Detecting Errors in Numerical Data via any Regression Model

Hang Zhou [1]   **Jonas Mueller** [2]   **Mayank Kumar** [2]   **Jane-Ling Wang** [1]   **Jing Lei** [3]

Abstract

Noise plagues many numerical datasets, where the recorded values in the data may fail to match the true underlying values due to reasons including: erroneous sensors, data entry/processing mistakes, or imperfect human estimates. Here we consider estimating *which* data values are incorrect along a numerical column. We present a model-agnostic approach that can utilize *any* regressor (i.e. statistical or machine learning model) which was fit to predict values in this column based on the other variables in the dataset. By accounting for various uncertainties, our approach distinguishes between genuine anomalies and natural data fluctuations, conditioned on the available information in the dataset. We establish theoretical guarantees for our method and show that other approaches like conformal inference struggle to detect errors. We also contribute a new error detection benchmark involving 5 regression datasets with real-world numerical errors (for which the true values are also known). In this benchmark and additional simulation studies, our method identifies incorrect values with better precision/recall than other approaches.

1. Introduction

Modern supervised machine learning has grown quite effective for most datasets thanks to development of highly-accurate models like random forests (RF), gradient-boosting machines (GBM), and neural networks (NN). Although it's generally assumed that the labels used for training are accurate, this is often not the case in real-world datasets (Müller & Markert, 2019; Northcutt et al., 2021; Kang et al., 2022; Kuan & Mueller, 2022a). For classification data, many techniques have been proposed to address this issue by mod-

ifying training objectives or directly estimating which data is erroneous (Jiang et al., 2018; Zhang & Sabuncu, 2018; Song et al., 2022; Northcutt et al., 2021).

In this paper, we consider methods to identify similar erroneous values in regression datasets where the labels are continuous-valued[1]. Incorrect numeric values lurk in real-world data for many reasons including: measurement error (e.g. imperfect sensors), processing error (e.g. incorrect transformation of some values), recording error (e.g. data entry mistakes), or bad annotators (e.g. poorly trained data labelers) (Wang & Mueller, 2022; Kuan & Mueller, 2022a; Nettle, 2018). We are particularly interested in straightforward *model-agnostic* approaches that can utilize **any** type of regression model to identify the errors. These desiderata ensure our approach is applicable across diverse datasets in practice and can take advantage of state-of-the-art regressors (including future regression models not yet invented). By fitting a regression model to predict each column in a numerical dataset (not necessarily the same type of model since different models may be better suited for predicting different target columns), we can use such model-agnostic approaches to estimate *all* erroneous values in the dataset. Once identified, the datapoints with erroneous values may be filtered out from a dataset or fixed via external confirmation of the correct value to replace the incorrect one.

To help prioritize review of the most suspicious values, we consider a *veracity score* for each datapoint that reflects how likely a specific value is correct or not. Many prediction-based scores have been explored, such as such as residuals, likelihood values, and entropies (Northcutt et al., 2021; Kuan & Mueller, 2022a; Wang & Mueller, 2022; Thyagarajan et al., 2022). While these methods are easy to implement and widely applicable, the uncertainties present in the observed data can impact prediction accuracy, consequently affecting both veracity scores and error detection. Two common types of uncertainties, epistemic and aleatoric, arise from a lack of observed data and intrinsic stochasticity in underlying relationships. Both types of uncertainties play a critical role in establishing the reliability of predictions.

[1]Department of Statistics, University of California, Davis, US [2]Cleanlab, Boston, US [3]Department of Statistics & Data Science, Carnegie Mellon University, Pittsburgh, US. Correspondence to: Jonas Mueller <jonas@cleanlab.ai>.

---

[1]Code to run our method: https://github.com/cleanlab/cleanlab. Code to reproduce paper: https://github.com/cleanlab/regression-label-error-benchmark

In this paper, we introduce novel *veracity scores* that incorporate both epistemic and aleatoric uncertainties. By accounting for these two types of uncertainties, an error detection procedure can more effectively distinguish between genuine anomalies and natural data fluctuations, ultimately resulting in more reliable identification of errors. Furthermore, we propose a simple yet efficient filtering procedure for eliminating potential errors. This algorithm automatically determines the number of errors to be removed and is compatible with any machine learning or statistical model. We introduce a comprehensive benchmark of datasets with naturally-occuring errors for which we also have corresponding ground truth values that can be used for evaluation. Results on this benchmark and extensive simulations illustrate the empirical effectiveness of our proposed approach to identify incorrect numerical values in a dataset.

2. Related Work

A significant body of research has focused on identifying numerical outliers or anomalies that depend on contextual or conditional information (Tang et al., 2013; Song et al., 2007; Hong & Hauskrecht, 2016). (Song et al., 2007) introduced the concept of conditional outliers, which model outliers as influenced by a set of behavioral attributes (e.g., temperature) that are conditionally dependent on contextual factors (e.g., longitude and latitude). (Valko et al., 2011) detected conditional anomalies using a training set of labeled examples, accounting for potential label noise. (Tang et al., 2013) proposed an algorithm for detecting contextual outliers in categorical data based on attribute-value sets. (Hong & Hauskrecht, 2016) employed conditional probability to detect anomalies in clinical applications. However, the methods proposed in these papers are model-specific and not universally applicable to all regression models, thereby limiting their utility in real-world data analysis involving complex data structures.

The random sample consensus (RANSAC) method proposed by Fischler & Bolles (1981) is a model-agnostic approach to error detection that iteratively identifies subsets of datapoints that are not well-fitted by a trained regression model. In contrast to RANSAC, our method effectively accounts for uncertainty in predictions from the regressor, which is crucial for differentiating confidently incorrect values from those that are merely inaccurately predicted. Conformal inference (Vovk et al., 2005; Lei et al., 2018; Bates et al., 2023) provides a framework to estimate the confidence in predictions from an arbitrary regressor, but we show here its direct application fares poorly when some data values are contaminated by noise.

3. Methods

3.1. Veracity scores

We consider a standard regression setting of covariates X and target values Y. Our goal is to utilize *any* fitted regression model to help detect observations Y_i where the recorded value in the dataset is actually incorrect (i.e. *corrupted*).

Our approach constructs a numeric *veracity score* for each datapoint X_i, which reflects how likely Y_i is correctly measured (based on how typical its value is given all of the other available information). For response variables Y that are categorical, the predictive likelihood/entropy-based scores proposed by (Kuan & Mueller, 2022a) have demonstrated effective performance for identifying erroneous labels via arbitrary classification models. Unlike standard classifiers, a typical regression model does not directly estimate the full conditional distribution of continuous target Y (most models simply output point estimates). Thus a *model-agnostic* method (that can use **any** regression model) to detect errors in numerical data cannot employ analogous likelihood/entropy measures.

Instead, the residual $\hat{S}_r(X_i, Y_i) = |Y_i - \hat{f}(X_i)|$ is a straightforward choice of score, where $\hat{f}$ represents the estimated regression function. Ideally, when the underlying relationship f is relatively simple and $\hat{f}(x)$ is a well-fitted regressor, datapoints with abnormally large $\hat{S}_r(X_i, Y_i)$ values are likely to be anomalous values that warrant suspicion. Throughout, all references to residuals and other prediction-based estimates (e.g. uncertainties) are assumed to be *out-of-sample*, i.e. produced for X_i from a copy of the regression model that was never fit to this datapoint. Out-of-sample predictions can be obtained for an entire dataset through K-fold cross-validation, and are important to ensure less biased estimates for our *veracity scores* that are less subject to overfitting. Complexities of real-world data analysis make error detection more challenging, for instance non-uniform epistemic or aleatoric uncertainty due to lack of observations or heteroscedasticity. In the context of prediction, epistemic uncertainty results from a scarcity of observed data that is a similar to a particular X, whose associated Y value is thus hard to guess. On the other hand, aleatoric uncertainty results from inherent randomness in the underlying relationship between X and Y that cannot be reduced with additional data of the same covariates (but could by enriching the dataset with additional covariates). Figure 1 illustrates these two types of uncertainties: the epistemic uncertainty is large at a datapoint x with few nearby datapoints, while the aleatoric uncertainty is large at x when the true underlying $Y|X = x$ is dispersed (e.g. a bimodal distribution).

