Minimal Data Cleaning for Model Training by MinPrep

Cheng Zhen
Oregon State University
Corvallis, Oregon, USA
zhenc@oregonstate.edu

Prayoga
Oregon State University
Corvallis, Oregon, USA
prayoga@oregonstate.edu

Nischal Aryal
Oregon State University
Corvallis, Oregon, USA
aryaln@oregonstate.edu

Arash Termehchy
Oregon State University
Corvallis, Oregon, USA
termehca@oregonstate.edu

Alireza Aghasi
Oregon State University
Corvallis, Oregon, USA
alireza.aghasi@oregonstate.edu

ABSTRACT

Real-world datasets often contain dirty data, such as missing values, outliers, and inconsistencies. To train accurate machine learning (ML) models over these datasets, users typically spend a substantial amount of time and resources determining proper replacement values for dirty entries in training samples. This paper demonstrates MinPrep, a system that learns accurate ML models with minimal data repair. MinPrep allows users to define their model accuracy requirements and efficiently assesses the necessity of repairing dirty values to meet those requirements across various widely used ML paradigms. If repair is deemed unnecessary, MinPrep directly trains an accurate model over the clean data subset. When repair is required, the system selects a minimal set of dirty training samples to repair and trains the model over the resulting dataset. MinPrep provides theoretical guarantees of the output model's optimality for models with convex loss functions, alongside practical support for non-convex models trained via stochastic gradient descent. Our interactive demonstration emphasizes the significant time and effort savings achieved by minimal repair compared to repairing all dirty data, while consistently delivering accurate ML models.

PVLDB Reference Format:
Cheng Zhen, Prayoga, Nischal Aryal, Arash Termehchy, and Alireza Aghasi. Minimal Data Cleaning for Model Training by MinPrep. PVLDB, 19(12): 4618 - 4621, 2026.
doi:10.14778/3827998.3828080

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/PrayogaPrayoga/VLDB-Demo-2026/tree/main.

1 INTRODUCTION

The performance of a machine learning (ML) model is substantially dependent on the quality of its training data. Real-world training data often contain dirty data, such as missing values, outliers, and domain constraint violations. One may train an ML model by excluding training samples with dirty data. However, this method can lead to the loss of important information and introduce bias.

To train with dirty data, users typically repair dirty data by replacing each dirty value with a value that is considered reasonable and train their models over the resulting *repaired data*. To accurately repair data, users often need to identify the mechanisms behind dirty data. For example, to replace a missing value, it is crucial to determine whether the value is missing completely at random or based on observed values of some features. Depending on this mechanism, they build a (statistical) model for missing data and replace the missing values with some estimates derived from this model. In sensitive domains, e.g., medical, data is often repaired manually by domain experts. These steps require substantial amounts of time and manual effort. Surveys indicate that most users spend about 80% of their time preparing and repairing data [4].

Researchers have recently shown that models can be learned from dirty data without repairing it or only repairing a subset to learn accurate models while reducing time and effort [2, 3]. These methods are limited to specific types of models, e.g., K-nearest neighbor [2] or convex models [3]. Some also do not provide any theoretical guarantee on the minimality of their selected subset [3]. Ideally, users would like a system that determines whether data needs to be repaired and, if so, identifies the minimal subset of the data to repair for a diverse set of ML models.

We demonstrate *MinPrep*, which significantly reduces the data-cleaning effort for ML by 1) determining whether data needs to be repaired prior to training and 2) if it needs repairing, identifying the minimal subset of data to repair in order to learn accurate models. It supports a diverse set of convex models (e.g., linear regression, linear SVM, logistic regression) and non-convex models (e.g., Multi-Layer Perceptrons (MLP) and FT-Transformers [1]). *MinPrep* is efficient, with low computational overhead, scalable to large datasets, and provides accurate models with theoretical guarantees.

2 BACKGROUND

In this section, we review terminology and notation in ML, along with the types of dirty data that MinPrep handles.

Supervised Learning. Given n training examples, a training set consists of feature input $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^T$ and label output $\mathbf{y} = [y_1, ..., y_n]^T$. Given a target function f that maps $\mathbf{X}$ to $\mathbf{y}$, the training process finds the optimal model $\mathbf{w}^*$ that minimizes the training loss $L(f(\mathbf{X}, \mathbf{w}), \mathbf{y})$. Formally, $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} L(f(\mathbf{X}, \mathbf{w}), \mathbf{y})$.

