Time Series Data Cleaning: From Anomaly Detection to Anomaly Repairing

Aoqian Zhang
School of Software,
Tsinghua University
zaq13@mails.tsinghua.edu.cn

Shaoxu Song
School of Software,
Tsinghua University
sxsong@tsinghua.edu.cn

Jianmin Wang
School of Software,
Tsinghua University
jimwang@tsinghua.edu.cn

Philip S. Yu
University of Illinois at Chicago
Institute for Data Science,
Tsinghua University
psyu@cs.uic.edu

ABSTRACT

Errors are prevalent in time series data, such as GPS trajectories or sensor readings. Existing methods focus more on anomaly detection but not on repairing the detected anomalies. By simply filtering out the dirty data via anomaly detection, applications could still be unreliable over the incomplete time series. Instead of simply discarding anomalies, we propose to (iteratively) repair them in time series data, by creatively bonding the beauty of temporal nature in anomaly detection with the widely considered minimum change principle in data repairing. Our major contributions include: (1) a novel framework of iterative minimum repairing (IMR) over time series data, (2) explicit analysis on convergence of the proposed iterative minimum repairing, and (3) efficient estimation of parameters in each iteration. Remarkably, with incremental computation, we reduce the complexity of parameter estimation from $O(n)$ to $O(1)$. Experiments on real datasets demonstrate the superiority of our proposal compared to the state-of-the-art approaches. In particular, we show that (the proposed) repairing indeed improves the time series classification application.

1. INTRODUCTION

Time series data are often found with dirty or imprecise values, such as GPS trajectories, sensor reading sequences [15], or even stock prices [16]. For example, the price of SALVEPAR (SY) is misused as the price of SYBASE (SY), both of which share the same notation (SY) in some sources. It is different from the interesting anomaly that actually happens in real life, e.g., the temperatures sudden change from 20C to 10C in one day when cold air rushes in. To distinguish such cases, we propose to employ some labeled truth of dirty observations. (See more detailed examples on dirty data and their labeled truth in Example 1.)

Proceedings of the VLDB Endowment, Vol. 10, No. 10

1.1 Motivation on Anomaly Repairing

Applications, such as pattern mining [17] or classification [26], built upon the dirty time series data are obviously not reliable. Anomaly detection over time series is often applied to filter out the dirty data (see [11] for a comprehensive and structured overview of anomaly detection techniques). That is, the detected anomaly data points are simply discarded as useless noises. Unfortunately, with a large number of consecutive data points eliminated, the applications could be barely performed over the rather incomplete time series.

Recent study [21] shows that repairing dirty values could improve clustering over spatial data. For time series data, we argue that repairing the anomaly can also improve the applications such as time series classification [26]. A repair close to the truth helps greatly the applications.

1.2 Potential Methods for Repairing

A straightforward idea is to directly interpret the predication values in anomaly detection, e.g., by AR [3, 14, 27] or ARX [3, 19], as repairs (see details in Section 2). A data point is considered as anomaly if its (truth) predication significantly differs from (noisy) observation. Unfortunately, noisy/erroneous data are often close to the truth in practice, under the intuition that human or systems always try to minimize their mistakes, e.g., misspellings (John Smith vs. Jhon Smith), typos (555-8145 vs. 555-8195) as illustrated in [1], rounding off (76,821,000 vs. 76M) or unit error (76M vs. 76B) as shown in [16]. Owing to such disagreement, the repairing performance of directly applying anomaly detection techniques is poor, as illustrated in both Example 1 below and experiments in Section 6.

On the other hand, constraint-based repairing SCREEN [22], strictly following the minimum change principle in data repairing [1], heavily relies on a proper constraint of speeds on value changes. The repairing is performed based on two consecutive points, i.e., considering only one historical point, and thus does not sense the temporal nature of errors. As shown in the following Example 1, the speed constraint-based SCREEN is effective in repairing spike errors, but can hardly handle a sequence of consecutive dirty points.