After fitting a regression model in an expert manner, there are generally three reasons a residual $\hat{S}_r(X_i, Y_i)$ might be
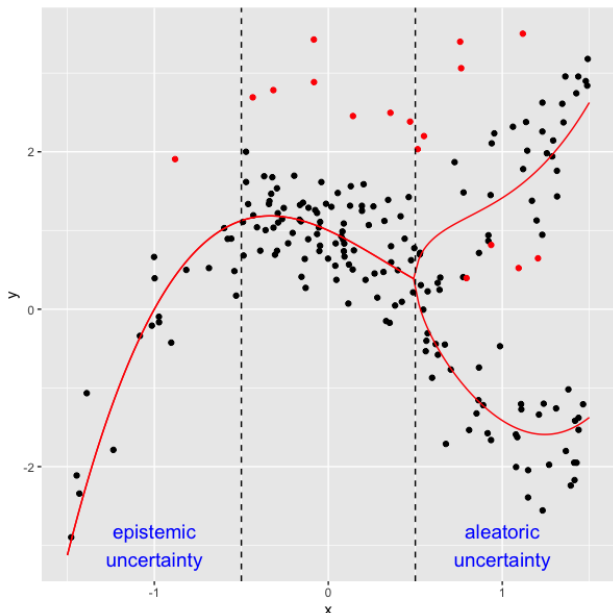
Figure 1. Dataset from our simulation study, with non-uniform epistemic and aleatoric uncertainties. Erroneous data are marked in red.

large:

- Y_i was incorrectly measured (i.e. a data error).

- The estimation quality of $\hat{f}(x)$ is poor around $x = X_i$, e.g. due to large epistemic uncertainty (lack of sufficiently many observations similar to X_i).

- There is high aleatoric uncertainty, i.e. the underlying conditional distribution over target values, $Y|x = X_i$, is not concentrated around a single value.

Therefore, the residual score might be suboptimal for precise error detection due to false positive scores arising simply due to large uncertainty. We instead propose two veracity scores that rescale the residual in order to account for both epistemic and aleatoric uncertainties:

$$\hat{S}_a(x, y) = \frac{\hat{S}_r(x, y)}{\hat{u}(x) + \hat{\sigma}(x)} \qquad \hat{S}_g(x, y) = \frac{\hat{S}_r(x, y)}{\sqrt{\hat{u}(x)\hat{\sigma}(x)}}.$$
$$(1)$$

Here, epistemic uncertainty estimate $\hat{u}(x) := \sqrt{\hat{\text{Var}}(\hat{f}(x))}$ is the standard deviation of $\tilde{f}(x)$ over many regressors $\tilde{f}$ (of the same type as $\hat{f}$) fit on bootstrap-resampled versions of the original data. Aleatoric uncertainty estimate $\hat{\sigma}(x) := \mathbb{E}(|\hat{f}(X) - Y| \mid X = x, \hat{f})$ is an estimate of the size of the regression error, producing by fitting a separate regressor

(of the same type as $\hat{f}$) to predict the residuals' size based on covariates X.

Our construction of $\hat{S}_a(x, y)$ and $\hat{S}_g(x, y)$ is a straightforward way to account for both epistemic and aleatoric uncertainties via their arithmetic or geometric mean. For datapoints where either uncertainty is abnormally large, the residuals are no longer reliable indicators. Thus presented with two datapoints whose Y values deviate greatly from the predicted values (high residuals of say equal magnitude), we should be more suspicious of the datapoint whose corresponding prediction uncertainty is lower.

3.2. Filtering procedure

A challenge arises in estimating regression model uncertainties from noisy data. Uncertainty estimates are based on the spread of $\hat{f}$ estimates, which are affected by the corrupted values in the dataset. Subsequent experiments in Section 5 and Supplement show this same issue plagues the probabilistic estimates required for conformal inference. Here we propose a straightforward approach to mitigate this issue: simply filter some of the top most-confident errors from the dataset, and refit the regression model and its uncertainty estimates on the remaining less noisy data. For the best results, we can iterate this process until the noise has been sufficiently reduced.

We use the following algorithm to iteratively filter potential errors in a data set $\mathbb{D}$. In Algorithm 1, the dataset $\mathbb{D} = (\boldsymbol{x}_i, y_i)_{i=1}^n$ imposes no restrictions on the covariates, allowing for numeric, text, images, or multimodal random variables in $\boldsymbol{x}_i$. The model $\mathcal{A}$ can be any parametric or non-parametric statistical regression model, or a machine learning model such as gradient boosting, random forest, or neural network. K represents the maximum proportion of erroneous values that the user believes may be present in $\mathbb{D}$, generally not expected to exceed 20%. To reduce computation time, the grid search over $k \leq K$ can be replaced by a binary search or a coarse-then-fine grid.

In this algorithm, we use the (out-of-sample) R^2 metric as the criterion to assess the performance of the current removal process. It is crucial to evaluate the R^2 metric on the entire dataset $\mathbb{D}$, rather than only on the remaining data. Evaluating it solely on the remaining data would cause the R^2 value to increase continuously as more data points are removed. Although the complete dataset $\mathbb{D}$ contains errors, its corresponding R^2 value should improve if $\hat{f}$ is trained on a dataset with fewer errors. Conversely, a smaller sample size may lead to poorer performance of model $\mathcal{A}$ and a decrease in the corresponding R^2 value evaluated on $\mathbb{D}$, especially if we have started removing data that has no errors. Like RANSAC (Fischler & Bolles, 1981), this approach iteratively discards data and re-fits $\hat{f}$, but each iteration in our approach utilizes the veracity scores.

Algorithm 1 Filtering procedure to reduce the amount of erroneous data

---

Input: Dataset $\mathbb{D}$; a regression model $\mathcal{A}$; the maximum proportion of corrupted data K_{err}.

1. Fit model $\mathcal{A}$ via K-fold cross-validation over the whole dataset, and compute veracity scores for each datapoint via out-of-sample predictions.

   for $k = 1, 2, \ldots, K_{err}$ **do**

      a. Remove $k\%$ of the datapoints with the worst veracity scores. Denote the indices of removed datapoints as IND_k.

      b. Re-fit the model $\mathcal{A}$ with the remaining data (again via K-fold cross-validation) and denote the estimated regression function $\hat{f}$. Calculate the out-of-sample R^2 performance of the resulting predictions: $1 - \sum_{i \in \mathbb{D}}(y_i - \hat{f}(x_i))^2 / \sum_{i \in \mathbb{D}}(y_i - \bar{y})^2$ over the *entire* dataset, where $\bar{y} = \frac{1}{|\mathbb{D}|}\sum_{i \in \mathbb{D}} y_i$.

   end for

2. Select the k^* that produces the largest R^2 among $k = 1, \ldots, K_{err}$.

   Output: Estimated corruption proportion k^*, indices of filtered data IND_{k^*}.

---

4. Theoretical Analysis

We use (X_i, Y_i) to denote a benign datapoint (whose Y-value is correct), where its distribution is given by $Y_i = f(X_i) + \epsilon_i(X_i)$, with $f(\cdot)$ denoting the true regression function and $\epsilon_i(X_i)$ denoting the traditional regression noise. On the other hand, an erroneous datapoint is represented as (X_i', Y_i'), with distribution $Y_i' = f(X_i') + \epsilon_i(X_i') + \epsilon_i^*(X_i')$, incorporating an additional corruption error: $\epsilon^*(X_i')$. In the scenario where the regression function is known, the $\hat{S}_r$ becomes: $S_r(X_i, Y_i) = |\epsilon_i(X_i)|$ for benign data (X_i, Y_i), and $S_r(X_i', Y_i') = |\epsilon_i(X_i') + \epsilon_i(X_i')^*|$ for erroneous data (X_i', Y_i'). Let F_x and G_x represent the cumulative distribution functions (CDF) of $|\epsilon(X)|$ and $|\epsilon(X) + \epsilon^*(X)|$ at $X = x$, respectively, where $\epsilon(x)$ and $\epsilon^*(x)$ are prototypes error functions of $\epsilon_i(x)$ and $\epsilon_i^*(x)$, and $\epsilon_i(x)$ and $\epsilon_i^*(x)$ are independent and identically distributed (i.i.d.) samples from $\epsilon(x)$ and $\epsilon^*(x)$.