Dirty Data. MinPrep covers dirty data that can be repaired by replacing values. In this demo, we showcase (1) *Missing value*: A value x_{ij} is considered a missing value if it is unknown (denoted as

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828080

null). (2) *Domain constraint violation*: a value in a table violates a domain constraint if it does not satisfy numerical constraints. (3) *Outlier*: a value x_{ij} is an outlier if it substantially deviates from the distribution of its corresponding feature.

Example 2.1. *In Table 1, Temperature and Humidity are features, and Rainfall is a binary label. New York's humidity (in red) is a missing value. London's humidity (in blue) violates the domain constraint (0-100%). Seattle's temperature (in orange) is an outlier, as it significantly deviates from the feature's distribution.*

Table 1: A training dataset for rain prediction

	Temperature(F)	Relative Humidity(%)	Rainfall
New York	75	*null*	-1
London	50	105	-1
Seattle	10	80	1

Repair. A repair $\mathbf{X}^r$ is a version of the original dataset $\mathbf{X}$ in which all dirty values are replaced with valid values while preserving the original tuples and feature dimensions.

Example 2.2. *We can obtain a valid repair* $\mathbf{X}^r$ *for Table 1 by replacing dirty values: for instance, by setting New York's humidity to 90, London's to 80 (satisfying the domain constraint), and Seattle's temperature to 60 (resolving the outlier). Conversely, deleting any rows does not constitute a valid repair, as it alters the dimension of* $\mathbf{X}$.

Set of Possible Repairs. The range of values that can be used to repair dirty data is often large. Thus, a large number of possible repairs may exist. We denote the set of all possible repairs as $\mathbf{X}^R$.

3 SYSTEM DESCRIPTION

We first introduce the concept and method to define and check the necessity of data repair for learning accurate models. If repair is deemed necessary, we propose approaches to repair the minimal subset of dirty data to learn accurate models. Finally, we outline the architecture of MinPrep based on our approaches.

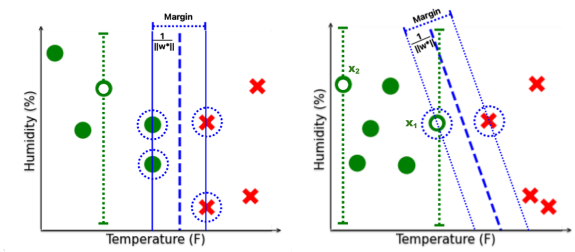
3.1 Certain and Approximately Certain Models

We first formally define a *certain model* (CM) that minimizes the training loss regardless of how dirty data is repaired. A model $\mathbf{w}^*$ is a CM if it is optimal across all possible repairs:

$$\forall \mathbf{X}^r \in \mathbf{X}^R, \mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} L(f(\mathbf{X}^r, \mathbf{w}), \mathbf{y})$$

where $\mathbf{X}^r$ is one possible repair, $\mathbf{X}^R$ is the set of all possible repairs, and $L(f(\mathbf{X}^r, \mathbf{w}), \mathbf{y})$ is the loss function. Intuitively, if a model is optimal (minimizes the training loss) for all possible repairs, then this model is a CM. When a CM exists, repairing dirty values does not affect model training and is thus unnecessary.

Example 3.1. *Consider the classification problem depicted in Figure 1, where the Support Vector Machine (SVM) model is used. The objective is to learn a linear boundary (blue rectangle) for rain prediction based on temperature and humidity features from various cities, which is consistent with the features and labels in Table 1. Figures 1a and 1b illustrate two sets of training examples, each with a missing humidity value. The green dashed line indicates the range of possible replacement values, with the empty circle denoting one possible replacement. In Figure 1a, different repairs lead to the same optimal model (decision boundary: blue dashed line), indicating the existence of a CM (*$\mathbf{w}^*$*). In other words, this CM,* $\mathbf{w}^*$*, creates the widest margin between examples with different labels (i.e., minimizes the training loss) for all possible repairs. Conversely, in Figure 1b, the optimal model varies depending on the chosen repair, suggesting that a CM does not exist.*



(a) Repair is not needed **(b) Only need to repair a subset**

Figure 1: Accurate Models via Zero or Subset Repair

The conditions for a CM may be too restrictive for many real-world datasets, as they require strict optimality across all possible repairs. In practice, users are often satisfied with a model that is *sufficiently close* to optimal. To address this, we relax the CM condition and propose the *approximately certain model* (ACM). Given a user-defined threshold $e \geq 0$, a model $\mathbf{w}^{\approx}$ is an ACM if, for every possible repair, its training loss is within e of the minimal training loss on that specific repair. If an ACM exists, users can confidently rely on it without any repairs, guaranteeing the model's accuracy meets their requirements while saving repair efforts.