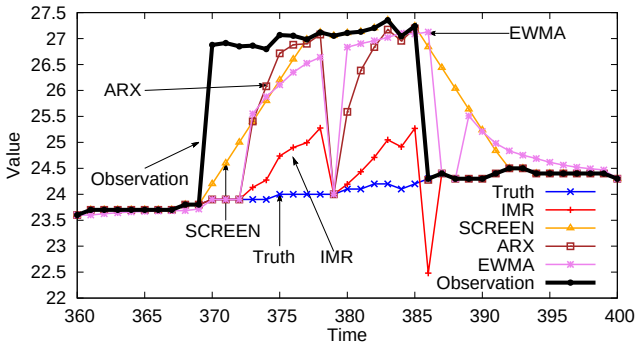

Figure 1: An example segment of sensor readings

In short, the anomaly detection method does not expect the minimized mistakes in practice, whereas the constraint-based repairing is not effective in addressing the temporal nature of errors.

1.3 Intuition of Our Proposal

Since completely automatic data repairing might not work well in repairing time series data (such as SCREEN [22] observed in our experiments in Figure 9), enlightened by the idea of utilizing master data (a single repository of high-quality data) in data repairing [9], we employ some labeled truth of dirty observations to advance the repair. The truth can be obtained either by manual labeling or automatically by more reliable sources. For instance, accurate locations are manually marked in the map by user check-in activities (and utilized to repair the imprecise GPS readings). Moreover, periodical automatic labeling may take place in certain scenarios, e.g., precise equipments report accurate air quality data (as labeled values) in a relatively long sensing period, while crowd and participatory sensing generates unreliable observations in a constant manner [28].

Being aware of both the error nature in anomaly detection and the minimum change principle in data repairing, we propose *iterative minimum repairing* (IMR). The philosophy behind the proposed iterative minimum repairing is that the high confidence repairs in the former iterations could help the latter repairing. Specifically, IMR minimally changes one point a time to obtain the most confident repair only, referring to the minimum change principle in data repairing that human or systems always try to minimize their mistakes. The high confidence repairs, together with the labeled truth of error points, are utilized to learn and enhance the temporal nature of errors in anomaly detection, and thus generate more accurate repair candidates in the latter iterations.

Example 1. *Figure 1 presents an example segment of sensor readings, denoted by black line. Suppose that sensor errors occur in the period from time point 370 to 385, where the observations are shifted from the truth, e.g., owing to granularity mismatch or unit error. To repair the errors, the truth of several observations are labeled, including time points* $\{370, 371, 372, 379, 387\}$.

Existing speed constraint-based cleaning (SCREEN) [22] could not effectively repair such continuous errors in a period (which is indeed also observed in Example 1 in [22]). The reason is that speed constraints, restricting the amount of value changing relative to time difference, can detect sharp deviations such as from time point 369 to 370, but not continuous errors, e.g., in 383 and 384. The exponentially weighted moving average (EWMA) [13] algorithm also hard to find a proper way to clean the trace. These two methods have similar repair trace.

By considering the predication values in anomaly detection as repairs (see more details in Section 2), the result of ARX based repairing is also reported. ARX, considering the errors between truth and observations, shows better repair results than EWMA and SCREEN methods.

Finally, our proposed IMR approach, with both error predication and minimum change considerations, obtains repairs closest to the truth.

The iterative minimum repairing leads to new challenges: (1) whether the repairing process converges; and (2) how to efficiently/incrementally update the parameter of the temporal model over the repaired data after each iteration. Both issues in anomaly repairing are not considered in the anomaly detection studies.

Contributions. Our major contributions in this paper are summarized as follows.

(1) We formalize the anomaly repairing problem, given a time series with some points having labeled truth, in Section 2. The adaption of existing anomaly detection techniques (such as AR and ARX) is introduced for anomaly repairing.

(2) We devise an iterative minimum-change-aware repairing algorithm IMR, in Section 3. Remarkably, we illustrate that the ARX-based approach (in Section 2) is indeed a special case of IMR with static parameter (Proposition 2).

(3) We study the convergence of IMR in various scenarios, in Section 4. In particular, the convergence is explicitly analyzed for the special case of IMR(1) with order $p = 1$, which is sufficient to achieve high repair accuracy in practice (as shown in the experiments in Section 6). We prove that under certain inputs, the converged repair result could be directly calculated without iterative computing (Proposition 8).

(4) We design efficient pruning and incremental computation for parameter estimation in each repair iteration, in Section 5. Rather than performing parameter estimation over all the n points, matrices for parameter estimation could be pruned by simply removing rows with value 0 (Proposition 9). It is also remarkable that the incremental computation among different repair iterations (Proposition 10) could further reduce the complexity of parameter estimation from $O(n)$ to $O(1)$.