Theorem 4.1. *Assume* $\mathbf{E}(\epsilon(X)|X) = 0$ *and is unimodal at 0,* $\epsilon(X)$ *and* $\epsilon^*(X)$ *are independent. If* $|\epsilon(X_i') + \epsilon^*(X_i')|$ *stochastically dominates* $|\epsilon(X_i)|$ *in the third order, that is,*

- $\int_{-\infty}^{x}\left[\int_{-\infty}^{z}\{F_{X_i}(t) - G_{X_i'}(t)\}\mathrm{d}t\right]\mathrm{d}z \geq 0$ *for all* x *and*

- $\int_{\mathbb{R}} x\mathrm{d}G_{X_i'}(x) \geq \int_{\mathbb{R}} x\mathrm{d}F_{X_i}(x)$.

Then, $\mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i')) \geq 1/2$.

Theorem 4.1 provides a sufficient condition ensuring that the probability of $S_r(X_i, Y_i) < S_r(X_i', Y_i')$ exceeds $1/2$. This implies that when the disparity between corrupted and clean target values is relatively large, the residual score can be effective for error detection. Third-order stochastic dominance is relatively weak and can be derived from first and second-order stochastic dominance. The subsequent corollary examines the case where the standard regression noise follows a Gaussian distribution, and the additional error corruption is a point mass at a. In that case, we have $F_{X_i}(t) - G_{X_i'}(t) \geq 0$, which implies that $|\epsilon(x') + \epsilon^*(x')|$ stochastically dominates $|\epsilon(x)|$ in the first order.

Corollary 4.2. *If* $\epsilon(x) \sim N(0,1)$, $\epsilon^*(x) = a$ *for all* x. *Then* $\mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i')) > 1/2$ *for all* $a \neq 0$ *and* $\mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i')) \to 1$ *exponentially as* $a \to \infty$.

When the estimated regression function $\hat{f}$ is consistent, $\hat{S}_r$ is asymptotically equivalent to the oracle case. The subsequent corollary directly follows from Theorem 4.1.

Corollary 4.3. *Denote* $\hat{f}$ *the estimator of* f *and* $\hat{S}_r(X_i, Y_i) := |\hat{f}(X_i) - Y_i|$ *the estimated residual scores. If* $\|\hat{f} - f\|_\infty \xrightarrow{p} 0$, *and* $G_{X_i'}$ *or* F_{X_i} *is absolutely continuous, then* $\mathbb{P}(\hat{S}_r(X_i, Y_i) < \hat{S}_r(X_i', Y_i')) = \mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i')) + o(1)$.

The following theorem illustrates the conditions under which our proposed scores outperform the residual-based approach.

Theorem 4.4. *Let* $\hat{S}_a(X_i, Y_i)$ *and* $\hat{S}_g(X_i, Y_i)$ *be the proposed veracity scores defined in 1,*

- *If* $\hat{u}(X_i) + \hat{\sigma}(X_i) \geq \hat{u}(X_i') + \hat{\sigma}(X_i')$, $\mathbb{P}(\hat{S}_a(X_i, Y_i) < \hat{S}_a(X_i', Y_i')) \geq \mathbb{P}(\hat{S}_r(X_i, Y_i) < \hat{S}_r(X_i', Y_i'))$.

- *If* $\hat{u}(X_i)\hat{\sigma}(X_i) \geq \hat{u}(X_i')\hat{\sigma}(X_i')$, $\mathbb{P}(\hat{S}_g(X_i, Y_i) < \hat{S}_g(X_i', Y_i')) \geq \mathbb{P}(\hat{S}_r(X_i, Y_i) < \hat{S}_r(X_i', Y_i'))$.

If $\hat{u}(X_i) > \hat{u}(X_i')$, that is, the bootstrap variance at $\hat{f}(X_i)$ is larger than that at $\hat{f}(X_i')$, it indicates greater epistemic uncertainty for (X_i, Y_i). Similarly, if the variance of the regression error at $\hat{f}(X_i)$ exceeds that at $\hat{f}(X_i')$, there is higher aleatoric uncertainty for (X_i, Y_i). In both instances, the residual might offer misleading information when assessing whether Y_i is corrupted or not. Theorem 4.4 suggests that, in the presence of both epistemic and aleatoric uncertainties in the data, our proposed scores demonstrate superior performance compared to the residual.

5. Simulation Study

Here we present two experiments using diverse simulated datasets to evaluate the empirical performance of our proposed veracity scores as well as the filtering procedure. Two underlying data-generating settings are considered, which are detailed in the Supplement. **Setting 1:** a 5-dimensional nonlinear relationship involving non-uniform epistemic and aleatoric uncertainties due to non-uniform covariate sampling and heteroscedasticity (depicted in Figure 1). **Setting 2:** a simpler 5-dimensional linear relationship. In both cases, we consider corruptions of different degrees a (larger magnitude values correspond to larger corruptions applied to the original Y values in the erroneous data). We consider settings in which we have clean training data and evaluate the error detection performance of methods in additional test data (no filtering needed), as well as settings where the entire dataset contains errors. In the latter case, we first apply our proposed filtering procedure, and subsequently compute veracity scores via the final model fit to the filtered dataset.

Our simulation study focuses on comparing our proposed veracity scores ($\hat{S}_a$ and $\hat{S}_g$) against the residual score $\hat{S}_r$, in order to investigate the empirical effect of additionally taking the regression uncertainties into account. In the Supplement, we compare many alternative veracity scores against the residual score $\hat{S}_r$ over a diverse set of real datasets, and find that none of these alternatives is able to consistently outperform the residual score (making $\hat{S}_r$ a worthy baseline). Even though we know the underlying relationship in these simulations, we nonetheless fit a variety of popular regressors that are often used in practice: Random Forest (RF) (Breiman, 2001) and Gradient Boosting with LightGBM (LGBM) (Ke et al., 2017).

5.1. Conformal inference using the proposed scores

We first examine the performance of our proposed veracity scores in conformal inference. To reduce the computational burden associated with grid search, we utilize the splitting conformal method, which is widely adopted due to its efficiency (Chernozhukov et al., 2021; Bates et al., 2023). The splitting conformal method requires a training set to fit the model, a calibration set to evaluate the rank of the scores, and a testing set to assess performance. For each setting, the training and calibration sets are generated based on the aforementioned settings without errors. For the testing set, 10% of the data are designated as errors with a corruption strength $a = -3, -2, -1, 1, 2, 3$, while the remaining 90% are benign datapoints, having the same distributions as those in the training and calibration sets. For each (X_i, Y_i) in the testing set, the conformal inference methodology enables us to obtain a p-value for the null hypothesis test $\mathcal{H}_{0,i} : X_i \sim P_0$ (Bates et al., 2023), where P_0 represents

---

Algorithm 2 Conformal Outlier Detection

Input: Training set $\mathbb{D}^{\text{train}}$, calibration set $\mathbb{D}^{\text{cal}}$, and testing set $\mathbb{D}^{\text{test}}$; a model $\mathcal{A}$; a conformal score $s(x, y)$; a target FDR level α.

1. Based on $\mathbb{D}^{\text{train}}$, obtain the estimated score $\hat{s}(X, Y)$.
2. Evaluate the scores $\{\hat{s}_i = \hat{s}(X_i, Y_i)\}_{i=1}^{\mathbb{D}^{\text{cal}}}$ for all datapoint in the calibration set, and denote the empirical CDF of $\{\hat{s}_i\}_{i=1}^n$ by $\hat{F}_{\hat{s}}$.
3. For each data point $(X_i, Y_i) \in \mathbb{D}^{\text{test}}$, get the conformal p-value $\hat{u}_i = (\hat{F}_{\hat{s}} \circ \hat{s})(X_i, Y_i)$.
4. Based on $\{\hat{u}_i\}_{i \in \mathbb{D}^{\text{test}}}$, apply BH procedure to determine which datapoint should be removed.

Input: Indices of outliers, i.e. datapoints expected to be erroneous.

---

the distribution of the benign data. The error detection problem is then transformed into a multiple testing problem, and we can apply the Benjamini-Hochberg (BH) procedure to control the false discovery rate (FDR). This entire procedure is demonstrated in Algorithm 2.

