{0,1,2\}} Q(S_{ij}, a)$, where the action space $\{0, 1, 2\}$ corresponds to {*No Action*, *Repair*, *Delete*}. Depending on the predicted action, the system updates the data accordingly: for **Repair**, authenticity and diversity is restored by setting $df_{ij} \leftarrow original\_df_{ij}$ (with Missing errors increasing diversity, Semantic errors maintaining diversity, and Syntactic errors decreasing diversity); for **Delete**, the corresponding data row is physically removed ($df \leftarrow df.drop(i)$), reducing dataset size without increasing diversity; and for **No Action**, the polluted state is retained, leaving both diversity and authenticity unchanged, with model tolerance as the only safeguard. The updated data state is returned to the RL agent for the next cycle of evaluation and decision-making.

(2) Model selection considering tolerance. To assess model tolerance to low-quality data, `DemandClean` evaluates multiple candidate models (e.g., Random Forest, SVM, Logistic Regression) on datasets processed by the RL-driven strategy. During model selection, we define *prior tolerance* as $Tolerance_{prior} = \frac{1}{|E|} \sum_{er \in E} \frac{P_{\text{DemandClean}}(M,er)}{P_{\text{do nothing}}(M,er)}$, where $P_{\text{do nothing}}$ represents performance without any cleaning and $P_{\text{DemandClean}}$ denotes the performance of model M under error rate er after cleaning. To further evaluate `DemandClean`'s efficacy, we compute *posterior tolerance* as

$$Tolerance_{post} = \frac{1}{|E|} \sum_{er \in E} \frac{P_{\text{DemandClean}}(M, er) - P_{\text{do nothing}}(M, er)}{P_{\text{repair all}}(M, er) - P_{\text{do nothing}}(M, er)},$$

with $P_{\text{repair all}}$ serving as the baseline when employing a "repair all" strategy. Comprehensive model selection is then conducted by considering basic performance, the degree of performance improvement from baseline, robustness (i.e., the inverse of the performance standard deviation across contamination scenarios), as well as cost and diversity metrics quantified from the type and frequency of RL actions, in addition to the aforementioned prior tolerance.

The weighted combination of these dimensions enables the identification of the optimal model tailored to the specific data environment and task demands. Thus, `DemandClean` achieves a balanced, interpretable, and actionable data quality management strategy suitable for complex real-world applications.

4 DEMONSTRATION

4.1 Experimental Results

As illustrated in Figure 3 and under the experimental configuration shown in Figure 2, we compared the cleaning action distributions produced by `DemandClean` for models across error rates from 10% to 90% with the proposed baseline strategies, thereby demonstrating its practical advantages. Specific conclusions are:

(a) Selective cleaning driven by model tolerance differences: From Figure 3(b), it is evident that models have varying tolerances to noise, with Random Forest being the most robust (T=0.91) and SVM the most sensitive (T=0.29). Figure 3(a) further demonstrates that `DemandClean` effectively captures this difference, performing more cleaning operations for sensitive models (e.g., SVM) while opting for "No Action" for robust models (e.g., Random Forest), significantly reducing redundant cleaning costs. Compared to "Repair All", `DemandClean` reduces repair actions by 80.0% and deletions by 80.7% for Random Forest, maintaining competitive performance across error rates.

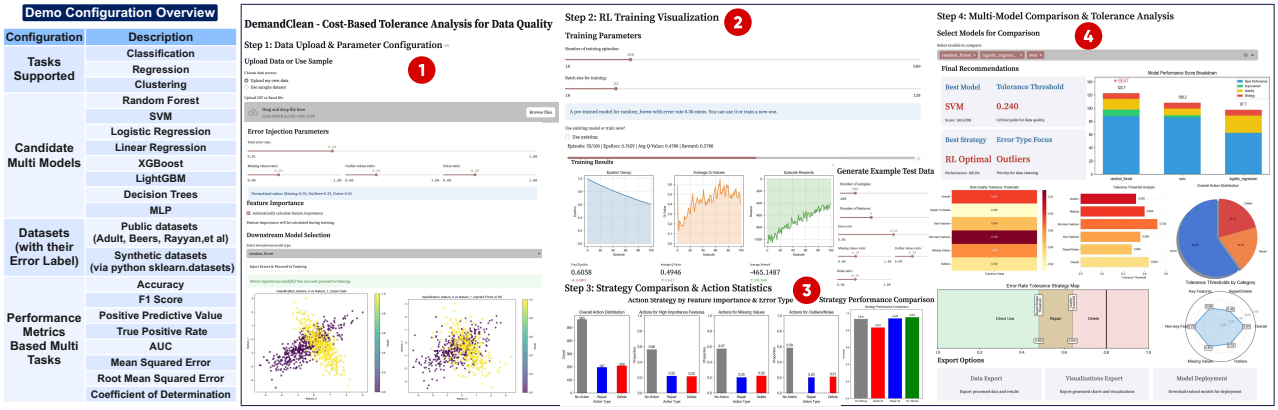

Figure 2: Pages demonstration of DemandClean with configuration overview, which the public dataset comes from [6].