(5) Experiments on real datasets with both real and synthetic errors, in Section 6, demonstrate that our IMR method shows significantly better repair performance than the state-of-the-art approaches, including the aforesaid anomaly detection and constraint-based repairing.

Table 1 lists the notations frequently used in this paper. All the proofs can be found in the full technical report (http://ise.thss.tsinghua.edu.cn/sxsong/doc/anomaly.pdf).

2. PRELIMINARIES

This section first introduces the problem of anomaly repairing. We then adapt the existing anomaly detection models for anomaly repairing, i.e., AR without considering labeled data and ARX supporting labeled data.

Table 1: Notations

Symbol	Description
x	observation sequence of n data points
x_i	value of i-th data point in x, a.k.a. $x[i]$
y	truth-labeled/repaired sequence of x
z	distance between x and labeled/repaired y
$y^{(k)}$	sequence y in the k-th iteration
ϕ	parameter of AR(p)/ARX(p) with order p
τ	predefined threshold of convergence
$\boldsymbol{Z}$, $\boldsymbol{V}$	input matrices for parameter estimation
$\boldsymbol{A}$, $\boldsymbol{B}$	intermediate matrices for parameter estimation

The major issues of this simple adaption are: (1) Applying predications with significant difference to the observation as repairs contradicts to the minimum change principle in data repairing [1], as discussed in the introduction. (2) A static parameter (ϕ in Equations 1 and 3 below) needs to be preset, e.g., estimated from the dirty data during the initialization.

2.1 Problem Statement

Consider a time series of n observations, $x = x[1], \ldots, x[n]$, where each $x[i]$ is the value of the i-th data point. For brevity, we write $x[i]$ as x_i.

Let y denote the labeled/repaired sequence of x. Each y_i is either the labeled truth or the repaired value of x_i.

Given a time series x and a partially labeled subset y of x, the repairing problem is to determine the repairs y_i of x_i that are not labeled in y.

Example 2 (Observation x, partially labeled y, and fully repaired y)**.** *Consider* $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *with twelve data points of observations in Figure 2, where shift (up) errors occur on four points* $x_2, \ldots, x_5$*. Suppose that five points are labeled with truth, i.e., the partially labeled* y*. By repairing (using the methods presented below), we propose to obtain a fully repaired* y*, e.g.,* $y = \{6, 5.6, 5.4, 5.2, 5.4, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *as shown in Figure 2. In the repaired* y*,* x_4 *and* x_5 *are changed from 8.3 and 7.7 to 5.2 and 5.39, respectively. The labeled* y_2 *and* y_3 *will not be modified in the repair result.*

Figure 2 also presents another repair y' *by the approach of connecting the dots with the labeled values, i.e., linear interpolation [25]. As shown in Figure 2 and also indicated in [22], the major issue of this (smoothing) approach is the serious damage of almost all the (unlabeled) data points, such as* $y'_7 \ldots y'_{11}$*, which are originally correct and should not be modified. In contrast, our proposed method repairs* y_4 *and* y_5 *while leaving* $y_7 \ldots y_{11}$ *unchanged.*

2.2 AR Model

Intuitively, anomaly detection techniques could be adapted to anomaly repairing. For instance, we consider the AR (autoregression) model [14, 27] as follows:

$$x'_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t \tag{1}$$

where x'_t is the prediction of x_t, p is the order, ϕ_i is the parameter of the model, c is a constant defined by $c = \mu(1 - \sum_{i=1}^{p} \phi_i)$, μ is mean value of the process and ϵ_t is white noise (usually Gaussian white noise [3], a normal random variable generated according to the Gaussian distribution with mean $\mu = 0$ and variance σ^2; in other words, $c = 0$).

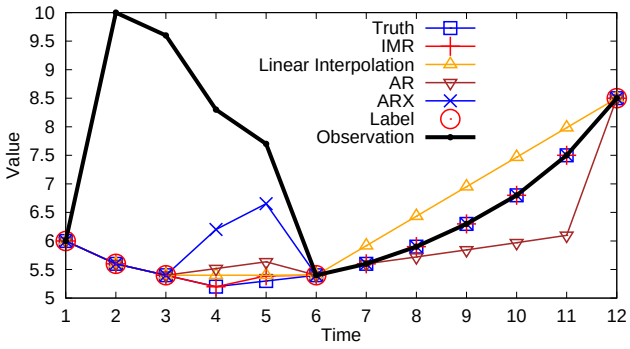

Figure 2: Example of observations and repairs

If x'_t significantly differs from the original observation x_t, having $|x'_t - x_t| > \tau$ where τ is a predefined threshold, this predication is accepted $x_t = x'_t$, a.k.a. a repair. The intuition behind is that a farther distance indicates the higher probability of being an outlier. The threshold τ can be decided by observing the statistical distribution of distances between x'_t and x_t, using the prediction interval [14, 12].