3.2 Checking and Learning CM and ACM

A *baseline algorithm* for identifying and learning a CM or ACM consists of the following steps: (1) learning *possible models* from each possible repair individually, (2) a CM exists if all repairs share at least one mutual optimal model, and (3) an ACM exists if there is a model that has sufficiently minimal training loss across all repairs. Here, the set of possible repairs is often large (Section 2). Therefore, *learning models from all repairs can be extremely slow.*

MinPrep efficiently identifies and learns CMs without materializing all repairs, providing theoretical guarantees for linear regression, linear SVM, and SVMs with polynomial or RBF kernels. These algorithms work by assessing whether any repair of a dirty example impacts the training loss. For instance, in linear SVM, Algorithm 1 checks whether any dirty example is a support vector under any possible repair; if none do, a CM exists.

To identify an ACM, MinPrep evaluates models by solving the optimization problem below. The objective is to find a model whose training loss is close to the minimal training loss in every repair.

$$\min_{\mathbf{w}'} \sup_{\mathbf{X}^r \in \mathbf{X}^R} [L(f(\mathbf{X}^r, \mathbf{w}'), \mathbf{y}) - \min_{\mathbf{w}} L(f(\mathbf{X}^r, \mathbf{w}), \mathbf{y})]$$

Table 2: MinPrep vs. baselines when CM/ACM exist

Dataset	#Samp.	#Feat.	#Repaired			Time			Acc.		
			AC	Full	MinPrep	AC	Full	MinPrep	AC	Full	MinPrep
Gisette	13.5K	5K	249	675	**0**	18	54	**10**	97.03	**97.60**	97.35
Intel	1.85M	11	30	75080	**0**	356	13777	**276**	98.80	**98.90**	98.43

Table 3: MinPrep vs. baselines when CM/ACM do not exist

Dataset	#Samp.	#Feat.	#Repaired			Time			Acc.		
			AC	Full	MinPrep	AC	Full	MinPrep	AC	Full	MinPrep
Bankruptcy	8.4K	64	27	8402	**8**	101	7620	**20**	96.8	**97.0**	96.9
Online-Ed	7K	36	2019	7026	**745**	94	9624	**9**	63.6	**65.2**	**65.2**

For models with convex loss functions, this objective is convex and optimized via gradient descent, which guarantees finding an ACM if one exists. If the minimized objective value is below the threshold e, the model $\mathbf{w}'$ is an ACM. Otherwise, no ACM exists.

For non-convex models, since a global optimum is generally unachievable, we cannot directly compute the difference between a model's loss and the minimal loss across repairs. Instead, we relax the ACM condition to a *robust gradient stationarity* constraint. In non-convex optimization, it is standard practice to consider a model sufficiently optimal when it reaches a stationary point. Thus, we adapt our objective to find a model $\mathbf{w}'$ that minimizes the maximum gradient norm across all valid repairs. If this minimized worst-case gradient norm is bounded by e, the model is an ACM.

MinPrep avoids materializing all repairs and reduces the search space to a subset for checking and learning CMs and ACMs. Detailed algorithms and proofs for both CMs and ACMs are provided in [6].

Algorithm 1 Checking and learning CM for SVM

$Clean(\mathbf{x}) \leftarrow$ set of clean examples that contain no dirty data
$Dirty(\mathbf{x}) \leftarrow$ set of examples that contain dirty data
$\mathbf{w}^\diamond \leftarrow$ model trained on clean examples in $Clean(\mathbf{x})$
for $\mathbf{x}_i \in Dirty(\mathbf{x})$ **do**
  if $\exists \mathbf{x}_i^r$ that is a support vector with respect to $\mathbf{w}^\diamond$ **then**
    return "Certain models do not exist"
  end if
end for
return "A certain model $\mathbf{w}^\diamond$ exists"

Benefits vs. Overhead in Checking CMs and ACMs: When a CM or ACM exists, significant time and resource savings are achieved—especially as datasets grow and ML usage scales. When neither exists, checking introduces some overhead; however, *this cost is justified* for three reasons. First, substantial data-cleaning savings are achieved when a CM or ACM exists. Second, as shown in Table 2 (a subset of extensive experiments in [6] where CM/ACM exists), the cost of checking CM/ACM is minimal compared to baselines including ActiveClean (AC) [3] and repairing all dirty samples (Full) using KNN-based replacement. Third, when neither CM nor ACM exists, our algorithm identifies a subset of dirty samples to repair (Section 3.3), reducing the cost by avoiding full repair.