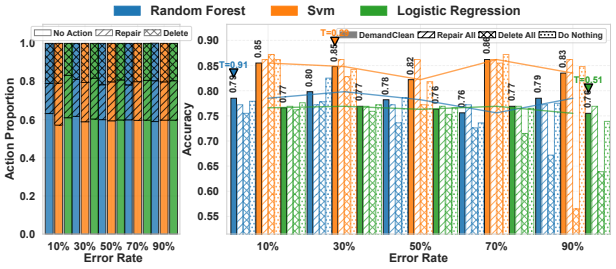

Figure 3: Comparison of cleaning action cost - multi-model benefits. Figure (b) marks the $Tolerance_{post}$(as T) of the corresponding posterior model.

(b) Adaptive Cleaning Based on Data Diversity and Authenticity: Figure 3(b) shows that DemandClean dynamically balances authenticity restoration and diversity retention through strategic cleaning actions, resulting in Random Forest outperforming the "Repair All" strategy. Conversely, the "Delete All" approach drastically reduces data diversity, causing performance degradation at high error rates. Figure 3(a) further highlights DemandClean's adaptability: with Random Forest, the proportion of "No Action" decreases as error rates rise to counteract increased noise severity, while for SVM, a more proactive cleaning strategy is consistently maintained. Thus, DemandClean effectively identifies critical data errors, implementing targeted interventions adapted to different models and error rates.

4.2 Demonstration Interface

As shown in Figure 2, DemandClean offers an intuitive and visually interactive demonstration interface, encompassing the complete workflow from data uploading, error injection, RL model training, to strategy visualization and results exporting:

Step 1: Data upload and parameter configuration. Users upload small clean datasets or auto-generated sklearn datasets via the "Upload Data" button. Users then configure error injection rates, assign proportions of error types, and set feature importance through sliders or numerical inputs. The system visualizes performance impacts before and after error injection, aiding user comprehension.

Step 2: RL agent training visualization. Upon initiating training, the system visualizes key metrics such as Epsilon Decay, Average Q-Values, and Episode Rewards in real-time. Users can monitor and intervene in training progress dynamically.

Step 3: Strategy comparison and action analysis. After training, users validate the performance of DemandClean against baseline strategies (Do Nothing, Delete All, Repair All). Detailed action distribution visualizations help users understand RL decisions and associated cost-benefit dynamics.

Step 4: Multi-model Comparison and Tolerance Analysis. The system supports simultaneous evaluation of various models, generating comprehensive metrics to recommend optimal models and preprocessing strategies. Visualizations illustrate how DemandClean dynamically adjusts strategies across varying error rates and model sensitivities.

Finally, users export the processed dataset, detailed analysis logs, explanatory reports, and the optimal RL agent model file, enabling high-quality data support for downstream applications.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2023YFB3308003), the National Natural Science Foundation of China (NSFC) (62202126, 62232005, 92267203), and the Natural Science Foundation Project of Heilongjiang Province of China (YQ2024F005).

REFERENCES

[1] Chengliang Chai, Jiabin Liu, Nan Tang, Ju Fan, Dongjing Miao, Jiayi Wang, Yuyu Luo, and Guoliang Li. 2023. GoodCore: Data-effective and Data-efficient Machine Learning through Coreset Selection over Incomplete Data. *Proc. ACM Manag. Data* 1, 2 (2023), 157:1–157:27. https://doi.org/10.1145/3589302

[2] Xiaoou Ding, Yichen Song, Hongzhi Wang, Chen Wang, and Donghua Yang. 2024. MTSClean: Efficient Constraint-based Cleaning for Multi-Dimensional Time Series Data. *Proc. VLDB Endow.* 17, 13 (2024), 4840–4852.

[3] Xiaoou Ding, Yichen Song, Hongzhi Wang, Donghua Yang, Chen Wang, and Jianmin Wang. 2024. Clean4TSDB: A Data Cleaning Tool for Time Series Databases. *Proc. VLDB Endow.* 17, 12 (2024), 4377–4380.

[4] Sanjay Krishnan, Michael J. Franklin, Ken Goldberg, and Eugene Wu. 2017. BoostClean: Automated Error Detection and Repair for Machine Learning. *CoRR* abs/1711.01299 (2017). arXiv:1711.01299

[5] Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. 2021. CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks. In *37th IEEE International Conference on Data Engineering, ICDE 2021*. IEEE, 13–24.

[6] Wei Ni, Xiaoye Miao, Xiangyu Zhao, Yangyang Wu, Shuwei Liang, and Jianwei Yin. 2024. Automatic Data Repair: Are We Ready to Deploy? *Proc. VLDB Endow.* 17, 10 (2024), 2617–2630.

[7] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proc. VLDB Endow.* 10, 11 (2017), 1190–1201.

[8] Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Qiu Jiantao, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. 2025. Harnessing Diversity for Important Data Selection in Pretraining Large Language Models. In *The Thirteenth International Conference on Learning Representations*.