The AR-based repairing procedure is thus: (1) replace x_t by y_t if it is labeled, (2) learn parameter ϕ of AR(p) from x, and (3) fill all unlabeled y_t by AR(p) over x, having

$$y_t = \begin{cases} x'_t & \text{if } y_t \text{ is unlabeled and } |x'_t - x_t| > \tau \\ x_t & \text{otherwise} \end{cases} \tag{2}$$

Example 3 (Example 2 continued)**.** *Consider again* $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *in Figure 2. For simplicity, we use AR(1) with order* $p = 1$ *and* $c = 0$*, i.e.,* $x'_t = \phi_1 x_{t-1}$*. By ordinary least square [20], we estimate the parameter* ϕ *from* x*, having* $\phi_1 = 1.022$*. Let* $\tau = 0.1$*.* y_1 *is labeled with truth. Referring to Equation 2, it outputs unchanged* $y_1 = 6$*. Similarly for* y_2 *and* y_3*. We have* $x'_4 = \phi_1 x_1 = 1.022 * 5.4 = 5.52$*. Since* $|5.52 - 8.3| = 2.78 > 0.1$*,* x'_4 *is accepted as new* x_4*.*

Similarly, we have $x'_5 = \phi_1 x_4 = 1.022 * 5.52 = 5.64$*. Referring to* $|5.64 - 7.7| = 2.06 > 0.1$*, the prediction is accepted. So on and so forth, we obtain the final repaired result* $y = \{6, 5.6, 5.4, 5.52, 5.64, 5.4, 5.6, 5.72, 5.84, 5.97, 6.10, 8.5\}$*. Its RMS error is 0.51 (see Section 6 for RMS definition).*

2.3 ARX Model

In order to utilize the labeled y, we consider the ARX model (autoregressive model with exogenous inputs) [19]

$$y'_t = x_t + \sum_{i=1}^{p} \phi_i (y_{t-i} - x_{t-i}) + \epsilon_t \tag{3}$$

where y'_t is the possible repair of x_t, and others are the same to the aforesaid AR model. As shown in Equation 3, not only the preceding observations x_{t-i} will affect the determination of y'_t, but also the previously labeled/repaired y_{t-i}.

The ARX-based repairing procedure is thus: (1) learn parameter ϕ of ARX(p) from x and partially labeled y, and

(2) fill all unlabeled y_t by ARX(p), similar to Equation 2 by replacing x'_t with y'_t.

Example 4 (Example 2 continued). *Consider again* $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *and the partially labeled* y *over five points, in Figure 2. For simplicity, we also use ARX(1) with order* $p = 1$*, i.e.,* $y'_t = x_t + \phi_1(y_{t-1} - x_{t-1})$*. Similar to AR in Example 3, we estimate the parameter* ϕ *by ordinary least square [20], having* $\phi_1 = 0.5$*. Let* $\tau = 0.1$*. Again, the labeled* $y_3 = 5.4$ *is not modified. For the fourth point, we have* $y'_4 = 8.3 + 0.5 * (5.4 - 9.6) = 6.2$*. Since* $|6.2 - 8.3| = 2.1 > 0.1$*, we assign* $y_4 = y'_4 = 6.2$*. Finally, the repair result by ARX is* $y = \{6, 5.6, 5.4, 6.20, 6.65, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *with RMS error 0.49, lower than that of AR in Example 3.*

We consider ARX model since it can capture the difference between the observed errors and labeled truths, while other methods such as AR and SCREEN ignore (cannot utilize) such differences. By modeling such differences between errors and truths rather than original values, both ARX and our proposed IMR may not deal with Spike errors (i.e., with # consecutive errors = 1 in Figure 9) as good as SCREEN [22]. Nevertheless, with iterative repairing, IMR always shows significantly better results in addressing a large number of consecutive errors (see Figure 9 as well). ARX cannot address well such consecutive errors either, since it applies only significant changes which is contradict to the minimum change principle in data repairing as discussed in Section 1.2 in the Introduction.