For each setting, we conduct 50 Monte-Carlo runs to mitigate the randomness that may occur in a single simulation. We use $\hat{S}_r$, $\hat{S}_a$ and $\hat{S}_g$ as the conformal scores $\hat{s}(x, y)$ in Algorithm 2 and the sample size $n = 200$ is the same for $\mathbb{D}^{\text{train}}$, $\mathbb{D}^{\text{cal}}$, and $\mathbb{D}^{\text{test}}$. For each run, we calculate the corresponding *False Discovery Rate* (FDR), the proportion of benign data among the test points incorrectly reported as errors, and the *Power*, the proportion of errors in the testing set correctly identified as errors. For each setting, two scenarios are considered, the first scenario is the typical conformal scenario where the training and calibration sets have no errors; while for the second scenario, the training and calibration sets are also contaminated and the errors proportion is the same to the testing set.

Table 1 presents the average FDR and Power for each setting where the training and calibration sets are clean. For Setting 1, which contains epistemic and aleatoric uncertainty, our proposed scores outperform the residual scores in both FDR and Power. For Setting 2, where the residual scores are expected to perform well, our proposed scores perform very closely to the residual scores and even surpass them in some cases.

Table 2 shows that the conformal method fails when the training and calibration sets are contaminated. This is because the validity of conformal inference crucially relies on the exchangeability (or some variant of exchangeability) between the calibration set and the future observation, thus if the calibration set is contaminated, the conformal prediction set will have biased coverage. Note that in the scenario where training, calibration and testing set are all equally noisy, this can be equivalently viewed as the performance

Table 1. Average FDR and Power for different settings by conformal inference. Each setting is trained on clean labels with Random Forest regressor. The target FDR is 10%.

Setting 1							
corruption strength		-3	-2	-1	1	2	3
FDR	$\hat{S}_r$	0.16	0.30	0.44	0.37	0.19	0.16
	$\hat{S}_a$	0.13	0.16	0.35	0.24	0.14	0.13
	$\hat{S}_g$	0.12	0.18	0.36	0.34	0.14	0.13
Power	$\hat{S}_r$	0.31	0.09	0.04	0.02	0.08	0.30
	$\hat{S}_a$	0.71	0.41	0.06	0.06	0.38	0.72
	$\hat{S}_g$	0.71	0.38	0.06	0.05	0.36	0.73
Setting 2							
corruption strength		-3	-2	-1	1	2	3
FDR	$\hat{S}_r$	0.13	0.15	0.31	0.41	0.20	0.11
	$\hat{S}_a$	0.13	0.15	0.34	0.27	0.20	0.13
	$\hat{S}_g$	0.13	0.13	0.36	0.29	0.20	0.13
Power	$\hat{S}_r$	0.77	0.24	0.03	0.04	0.29	0.81
	$\hat{S}_a$	0.77	0.33	0.06	0.05	0.37	0.79
	$\hat{S}_g$	0.77	0.32	0.05	0.04	0.37	0.80

Table 2. Average FDR and Power for different settings by conformal inference. Each setting is trained on 10% contaminated data with Random Forest regressor. The target FDR is 10%.

Setting 1							
corruption strength		-3	-2	-1	1	2	3
FDR	$\hat{S}_r$	0.03	0.13	0.38	0.48	0.14	0.08
	$\hat{S}_a$	0.01	0.03	0.18	0.29	0.04	0.00
	$\hat{S}_g$	0.01	0.03	0.23	0.29	0.03	0.00
Power	$\hat{S}_r$	0.05	0.04	0.01	0.00	0.03	0.05
	$\hat{S}_a$	0.05	0.05	0.03	0.02	0.03	0.05
	$\hat{S}_g$	0.05	0.05	0.02	0.02	0.04	0.06
Setting 2							
corruption strength		-3	-2	-1	1	2	3
FDR	$\hat{S}_r$	0.00	0.05	0.31	0.23	0.01	0.01
	$\hat{S}_a$	0.00	0.00	0.17	0.23	0.01	0.00
	$\hat{S}_g$	0.00	0.01	0.15	0.20	0.01	0.00
Power	$\hat{S}_r$	0.05	0.05	0.03	0.04	0.04	0.06
	$\hat{S}_a$	0.04	0.05	0.02	0.04	0.05	0.05
	$\hat{S}_g$	0.04	0.05	0.02	0.04	0.05	0.05

for identifying errors in a given dataset.

5.2. Filtering procedure

In this subsection, we examine the numerical performance of our proposed filtering procedure. For each setting in each Monte-Carlo run, we have $n = 200$ data points with 10% errors and corruption strength a. Given error detection can be viewed as an information retrieval problem, we follow Kuan & Mueller (2022a) and use the Area Under the Precision-Recall Curve (AUPRC) metric to evaluate various veracity scores. AUPRC quantifies how well these scores are able to rank erroneous datapoints above those with correct values, which is essential to effectively handle errors in practice.

Table 3 presents the average AUPRC based on the original dataset, proportion of corruptions removed, proportion of corruptions in the remaining data, and AUPRC based on the remaining data for 50 Monte-Carlo runs. We observe that in both Setting 1 and Setting 2, the corruption proportions in the remaining data decrease as the corruption strength increases, and the AUPRC improves after running our removal algorithm. Furthermore, the removed proportion is very close to the true corruption proportion in the original dataset. In Setting 1, which includes epistemic and aleatoric uncertainty, our proposed scores $\hat{S}_a$ and $\hat{S}_g$ outperform the residual score in AUPRC across all scenarios. In Setting 2, where the underlying uncertainty should be relatively uniform, our proposed scores perform similarly to the residual

score.

6. Benchmark with Real Data and Real Errors

Here, we evaluate the performance of our proposed methods using five publicly available datasets. For each dataset, we have an observed target value that we use for fitting regression models and computing veracity scores and other estimates. For evaluation, we also have a true target value available in each dataset (not made available to any of our estimation procedures). For instance, in the **Air CO** air quality dataset, the observed target values stem from an inferior sensor device, whereas the true target values stem from a much high-quality sensor placed in the same locations. Detailed information regarding these datasets can be found in Section B of the Supplement.

The proportion of actual errors contained in each dataset varies. We first evaluate the performance of our proposed scores compared to the residual when the regression models are trained on clean labels. We consider four types of regression models: Gradient Boosting with LightGBM (Ke et al., 2017), Feedforward Neural Network (NN) (Gurney, 1997), Random Forest (Breiman, 2001), and a Weighted Ensemble of these models fit via Ensemble Selection (WE) (Caruana et al., 2004). Estimates are evaluated using four metrics popular in information retrieval applications: area under the receiver operating characteristic curve (AUROC),

Table 3. Model is trained by LightGBM and cross-validation is used in calculating the scores.

corruption		Setting 1		
strength		AUPRC before	removed prop	error prop	AUPRC after
a=-3	$\hat{S}_r$	0.60	11.74%	5.13%	0.66
	$\hat{S}_a$	0.64	11.22%	4.98%	0.72
	$\hat{S}_g$	0.62	9.60%	5.62%	0.71
a=-2	$\hat{S}_r$	0.33	12.34%	7.55%	0.34
	$\hat{S}_a$	0.40	11.64%	6.89%	0.44
	$\hat{S}_r$	0.37	11.74%	7.01%	0.43
a=-1	$\hat{S}_r$	0.15	11.26%	9.37%	0.16
	$\hat{S}_a$	0.16	11.98%	8.82%	0.19
	$\hat{S}_g$	0.16	10.66%	9.14%	0.18
a=1	$\hat{S}_r$	0.16	11.04%	9.39%	0.15
	$\hat{S}_a$	0.21	12.72%	8.29%	0.20
	$\hat{S}_g$	0.20	13.88%	8.24%	0.19
a=2	$\hat{S}_r$	0.33	11.18%	7.01%	0.31
	$\hat{S}_a$	0.40	11.36%	6.11%	0.43
	$\hat{S}_g$	0.38	11.76%	6.19%	0.40
a=3	$\hat{S}_r$	0.59	10.56%	5.29%	0.67
	$\hat{S}_a$	0.62	10.04%	5.24%	0.70
	$\hat{S}_g$	0.61	11.34%	5.20%	0.68
corruption		Setting 2		
strength		AUPRC before	removed prop	error prop	AUPRC after
a=-3	$\hat{S}_r$	0.84	14.64%	2.62%	0.94
	$\hat{S}_a$	0.78	14.88%	2.58%	0.93
	$\hat{S}_g$	0.78	15.36%	2.50%	0.93
a=-2	$\hat{S}_r$	0.63	12.92%	4.43%	0.70
	$\hat{S}_a$	0.60	13.14%	4.52%	0.68
	$\hat{S}_r$	0.60	13.18%	4.54%	0.66
a=-1	$\hat{S}_r$	0.25	11.38%	8.01%	0.28
	$\hat{S}_a$	0.24	12.08%	8.09%	0.27
	$\hat{S}_g$	0.24	12.00%	8.08%	0.27
a=1	$\hat{S}_r$	0.25	11.66%	7.86%	0.28
	$\hat{S}_a$	0.24	10.92%	8.15%	0.27
	$\hat{S}_g$	0.24	11.72%	8.01%	0.27
a=2	$\hat{S}_r$	0.59	12.48%	4.71%	0.71
	$\hat{S}_a$	0.57	13.50%	4.79%	0.69
	$\hat{S}_g$	0.55	12.56%	5.07%	0.66
a=3	$\hat{S}_r$	0.84	11.66%	7.86%	0.95
	$\hat{S}_a$	0.79	10.92%	8.15%	0.92
	$\hat{S}_g$	0.79	11.72%	8.01%	0.91