3.3 Minimal Repair

When a CM or ACM does not exist, data repair becomes necessary for learning accurate models. However, repairing all dirty data is often expensive, requiring extensive computational resources or manual effort from domain experts. MinPrep addresses this by identifying a *minimal repair*—the smallest subset of dirty training samples that, once repaired, guarantees the existence of a CM or ACM. For example, in Figure 1b, while two samples are incomplete, we may only need to repair $\mathbf{x}_1$ instead of both to achieve a CM, as $\mathbf{x}_2$ is not a support vector in any repair and never affects the SVM decision boundary. By repairing only a subset, users can drastically reduce data cleaning cost without sacrificing model accuracy.

Finding the exact minimal number of dirty samples to repair to achieve a CM or ACM is proven to be NP-hard [5]. To bypass this combinatorial complexity, MinPrep employs efficient approximation approaches. **For achieving CMs**, MinPrep supports models such as linear SVM and linear regression through tailored iterative approximation algorithms. **For achieving ACMs with convex losses**, MinPrep approximates minimal repair via a constrained optimization task, utilizing a continuous primal-dual stochastic gradient descent (SGD) approach. This method provides provable bounds for the approximation rate and a sublinear convergence rate guarantee for finding the minimal subset. **For achieving ACMs with non-convex losses** (e.g., neural networks), finding a global optimum is intractable. MinPrep instead approximates minimal repair via an adversarial gradient ascent method, approximating the minimal subset of samples necessary to guarantee the model reaches a robust stationary point across all possible repairs. Our extensive experiments, a subset of which is shown in Table 3, indicate that MinPrep's time savings scale with dataset size [5]. The overhead of finding the minimal subset becomes negligible compared to the massive savings of avoiding full repair. As summarized in Table 3, MinPrep identifies a significantly smaller subset of dirty samples to repair compared to the baselines. Consequently, employing a diffusion-model-based repair method, MinPrep reduces total program running time while achieving test accuracy comparable to the costly AC and full repair methods.

3.4 System Workflow

MinPrep takes a dataset where dirty samples are assumed to be pre-detected by the user. First, MinPrep checks if a CM exists for the target ML model; if so, it directly returns the optimal model without repair. If no CM exists, MinPrep prompts for an error tolerance threshold e. If strict optimality is required (no threshold is allowed), it identifies a minimal subset of dirty samples for the user to repair using their chosen method, and then returns a CM. If a threshold is provided, it checks for an ACM, returning the ACM if one exists; otherwise, it identifies a minimal subset of dirty samples for the user to repair using their chosen method, and then returns an ACM.

4 DEMONSTRATION PLAN

4.1 Models, Datasets, and Repair Methods

For CMs, MinPrep supports and demonstrates linear regression, linear SVM, SVMs with polynomial or RBF kernels, and approximated feed-forward neural networks. For ACMs, MinPrep supports ML models with either convex or non-convex loss functions, and we will demonstrate linear regression, linear SVM, logistic regression, Multi-Layer Perceptrons (MLP), and FT-Transformers [1].