3. REPAIR ALGORITHM

Unlike the existing anomaly detection model that may over-change the data via one-pass repairing (as discussed in the introduction, illustrated in Figure 2, and observed in experiments in Section 6), we propose to progressively repair the data, being aware of both the error nature in anomaly detection and the minimum change principle in data repairing, so that the high confidence repairs in the former iterations could help the repairing in the latter steps.

3.1 Iterative Repairing

Let $y^{(k)}$ denote the sequence y in the k-th iteration, where $y^{(0)}$ is the partially labeled time series in the input. Since $y^{(0)}$ is incomplete (partially labeled), to initialize, we assign $y_t^{(0)} = x_t$ if $y_t^{(0)}$ is not labeled. Recall that the labeled values should not be repaired, i.e., $y_t^{(k)} = y_t^{(0)}$ if $y_t^{(0)}$ is labeled.

Algorithm 1 presents the iterative minimum repairing procedure, IMR(p), whose inputs are the observation time series x and partially labeled $y^{(0)}$. It outputs $y^{(k)}$ with all the labeled $y_t^{(0)}$ unchanged and unlabeled $y_t^{(0)}$ repaired.

The major steps include:

(S1) Parameter estimation, in Line 2, learns the parameter of ARX(p) in the k-th iteration, denoted by $\phi^{(k)}$, from x and the current $y^{(k)}$.

(S2) Repair candidate generation, in Line 3, computes the possible repairs $\hat{y}^{(k)}$, according to ARX(p) w.r.t. x, $y^{(k)}$ and $\phi^{(k)}$.

(S3) Repair evaluation, in Line 4, determines one of the repairs to accept, $y_t^{(k+1)} = \hat{y}_t^{(k)}$, referring to the minimum change principle in data repairing [1].

As shown in Line 5, the procedure repeats, until the repair converges, e.g., having

$$|y_j^{(k)} - y_j^{(k+1)}| \leq \tau, j = 1, \ldots, n. \tag{4}$$

where τ a threshold of convergence, or a maximum number of iterations is reached. Setting *max-num-iterations* is a remedy to avoid waiting for convergence in practice (see Section 6.1.4 for discussion and evaluation).

Algorithm 1: IMR(p)

Input: time series x and partially labeled $y^{(0)}$
Output: $y^{(k)}$ with all the labeled $y_i^{(0)}$ unchanged and unlabeled $y_j^{(0)}$ repaired

```
1 for k ← 0 to max-num-iterations do
2     φ^(k) ← Estimate(x, y^(k));
3     ŷ^(k) ← Candidate(x, y^(k), φ^(k));
4     y^(k+1) ← Evaluate(x, y^(k), ŷ^(k));
5     if Converge(y^(k), y^(k+1)) then
6         break;
7     k ← k + 1;
8 return y^(k)
```

Example 5 (Algorithm overview, Example 2 continued). *Consider again* $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *in Figure 2. According to five labeled data points, we assign* $y^{(0)} = \{6, 5.6, 5.4, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$*, where the unlabeled points are initialized by* $y_t^{(0)} = x_t$*, e.g.,* $y_4^{(0)} = x_4 = 8.3$.

In each iteration, the IMR algorithm (1) learns the parameter, e.g., $\phi_1^{(0)} = 0.5$ *for* $p = 1$*; (2) generates candidates for repairing, such as* $\hat{y}^{(0)} = \{-, -, -, 6.2, 7.7, -, 5.6, 5.9, 6.3, 6.8, 7.5, -\}$*; and (3) selects one repair to conduct, and form the new sequence, say* $y^{(1)} = \{6, 5.6, 5.4, 6.2, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$.