AUPRC, lift at T (where T is the true underlying number of errors in dataset), and lift at 100. Each metric evaluates how well a method is able to retrieve or rank the corrupted datapoints ahead of the benign data.

Table 4 shows the average improvement of our proposed scores compared to the residual scores. For example, the first four numbers in the first row represent $(\mathrm{AUROC}(\hat{S}_a) - \mathrm{AUROC}(\hat{S}_r))/\mathrm{AUROC}(\hat{S}_r)$ corresponding to the Light-GBM, Neural Network, Random Forest, and Weighted Ensemble models. A larger positive percentage indicates better performance of our proposed scores. Table 4 reveals that our proposed scores outperform the residuals $\hat{S}_r$ in most cases.

Next, we compare our proposed filtering procedure with the RANSAC algorithm (Fischler & Bolles, 1981). Here we separately run each of these data filtering procedures, and then compute three veracity scores $\hat{S}_a, \hat{S}_g, \hat{S}_r$ from the same type of model fit to the filtered data. Some values in the table are left blank because the RANSAC algorithm from the `scikit-learn` package can only handle numeric covariates, which excludes the "Stanford Politeness Wiki" dataset. Table 5 and Table 5 show that, when the entire dataset may contain corrupted values, our proposed filtering procedure generally performs better than RANSAC, which tends to overestimate or underestimate the corruption proportions. Furthermore, our veracity score combined with our filtering procedure leads to the best overall error detection performance across these datasets.

7. Discussion

For detecting erroneous numerical values in real-world data, this paper introduces novel veracity scores to quantify how likely each datapoint's Y-value has been corrupted. When we have a clean training dataset that is used to detect errors in subsequent test data, these veracity scores significantly outperform residuals alone, by properly accounting for epistemic and aleatoric uncertainties. When the entire dataset may contain corruptions, the uncertainty estimates degrade. For this setting, we introduce a filtering procedure that reduces the amount of corruption in the dataset. Such filtering helps us obtain better uncertainty estimates that result in more effective veracity scores for detecting erroneous values. We present a comprehensive benchmark of real-world regression datasets with naturally occuring erroneous values, over which our proposed approaches outperform other methods. All of our proposed approaches work with any regression model, which makes them widely applicable. Armed with our methods to detect corrupted data, data scientists will be able to produce more reliable models/insights out of noisy datasets.

Table 4. Average improvement of arithmetic and geometric scores compared with residual. Models are trained on clean labels and evaluated on the whole dataset with different methods.

data set	metric	$\hat{S}_a$			
		LGBM	NN	RF	WE
Air CO	auroc	-0.37%	-0.40%	2.57%	1.48%
	auprc	38.47%	13.33%	41.58%	138.76%
	lift_at_num_errors	23.48%	10.53%	15.76%	64.20%
	lift_at_100	44.00%	70.00%	39.71%	156.76%
metaphor	auroc	3.10%	1.20%	5.05%	4.62%
	auprc	64.76%	62.95%	92.87%	64.73%
	lift_at_num_errors	21.95%	39.13%	39.76%	49.47%
	lift_at_100	55.00%	53.85%	74.42%	66.10%
stanford stack	auroc	1.20%	0.96%	0.99%	0.48%
	auprc	11.53%	13.30%	10.77%	1.76%
	lift_at_num_errors	11.92%	11.88%	10.00%	6.70%
	lift_at_100	6.38%	9.89%	6.38%	0.00%
stanford wiki	auroc	0.85%	-0.91%	1.01%	1.01%
	auprc	13.54%	5.29%	8.65%	8.76%
	lift_at_num_errors	6.07%	2.66%	4.22%	6.01%
	lift_at_100	14.94%	6.98%	6.38%	5.26%
telomere	auroc	-0.34%	-0.08%	0.15%	0.00%
	auprc	0.77%	-1.18%	4.38%	0.14%
	lift_at_num_errors	-3.41%	-3.89%	8.37%	0.22%
	lift_at_100	3.09%	0.00%	1.01%	0.00%

data set	metric	$\hat{S}_g$			
		LGBM	NN	RF	WE
Air CO	auroc	-1.17%	-3.25%	2.92%	0.64%
	auprc	44.42%	-1.39%	43.21%	117.49%
	lift_at_num_errors	24.35%	-3.51%	18.79%	55.56%
	lift_at_100	50.00%	35.00%	41.18%	127.03%
metaphor	auroc	3.21%	3.04%	6.56%	7.72%
	auprc	70.80%	99.15%	114.37%	73.55%
	lift_at_num_errors	25.61%	43.48%	55.42%	56.84%
	lift_at_100	67.50%	96.15%	104.65%	66.10%
stanford stack	auroc	1.24%	0.59%	1.08%	0.66%
	auprc	11.72%	10.26%	11.31%	2.21%
	lift_at_num_errors	13.25%	5.00%	11.88%	6.70%
	lift_at_100	6.38%	8.79%	6.38%	0.00%
stanford wiki	auroc	1.14%	-1.02%	1.10%	1.44%
	auprc	14.99%	5.16%	8.61%	10.12%
	lift_at_num_errors	6.54%	3.19%	4.22%	9.44%
	lift_at_100	14.94%	5.81%	6.38%	5.26%
telomere	auroc	-1.39%	-1.33%	0.15%	-0.28%
	auprc	-10.15%	-15.50%	4.29%	-2.81%
	lift_at_num_errors	-15.61%	-20.14%	7.14%	-6.87%
	lift_at_100	3.09%	-3.00%	1.01%	1.01%

Table 5. Our proposed filtering procedure applied with a LightGBM regressor. Veracity score is computed after data filtering.

data set		Our proposed filtering procedure				
		original error prop	removed prop	error prop after	AUROC	AUPRC
Air CO	$\hat{S}_r$		2.00%	5.03%	0.58	0.08
	$\hat{S}_a$	5.13%	7.01%	4.89%	0.58	0.07
	$\hat{S}_g$		6.00%	4.98%	0.54	0.06
metaphor	$\hat{S}_r$		5.03%	6.33%	0.93	0.09
	$\hat{S}_a$	6.55%	17.99%	5.98%	0.94	0.09
	$\hat{S}_g$		19.01%	6.01%	0.94	0.09
Stanford stack	res		3.06%	10.09%	0.93	0.65
	$\hat{S}_a$	11.86%	5.01%	8.17%	0.94	0.68
	$\hat{S}_g$		4.03%	8.79%	0.94	0.73
Stanford wiki	$\hat{S}_r$		22.04%	13.31%	0.82	0.60
	$\hat{S}_a$	22.96%	24.03%	12.55%	0.82	0.64
	$\hat{S}_g$		10.07%	17.22%	0.82	0.64
telomere	$\hat{S}_r$		16.00%	0.04%	0.99	0.78
	$\hat{S}_a$	4.66%	18.00%	0.07%	0.99	0.83
	$\hat{S}_g$		22.00%	0.10%	0.97	0.73

Table 6. RANSAC algorithm applied with a LightGBM regressor. eracity score is computed after data filtering.

data set		RANSAC in sklearn				
		original error prop	removed prop	error prop after	AUROC	AUPRC
Air CO	$\hat{S}_r$				0.56	0.08
	$\hat{S}_a$	5.13%	2.11%	4.90%	0.55	0.08
	$\hat{S}_g$				0.54	0.07
metaphor	$\hat{S}_r$				0.64	0.10
	$\hat{S}_a$	6.55%	43.58%	4.88%	0.64	0.10
	$\hat{S}_g$				0.62	0.09
Stanford stack	res				0.95	0.69
	$\hat{S}_a$	11.86%	72.37%	0.44%	0.93	0.65
	$\hat{S}_g$				0.94	0.67
Stanford wiki	$\hat{S}_r$					
	$\hat{S}_a$	22.96%				
	$\hat{S}_g$					
telomere	$\hat{S}_r$				0.99	0.77
	$\hat{S}_a$	4.66%	0.62%	4.18%	0.99	0.80
	$\hat{S}_g$				0.97	0.72

References

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1):149–178, 2023.

Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Caruana, R., Niculescu-Mizil, A., Crew, G., and Ksikes, A. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 18, 2004.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Gurney, K. *An introduction to neural networks*. CRC press, 1997.

Hong, C. and Hauskrecht, M. Multivariate conditional outlier detection and its clinical application. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Hu, X. and Lei, J. A distribution-free test of covariate shift using conformal prediction. *arXiv preprint arXiv:2010.07147*, 2020.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pp. 2304–2313. PMLR, 2018.

Kang, D., Arechiga, N., Pillai, S., Bailis, P. D., and Zaharia, M. Finding label and model errors in perception data with learned observation assertions. In *Proceedings of the 2022 International Conference on Management of Data*, pp. 496–505, 2022.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Kuan, J. and Mueller, J. Model-agnostic label quality scoring to detect real-world label errors. In *ICML DataPerf Workshop*, 2022a.

Kuan, J. and Mueller, J. Back to the basics: Revisiting out-of-distribution detection baselines. In *ICML Workshop on Principles of Distribution Shift*, 2022b.

Lei, J. and Wasserman, L. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76 (1):71–96, 2014.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Müller, N. M. and Markert, K. Identifying mislabeled instances in classification datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Nettle, D. Code and data archive for nettle et al. 'consequences of measurement error in qpcr telomere data: A simulation study'. December 2018. doi: 10. 5281/zenodo.2615735. URL https://doi.org/10. 5281/zenodo.2615735.

Northcutt, C., Jiang, L., and Chuang, I. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Song, X., Wu, M., Jermaine, C., and Ranka, S. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering*, 19(5):631–645, 2007.

Tang, G., Bailey, J., Pei, J., and Dong, G. Mining multidimensional contextual outliers from categorical relational data. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, pp. 1–4, 2013.

Thyagarajan, A., Snorrason, E., Northcutt, C., and Mueller, J. Identifying incorrect annotations in multi-label classification data. *arXiv preprint arXiv:2211.13895*, 2022.

Valko, M., Kveton, B., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. Conditional anomaly detection with soft harmonic functions. In *2011 IEEE 11th international conference on data mining*, pp. 735–743. IEEE, 2011.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Wang, W.-C. and Mueller, J. Detecting label errors in token classification data. *arXiv preprint arXiv:2210.03920*, 2022.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Appendix: Detecting Errors in Numerical Data via any Regression Model

A. Proofs of theorems in Section 4

of Theorem 4.1. Note that

$$\mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i')) = \iint_{x>y} \mathrm{d}G_{X_i'}(x)\mathrm{d}F_{X_i}(y) = \int_{-\infty}^{\infty}\int_{-\infty}^{x} \mathrm{d}F_{X_i}(y)\mathrm{d}G_{X_i'}(x)$$

$$= \int_{-\infty}^{\infty} F_{X_i}(x)\mathrm{d}G_{X_i'}(x).$$

If G stochastically dominates F in the third order, then $\mathbf{E}_G U(x) \geq \mathbf{E}_F U(x)$, for all nondecreasing, concave utility functions U that are positively skewed. Since $F_{X_i}(x)$ is the CDF of absolute value of the regression error, it is obviously nondecreasing and positively skewed. To see $F_{X_i}(x)$ is concave, note that

$$\frac{\mathrm{d}^2 F_{X_i}(x)}{\mathrm{d}x^2} = \frac{\mathrm{d}f_{X_i}(x)}{\mathrm{d}x} + \frac{\mathrm{d}f_{X_i}(-x)}{\mathrm{d}x} = 2\frac{\mathrm{d}f_{X_i}(x)}{\mathrm{d}x} \leq 0, \text{ for } x > 0,$$

where $f_{X_i}(x)$ is the density function of $F_{X_i}(x)$ and the last inequality is from $\mathbf{E}(\epsilon(x)) = 0$ and is unimodal at 0. Thus, $F_{X_i}(x)$ is a concave utility function and

$$\int_{-\infty}^{\infty} F_{X_i}(x)\mathrm{d}G_{X_i'}(x) \geq \int_{-\infty}^{\infty} F_{X_i}(x)\mathrm{d}F_{X_i}(x) = \int_0^1 x\mathrm{d}x = \frac{1}{2},$$

which complete the proof. $\square$

of Corollary 4.2. Under the assumption of Corollary 4.2, $\epsilon_i(X_i) \sim N(0, 1)$, $\epsilon(X_i') + \epsilon^*(X_i') \sim N(a, 1)$. Thus

$$\mathrm{d}G_{X_i'}(x) = \left\{ \frac{1}{\sqrt{2\pi}}e^{-\frac{(x-a)^2}{2}} + \frac{1}{\sqrt{2\pi}}e^{-\frac{(x+a)^2}{2}} \right\}\mathrm{d}x, \mathrm{d}F_{X_i}(y) = \frac{2}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\mathrm{d}y.$$

Denote

$$f(a) := \mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i'))$$

$$= \int_0^\infty \int_0^x \left\{ \frac{1}{\sqrt{2\pi}}e^{-\frac{(x-a)^2}{2}} + \frac{1}{\sqrt{2\pi}}e^{-\frac{(x+a)^2}{2}} \right\}\frac{2}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\mathrm{d}x\mathrm{d}y$$

$$= \frac{1}{2}\sqrt{\frac{2}{\pi}}\int_0^\infty \left\{ e^{-\frac{(x-a)^2}{2}} + e^{-\frac{(x+a)^2}{2}} \right\}\mathrm{Erf}\left(\frac{x}{\sqrt{2}}\right)\mathrm{d}x,$$

where $\mathrm{Erf}(z) = 2\pi^{-1/2}\int_0^z e^{-t^2}\mathrm{d}t$ is the error function. Note that $f(0) = 1/2$, we hope to show

$$f(a) - f(0) = \frac{1}{2}\sqrt{\frac{2}{\pi}}\int_0^\infty \left\{ e^{-\frac{(x-a)^2}{2}} + e^{-\frac{(x+a)^2}{2}} - 2e^{-\frac{x^2}{2}} \right\}\mathrm{Erf}\left(\frac{x}{\sqrt{2}}\right)\mathrm{d}x \geq 0.$$

A change of variable leads to

$$\int_0^\infty \left\{ e^{-\frac{(x-a)^2}{2}} + e^{-\frac{(x+a)^2}{2}} - 2e^{-\frac{x^2}{2}} \right\}\mathrm{Erf}\left(\frac{x}{\sqrt{2}}\right)\mathrm{d}x$$

$$= \int_0^\infty \left\{ e^{-\frac{(x-a)^2}{2}} + e^{-\frac{(x+a)^2}{2}} - 2e^{-\frac{x^2}{2}} \right\}\int_0^{\frac{x}{\sqrt{2}}} e^{-t^2}\mathrm{d}t\mathrm{d}x$$

$$= \frac{2}{\sqrt{\pi}}\int_0^\infty e^{-t^2}\int_{\sqrt{2}t}^\infty \left\{ e^{-\frac{(x-a)^2}{2}} + e^{-\frac{(x+a)^2}{2}} - 2e^{-\frac{x^2}{2}} \right\}\mathrm{d}x\mathrm{d}t$$

$$= \sqrt{2}\int_0^\infty e^{-t^2}\left\{ 2\mathrm{Erf}(t) - \mathrm{Erf}\left(t - \frac{a}{\sqrt{2}}\right) - \mathrm{Erf}\left(t + \frac{a}{\sqrt{2}}\right) \right\}\mathrm{d}t$$

$$= \sqrt{2}\int_0^\infty e^{-t^2}\left\{ \int_{t-\frac{a}{\sqrt{2}}}^t e^{-u^2}\mathrm{d}u - \int_t^{t+\frac{a}{\sqrt{2}}} e^{-u^2}\mathrm{d}u \right\}\mathrm{d}t.$$