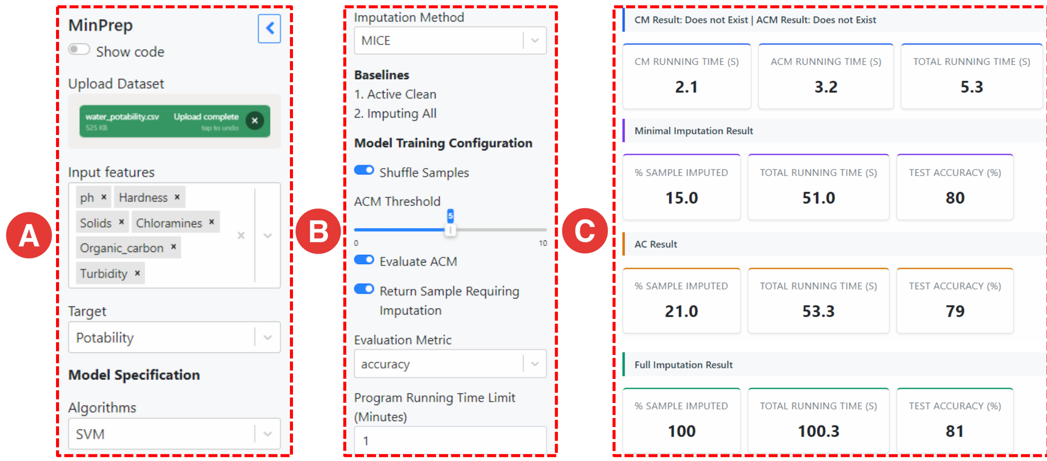


Figure 2: System Interface for MinPrep

We use fourteen real-world classification and regression datasets from environmental monitoring, healthcare, and sports domains. These datasets have varying proportions of dirty samples, ranging from about 10 to 7,000 features and from roughly 600 to 6.4 million samples, allowing users to select during the demo session.

MinPrep does not provide new repair methods. Instead, it is a system that supports existing methods for replacing dirty values to help users achieve no or minimal repair while keeping the output models accurate. In the demonstration, we will include a wide range of repair methods, including popular statistical, diffusion-based, and LLM-based methods, as well as custom user-defined programs to show the time and resources saved for various repair methods.

4.2 Backend Execution and System Interface

In the backend, when minimal repair is triggered, users provide a time budget for repair and training. For both CM and ACM routes, MinPrep executes its minimal repair concurrently with two baselines: 1) *Full Repair* (repairing all dirty samples), and 2) *ActiveClean* (for convex-loss models). This parallel execution provides a comparison of the methods, allowing users to evaluate trade-offs between repair time and ML model performance.

We implemented the MinPrep interface as a computational notebook extension (Figure 2). The *data and model view* (A) handles dataset file uploads and target ML model specification. In the *MinPrep configuration panel* (B), users choose a repair method, define the error tolerance threshold e for ACM, and set a time cap for repair and training. Finally, the *result visualization view* (C) indicates whether a CM or ACM exists alongside checking times; if neither exists, the system automatically performs minimal repair and displays a comparative analysis of downstream ML model performance, execution time, and the portion of repaired samples across the minimal repair approach and baselines.

4.3 Demonstrated Scenarios

Scenario 1: No Repair Needed. When a CM or ACM exists, we show that MinPrep directly outputs an accurate model efficiently, while Full Repair and ActiveClean spend substantial time and resources on unnecessary repairs. In such cases, dirty samples can be safely ignored without repair; however, simply dropping dirty data without MinPrep's verification lacks any guarantee of model correctness or optimality.

Scenario 2: Saving Repair Time. If neither a CM nor an ACM exists, we demonstrate that MinPrep finds the minimal repair and delivers accurate models faster than the baselines. We compare the minimal repair from MinPrep and the subset from ActiveClean and show that MinPrep often finds a smaller subset.

Scenario 3: Scalability. We demonstrate the scalability of MinPrep over very large datasets (many samples and/or features). We compare MinPrep with Full Repair and ActiveClean when neither a CM nor an ACM exists. We show that these methods often do not finish over very large datasets, particularly using complex repair methods, while MinPrep successfully returns accurate models within the same time constraints.

ACKNOWLEDGMENTS

This research was partially funded through NSF grant CNS-1941892 and the Industry-University Cooperative Research Center on Pervasive Personalized Intelligence.

REFERENCES

[1] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems* 34 (2021), 18932–18943.

[2] Bojan Karlaš, Peng Li, Renzhi Wu, Nezihe Merve Gürel, Xu Chu, Wentao Wu, and Ce Zhang. 2020. Nearest neighbor classifiers over incomplete information: From certain answers to certain predictions. *arXiv preprint arXiv:2005.05117* (2020).

[3] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. 2016. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment* 9, 12 (2016), 948–959.

[4] Steven Euijong Whang and Jae-Gil Lee. 2020. Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment* 13, 12 (2020), 3429–3432.

[5] Cheng Zhen, Nischal Aryal, Arash Termehchy, Garrett Biwer, Sankalp Patil, et al. 2025. Learning Accurate Models on Incomplete Data with Minimal Imputation. *arXiv preprint arXiv:2503.13921* (2025).

[6] Cheng Zhen, Nischal Aryal, Arash Termehchy, and Amandeep Singh Chabada. 2024. Certain and Approximately Certain Models for Statistical Learning. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–25.