The procedure repeats until converging. The final output is $y^{(7)} = \{6, 5.6, 5.4, 5.20, 5.39, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *with RMS error 0.03. Details on each step are presented in the following examples.*

3.2 Parameter Estimation

The parameter estimation step S1 (in Line 2 in Algorithm 1) estimates the parameter $\phi^{(k)}$ for ARX(p), given $x, y^{(k)}$. Existing methods such as Ordinary Least Square [20] or Yule-Walker Equations [7] can be directly employed. For instance, by Ordinary Least Square, we have

$$\phi^{(k)} = ((\boldsymbol{Z}^{(k)})' \boldsymbol{Z}^{(k)})^{-1} (\boldsymbol{Z}^{(k)})' \boldsymbol{V}^{(k)} \tag{5}$$

where

$$\boldsymbol{V}^{(k)} = \begin{pmatrix} y_{p+1}^{(k)} - x_{p+1} \\ y_{p+2}^{(k)} - x_{p+2} \\ \vdots \\ y_n^{(k)} - x_n \end{pmatrix}, \qquad \phi^{(k)} = \begin{pmatrix} \phi_1^{(k)} \\ \phi_2^{(k)} \\ \vdots \\ \phi_p^{(k)} \end{pmatrix},$$

$$\boldsymbol{Z}^{(k)} = \begin{pmatrix} y_p^{(k)} - x_p & y_{p-1}^{(k)} - x_{p-1} & \ldots & y_1^{(k)} - x_1 \\ y_{p+1}^{(k)} - x_{p+1} & y_p^{(k)} - x_p & \ldots & y_2^{(k)} - x_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-1}^{(k)} - x_{n-1} & y_{n-2}^{(k)} - x_{n-2} & \ldots & y_{n-p}^{(k)} - x_{n-p} \end{pmatrix}.$$

Example 6 (Parameter estimation on $y^{(0)}$, Example 5 continued). *Consider* $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *and* $y^{(0)} = \{6, 5.6, 5.4, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$. *Given order* $p = 1$, *we have* $\boldsymbol{V}^{(0)} = \{-4.4, -4.2, 0, 0, 0, 0, 0, 0, 0, 0\}'$ *with 11 rows and 1 column, and* $\boldsymbol{Z}^{(0)} = \{0, -4.4, -4.2, 0, 0, 0, 0, 0, 0, 0, 0\}'$ *with 11 rows and 1 column. Referring to Equation 5, the parameter is estimated by*

$$\phi_1^{(0)} = \frac{(-4.4) * (-4.2)}{(-4.4)^2 + (-4.2)^2} = 0.5.$$

Owing to the iterative repairing, online incremental parameter estimation is necessary, which is not studied in the previous studies (see our approach in Section 5).

3.3 Candidate Generation

The repair candidate generation step S2 (in Line 3 in Algorithm 1) employs ARX(p) to infer the candidate repair $\hat{y}^{(k)} = \phi^{(k)} \cdot (y^{(k)} - x) + x$, referring to the estimated parameter $\phi^{(k)}$. More specifically, for each point t, $\hat{y}_t^{(k)}$ is given by

$$\hat{y}_t^{(k)} = \sum_{i=1}^{p} \phi_i^{(k)} (y_{t-i}^{(k)} - x_{t-i}) + x_t \tag{6}$$

according to $y_{t-1}^{(k)}, \ldots, y_{t-p}^{(k)}$. We note that only candidates with $|\hat{y}_t^{(k)} - y_t^{(k)}| > \tau$ need to be considered referring to the convergence condition in Equation 4.

Example 7 (Repair candidate $\hat{y}^{(0)}$, Example 6 continued). *Consider the parameter* $\phi_1^{(0)} = 0.5$ *estimated in Example 6. Let threshold* $\tau = 0.1$. *Referring to Equation 6, we have* $\hat{y}_4^{(0)} = 0.5 * (5.4 - 9.6) + 8.3 = 6.2$ *with* $|\hat{y}_4^{(0)} - y_4^{(0)}| = |6.2 - 8.3| = 2.1 > 0.1$, *and* $\hat{y}_5^{(0)} = 0.5 * (8.3 - 8.3) + 7.7 = 7.7$ *with* $|7.7 - 7.7| = 0 < 0.1$. *The repair candidates are* $\hat{y}^{(0)} = \{+, +, +, 6.2, -, +, -, -, -, -, -+\}$ *where '+' corresponds to the labeled points and '−' denotes no candidates. That is, we need to consider only one candidate* $\hat{y}_4^{(0)}$ *for repairing.*

3.4 Repair Evaluation

The repair evaluation step S3 (in Line 4 in Algorithm 1) selects one repair to accept, i.e., assigning $y_t^{(k+1)} = \hat{y}_t^{(k)}$ the aforesaid generated repair candidate. Following the minimum change principle in data repairing [1], the repair that minimally differs from its original input is preferred with higher confidence. The repaired result in each iteration is:

$$y_t^{(k+1)} = \begin{cases} \hat{y}_t^{(k)} & \text{if } t = \arg\min_i |\hat{y}_i^{(k)} - x_i| \\ y_t^{(k)} & \text{otherwise} \end{cases}. \tag{7}$$

Remarkably, only one data point with the minimum change (most confident) is repaired in each iteration, which is more efficient than the NP-hard problem of minimizing the overall changes w.r.t. integrity constraints [1].