Since $\int_{t-\frac{a}{\sqrt{2}}}^{t} e^{-u^2}\,\mathrm{d}u - \int_{t}^{t+\frac{a}{\sqrt{2}}} e^{-u^2}\,\mathrm{d}u$ is always positive due to the monotonicity of e^{-u^2}, which implies $f(a) - f(0) > 0$. Thus $\mathbb{P}(S_r(X_i, Y_i) < S_r(X_i', Y_i')) > 1/2$. Furthermore, note that $\lim_{a\to\infty} 2\,\mathrm{Erf}(t) - \mathrm{Erf}\left(t - a/\sqrt{2}\right) - \mathrm{Erf}\left(t + a/\sqrt{2}\right) \to 2\,\mathrm{Erf}(t)$ exponentially. Thus

$$\sqrt{2}\int_0^\infty e^{-t^2}\left\{2\,\mathrm{Erf}(t) - \mathrm{Erf}\left(t - \frac{a}{\sqrt{2}}\right) - \mathrm{Erf}\left(t + \frac{a}{\sqrt{2}}\right)\right\}\mathrm{d}t$$
$$\to \sqrt{2}\int_0^\infty e^{-t^2} 2\,\mathrm{Erf}(t)\mathrm{d}t = \sqrt{\frac{\pi}{2}},$$

which complete the proof. $\qquad\square$

of Corollary 4.3. For all $\delta > 0$, note that

$$\mathbb{P}(\hat{S}_r(X_i, Y_i) < \hat{S}_r(X_i', Y_i'))$$
$$\geq \mathbb{P}(\hat{S}_r(X_i, Y_i) < \hat{S}_r(X_i', Y_i'), \|\hat{f} - f\|_\infty \leq \delta)$$
$$= \mathbb{P}(|f(X_i) - Y_i + \hat{f}(X_i) - f(X_i)| < |f(X_i') - Y_i' + \hat{f}(X_i') - f(X_i')|, \|\hat{f} - f\|_\infty \leq \delta)$$
$$\geq \mathbb{P}(|f(X_i) - Y_i| < |f(X_i') - Y_i'| - 2\delta, \|\hat{f} - f\|_\infty \leq \delta)$$
$$\geq \mathbb{P}(|f(X_i) - Y_i| < |f(X_i') - Y_i'| - 2\delta) - \mathbb{P}(\|\hat{f} - f\|_\infty > \delta).$$

For the first term in the right hand side of last equation,

$$\mathbb{P}(|f(X_i) - Y_i| < |f(X_i') - Y_i'| - 2\delta) = \iint_{x>y+2\delta} \mathrm{d}G_{X_i'}(x)\mathrm{d}F_{X_i}(y)$$
$$= \int\left[\int_{x>y} - \int_{x\in(y,y+2\delta)}\right] \mathrm{d}G_{X_i'}(x)\mathrm{d}F_{X_i}(y).$$

If $G_{X_i'}$ or F_{X_i} is absolutely continuous, then

$$\int\left[\int_{x>y} - \int_{x\in(y,y+2\delta)}\right] \mathrm{d}G_{X_i'}(x)\mathrm{d}F_{X_i}(y) = o(1) \text{ as } \delta \to 0.$$

Under the assumption $\|\hat{f} - f\|_\infty \xrightarrow{p} 0$, $\mathbb{P}(\|\hat{f} - f\|_\infty > \delta) = o(1)$ for all $\delta > 0$, which complete the proof. $\qquad\square$

of Theorem 4.4. The proof is straight forward since

$$\mathbb{P}(\hat{S}_a(X_i, Y_i) < \hat{S}_a(X_i', Y_i'))$$
$$= \mathbb{P}\left(\frac{\hat{S}_r(X_i', Y_i')}{\hat{u}(X_i') + \hat{\sigma}(X_i')} \leq \frac{\hat{S}_r(X_i, Y_i)}{\hat{u}(X_i) + \hat{\sigma}(X_i)}\right) = \mathbb{P}\left(\hat{S}_r(X_i', Y_i') \leq \frac{\hat{u}(X_i') + \hat{\sigma}(X_i')}{\hat{u}(X_i) + \hat{\sigma}(X_i)}\hat{S}_r(X_i, Y_i)\right)$$
$$\geq \mathbb{P}\left(\hat{S}_r(X_i', Y_i') \leq \hat{S}_r(X_i, Y_i)\right) = \mathbb{P}(\hat{S}_r(X_i, Y_i) < \hat{S}_r(X_i', Y_i')).$$

$\qquad\square$

B. Benchmark Details

For each dataset, we have a given_label representing the noisily-measured response variable typically available in real-world datasets, and a true_label representing a higher fidelity approximation of the true Y value one wishes to measure. The true_label would be unavailable for most datasets in practice and is here solely used for evaluation of different error detection methods. To determine *which* datapoints should be considered truly erroneous in a particular dataset, we conducted a histogram and Gaussian kernel density analysis of true_label - given_label in each dataset, and identified where these deviations became atypically large. Below we list some additional details about each dataset.

Air Quality dataset: This benchmark dataset is a subset of data provided by the UCI repository at `https://archive.ics.uci.edu/ml/datasets/air+quality`. The covariates include information collected from sensors and environmental parameters, such as temperature and humidity, and we aim to predict the CO gas sensor measurement. The true_label is collected using a certified reference analyzer. While the given_label is collected through an Air Quality Chemical Multisensor Device, which is susceptible to sensor drift that can affect the sensors' concentration estimation capabilities.

Metaphor Novelty dataset: This dataset is derived from data provided by `http://hilt.cse.unt.edu/resources.html`. The regression task is to predict metaphor novelty scores given two syntactically related words. We have used FastText word embeddings to calculate vectors for both words available in the dataset. The true_label is collected using expert annotators, and the given_label is the average of all five annotations collected through Amazon Mechanical Turk.

Stanford Politeness Dataset (Stack edition): This dataset is derived from data provided by `https://convokit.cornell.edu/documentation/stack_politeness.html`. The regression task is to predict the level of politeness conveyed by some text, in this case requests from the Stack Exchange website. The given_label is randomly selected from one of five human annotators that rated the politeness of each example, while the median of all five annotators' politeness ratings is considered as the true_label. As covariates for our regression models, we use numerical features obtained by embedding each text example via a pretrained Transformer network from the Sentence Transformers package (Reimers & Gurevych, 2019).

Stanford Politeness Dataset (Wikipedia edition): This dataset is derived from data provided by `https://convokit.cornell.edu/documentation/wiki_politeness.html`. The regression task, feature embeddings, given_label, and true_label are the same as those in the Stanford Politeness Dataset (Stack edition), but here the text is a collection of requests from Wikipedia Talk pages.

qPCR Telomere: This dataset is a subset of the dataset generated through an R script provided by `https://zenodo.org/record/2615735#.ZBpLES-B30p`. It is a simple regression task where independent covariates are taken from a normal distribution, and the true_label is generated by $f(x_i)$. The given_label is defined as true_label + error. While this is technically a simulated dataset, the simulation was specifically aimed to closely mimic data noise encountered in actual qPCR experiments.

C. Simulation Details and Additional Results

Setting 1: Non-parametric Regression with Epistemic/Aleatoric Uncertainty:

- The covariates are i.i.d. from $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{i5}) \in \mathbb{R}^5$ with $x_{ij} \sim 0.1\mathrm{Unif}(-1.5, -0.5) + 0.9\mathrm{Unif}(-0.5, 1.5)$ for $j = 1, \ldots, 5$.

- The responses are generated by sampling from a mixture distribution

$$y_i \sim \frac{1}{2} \cdot N\left(f(x_{i1}) - g(x_{i1}), 0.5 \right) + \frac{1}{2} \cdot N\left(f(x_{i1}) + g(x_{i1}), 0.5 \right),$$

where $f(x) = (x-1)^2(x+1)$, $g(x) = 2\sqrt{x - 0.5}\mathbb{1}(x \geq 0.5)$.