Example 8 (Minimum repair $\hat{y}_t^{(1)}$, Example 7 continued). *Since there is only one repair candidate obtained in Example 7, i.e.,* $\hat{y}_4^{(0)} = 6.2$, *it is the minimum repair (among all candidates). The sequence after the first iteration becomes* $y^{(1)} = \{6, 5.6, 5.4, 6.2, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$.

Note that the minimum change principle [1] in data repairing is based on the intuition that human or systems always *try to* minimize their mistakes. However, it is not guaranteed that the minimum change repair always corresponds to the true value. Therefore, similar to other minimum change-based data repairing studies [1, 8], the accuracy of the final results is unlikely to have theoretical guarantees, since there is no constraint on how far the errors may diverge from the truth. For this reason, we can only evaluate the correctness of the proposed repair by comparing to the ground truth in experiments, similar to other data repairing studies [1, 8] as well. Nevertheless, we can show that the efficient pruning and incremental computation are safe (Propositions 9 and 10), i.e., the accuracy of the final results with efficient computing is theoretically guaranteed to be the same as the results of original IMR without pruning and incremental computation.

4. CONVERGENCE ANALYSIS

In this section, we analyze the convergence of iterative repairing, i.e., $\lim_{k \to +\infty} \sum_{i=1}^{n} \left(y_i^{(k+1)} - y_i^{(k)} \right) = 0$, which is essential to the termination of Algorithm 1. While the general convergence problem is still open, we study the convergence of the proposed method in certain special cases in this section for two reasons: (1) we illustrate that the ARX-based approach is indeed a special case of the proposed IMR with static parameter (Proposition 2) in Section 4.1; (2) we identify another special case with convergence guarantee in Section 4.2, which enables online repairing over streaming data without iteration (also see Sections 4.2.2 and 6.2.2 for more details and experiments).

4.1 Static Parameter

We study this special case in order to illustrate the relationship between our proposed IMR and the existing ARX. Let us first analyze the convergence of IMR (Proposition 1) and then illustrate their equivalence in certain case (Proposition 2).

Rather than dynamically updating the parameter $\phi^{(k)}$ in each iteration, in Line 2 in Algorithm 1, a special case is to specify a static parameter, $\phi^{(k)} = \phi^{(0)}$, for all the iterations.

Proposition 1. *With a static parameter* $\phi^{(k)} = \phi, \forall k$, *the repair result converges, i.e.,*

$$\lim_{k \to +\infty} \sum_{i=1}^{n} \left(y_i^{(k+1)} - y_i^{(k)} \right) = 0.$$

Special Case of IMR(1)

We show in the following that for $p = 1$, the ARX(p)-based repairing in Section 2.3 is a special case of our proposed IMR(p) with static parameter $\phi^{(0)}$. This equivalence demonstrates the rationale of our proposal.

Proposition 2. *For IMR(1) with static parameter* $\phi^{(k)} = \phi, \forall k$, *Algorithm 1 is equivalent to ARX(1)-based repairing.*

4.2 Converged Parameter

We now consider the dynamically updated parameter $\phi^{(k)}$ in each iteration, in Line 2 in Algorithm 1. As shown in the following Proposition 3, if the dynamic parameter converges, the repair converges as well. This converged parameter case is interesting, since the corresponding converged repair result could be directly calculated without iterative computing in certain cases as illustrated below.

Proposition 3. *If the parameter converges,* $\lim_{k\to+\infty}\phi^{(k)} = \phi$*, then the repair also converges*

$$\lim_{k\to+\infty}\sum_{i=1}^{n}\left(y_i^{(k+1)} - y_i^{(k)}\right) = 0.$$

4.2.1 Special Case of IMR(1)

Again, we consider the special case of IMR(1) with order $p = 1$. To show how the repair results could be directly computed without iterations, we first illustrate that any $y_t^{(k)}$ generated during Algorithm 1 can be represented as follows, a.k.a. provenance of $y_t^{(k)}$.