Setting 2: 5-D Linear Regression

- Our second setting is a 5-dimensional underlying linear relationship $y_i = \boldsymbol{\beta}^T \boldsymbol{x}_i + \epsilon_i$, where $\boldsymbol{x}_i \in R^5$ and each coordinate of $\boldsymbol{x}_i$ is generated from $\mathrm{Unif}(-1.5, 1.5)$. The coefficient entries in $\boldsymbol{\beta}$ are set to -1 or 1 with random signs and ϵ_i are sampled i.i.d. from a $N(0, 0.5)$ distribution.

Setting 1 is designed to contain non-uniform epistemic and aleatoric uncertainty, inspired by (Lei & Wasserman, 2014). For each coordinate of $\boldsymbol{x}_i$, 90% of the x_{ij} values are from $\mathrm{Unif}[-0.5, 1.5]$, while only 10% of the x_{ij} values are from $\mathrm{Unif}[-1.5, -0.5]$. Consequently, the epistemic uncertainty for $x_{ij} \in [-1.5, -0.5]$ is larger due to insufficient observations for these x_{ij}. It is important to note that the response y_i depends only on the first coordinate of $\boldsymbol{x}_i$, and the aleatoric

uncertainty for those $x_{i1} \in [0.5, 1.5]$ is larger since $\mathbb{P}(y_i|\boldsymbol{x}_i)$ is bimodal. Figure 1 illustrates the observed y_i with respect to the first coordinate of $\boldsymbol{x}_i$. Setting 2 corresponds to the standard idealized linear regression setting adapted from (Hu & Lei, 2020), where the residual veracity score should perform best as the model is simple and no additional uncertainty is involved.

In both settings, the corrupted data is set to be $y_i^* = y_i + a$, that is, a point mass at a with different corruption strength $a = -3, -2, -1, 1, 2, 3$. In our simulation study, the fraction of corrupted data is set to be 10% for all contaminated datasets. Table 7 and 8 show additional results for our study of conformal methods in Section 5.1, here instead using a LightGBM regressor instead of the Random Forest model presented in the main text.

Table 7. Average FDR and Power for different settings by conformal inference. Each setting is trained on clean labels with Light GBM regressor. The target FDR is 10%.

	Setting 1						Setting 2					
corruption strength	-3	-2	-1	1	2	3	-3	-2	-1	1	2	3
FDR	0.13	0.14	0.28	0.31	0.20	0.14	0.12	0.11	0.29	0.23	0.13	0.12
	0.12	0.15	0.28	0.33	0.16	0.13	0.12	0.10	0.30	0.26	0.13	0.12
	0.13	0.15	0.35	0.37	0.21	0.12	0.12	0.11	0.29	0.28	0.13	0.12
Power	0.40	0.15	0.05	0.02	0.09	0.54	0.94	0.46	0.05	0.07	0.59	0.97
	0.59	0.24	0.04	0.04	0.30	0.68	0.92	0.43	0.06	0.07	0.54	0.92
	0.57	0.18	0.04	0.03	0.25	0.67	0.93	0.48	0.05	0.06	0.56	0.93

Table 8. Average FDR and Power for different settings by conformal inference. Each setting is trained on 10% contaminated data with Light GBM regressor. The target FDR is 10%.

	Setting 1						Setting 2					
corruption strength	-3	-2	-1	1	2	3	-3	-2	-1	1	2	3
FDR	0	0.04	0.30	0.38	0.20	0.01	0	0.05	0.19	0.22	0.01	0.00
	0.02	0.09	0.31	0.43	0.25	0.03	0.00	0.04	0.22	0.18	0.00	0.01
	0.01	0.09	0.33	0.45	0.26	0.03	0.00	0.05	0.19	0.12	0.00	0.01
Power	0.04	0.05	0.02	0.01	0.02	0.06	0.05	0.04	0.03	0.04	0.04	0.05
	0.04	0.05	0.02	0.01	0.03	0.05	0.05	0.05	0.02	0.03	0.05	0.05
	0.04	0.05	0.02	0.01	0.03	0.05	0.05	0.04	0.02	0.03	0.05	0.05

D. Experiment Details

All regression models fit in this paper (including the weighted ensemble) were implemented via the `autogluon` AutoML package (Erickson et al., 2020) which automatically provides good hyperparameter settings and manages the training of each model. When applying the RANSAC algorithm, we used its default settings given in the `scikit-learn` package.

E. Additional Benchmark Comparisons

For more comprehensive evaluation, we additionally compare against a number of other model-agnostic baseline approaches to detect errors in numeric data. Each baseline here produces a veracity score which can be used to rank data by their likelihood of error, as done for our proposed methodology.

We evaluated these alternative scores following the same procedure (same metrics and datasets) from our real dataset benchmark. No data filtering procedure was applied for any of these methods, models were simply fit via K-fold cross-validation to produce out-of-sample predictions for the entire dataset, which were then used to compute veracity scores under each approach. Table 9 below shows that none of these alternative methods are consistently superior to the straightforward residual veracity score studied in our other evaluations.

Here are descriptions of the baseline methods we considered as alternative veracity scores:

Relative Residual. This baseline veracity score is defined as:

$$\exp\left(-\frac{|y - \hat{y}|}{|y| + \epsilon}\right) \tag{2}$$

where $\epsilon = 1e - 6$ is a small constant for numeric stability. The *relative residual* rescales the basic residual by the magnitude of the target variable Y, since values of Y with greater magnitude are often expected to have larger residuals.

Marginal Density. This baseline veracity score is defined as: $\hat{p}(y)$, the (estimated) density of the observed y value under the marginal distribution over Y. Here we use kernel density estimates, and this approach does not consider the feature values X at all. The *marginal density* score is thus just effective to detect Y values that are atypical in the overall dataset (i.e. overall outliers rather than contextual outliers).

Local Outlier Factor. This baseline veracity score attempts to better capture datapoints which have either high residual or low marginal density, since either case may be indicative of an erroneous value. First we form a 2D scatter plot representation of the data in which one axis is the residual: $|\hat{y} - y|$ and the other axis is the original y-value. Intuitively, outliers in this 2D space correspond to the datapoints with abnormal residual or y-value. Thus we employ the *local outlier factor* (LOF) score to quantify outliers in this 2D space, employing this as an alternative veracity score (Breunig et al., 2000).

Outlying Residual Response (OUTRE). This baseline veracity score is similar to the Local Outlier Factor approach above, and identifies outliers in the same 2D space in which each datapoint is represented in terms of its residual and y-value. Instead of the LOF score, here we score outliers via their average distance to the k-nearest neighbors of each datapoint (Kuan & Mueller, 2022b), and use the inverse of these distances as an alternative veracity score.

Discretized. This baseline veracity score is defined by reformulating the regression task as a classification setting, and then applying methods that are effective to detect label errors in classification. More specifically, we discretize the y values in the dataset into 10 bins (defined by partioning the overall range of the target variable). For each bin k, we construct a model-predicted "class" probability for that bin proportionally to: $exp(-|\hat{y} - c_k|)$ where c_k is the center of bin k. After renormalizing these probabilities to sum to 1 over k, this offers a straightforward conversion of regression model outputs to predicted class probabilities if the bins are treated as the possible values in a classification task. Finally, we apply Confident Learning with the self-confidence label quality score to produce a veracity score for each datapoint (Northcutt et al., 2021; Kuan & Mueller, 2022a). This approach uses the given class label (identity of the bin containing each y_i) and model-predicted class probabilities to identify which datapoints are most likely mislabeled.

Table 9. Evaluation of alternative scoring methods across various metrics. The results are calculated as the mean across all datasets and across all models, as discussed in Section B.

scoring method	AUPRC	AUROC	lift_at_100	lift_at_num_errors	spearman_corr
residual	0.42	0.71	5.78	4.96	0.30
relative residual	0.36	0.66	4.61	4.32	0.21
marginal density	0.19	0.57	0.58	1.47	0.05
local outlier factor	0.20	0.64	1.99	2.03	0.23
OUTRE	0.42	0.73	5.76	4.97	0.35
discretised	0.31	0.66	4.85	3.01	0.18