Lemma 4. *For IMR(1), we can represent each* $y_t^{(k)}$ *by*

$$y_t^{(k)} = \phi_1^{(k_s)}\phi_1^{(k_{s-1})}\dots\phi_1^{(k_1)}(y_{t-s}^{(0)} - x_{t-s}) + x_t,$$

where $0 < k_1 < \dots < k_{s-1} < k_s < k$ *denote iteration numbers,* $y_{t-s}^{(0)}$ *is labeled truth, and time points* $t-s+1,\dots,t$ *are not labeled.*

We denote the labeled points in $y^{(0)}$ by multiple (say m) segments. Let $s(j)$ and $e(j)$ denote the start and end point of the j-th labeled segment, $j = 1,\dots,m$. For instance, there are 3 segments of labeled data points in Figure 2, having $s(1) = 1, e(1) = 2, s(2) = 4, e(2) = 5, s(3) = 9, e(3) = 9$.

Let $z_i = y_i^{(0)} - x_i$ for all labeled points i. (We set $z_{e(0)} = 0$ over undefined segment 0.)

Proposition 5. *For IMR(1), if the parameter converges, having* $\lim_{k\to+\infty}\phi_1^{(k)} = \phi_1$*, then the converged repair result can be directly given by*

$$\lim_{k\to+\infty} y_i^{(k)} = y_i,$$

where

$$y_i = \begin{cases} y_i^{(0)} & \text{if } i \in [s(j), e(j)] \\ \phi_1^{i-e(j)}(y_{e(j)}^{(0)} - x_{e(j)}) + x_i & \text{if } i \in (e(j), s(j+1)) \end{cases} \tag{8}$$

and the converged parameter ϕ_1 *is a solution to*

$$\phi_1 = \frac{\sum_{j=1}^{m}\left(\phi_1^{s(j)-1-e(j-1)} z_{e(j-1)} z_{s(j)} + \sum_{i=s(j)}^{e(j)-1} z_i z_{i+1}\right)}{\sum_{j=1}^{m}\left((\phi_1^{s(j)-1-e(j-1)} z_{e(j-1)})^2 + \sum_{i=s(j)}^{e(j)-1}(z_i)^2\right)}. \tag{9}$$

4.2.2 IMR(1) with One Labeled Segment

We consider the case that only one segment with length ℓ is labeled at the beginning of $y^{(0)}$, i.e., $y_1^{(0)}, y_2^{(0)}, \dots, y_\ell^{(0)}$ are labeled. In this special case, the converged parameter and repair result can be directly calculated without iterating, and most importantly it enables efficient online computation, by interpreting all the historical data as one segment labeled (see Section 6.2.2 for details and evaluation). Remarkably, no threshold needs to be set in this case.

The idea is: (1) We first show in Lemma 6 that under certain inputs, the estimated parameter in each iteration is indeed bounded; (2) Proposition 7 then illustrates that the bounded parameter leads to converged parameter; (3) Finally, analogous to Proposition 5, given the converged parameter, we directly calculate the converged repair without iterative computing in Proposition 8.

Lemma 6. *For IMR(1) with first* ℓ *data points labeled in* $y^{(0)}$*. If the input satisfies* $\left|\sum_{t=1}^{\ell-1} z_t^{(0)} z_{t+1}^{(0)}\right| < \sum_{t=1}^{\ell-1} z_t^{(0)} z_t^{(0)}$*, i.e.,*

$$\left|\sum_{t=1}^{\ell-1}(y_t^{(0)} - x_t)(y_{t+1}^{(0)} - x_{t+1})\right| < \sum_{t=1}^{\ell-1}(y_t^{(0)} - x_t)^2,$$

then we have $|\phi_1^{(k)}| < 1$ *in the iterations* $k, 0 \le k \le n - \ell$*.*

The following conclusion illustrates that with a bounded parameter in each iteration, the parameter converges.

Proposition 7. *For IMR(1) with first* ℓ *data points labeled in* $y^{(0)}$*, if* $|\phi_1^{(k)}| < 1$ *in the iterations* $k, 0 \le k \le n - \ell$*, then the parameter converges, i.e.,*

$$\lim_{k\to+\infty}\phi_1^{(k)} = \phi_1,$$

It is worth noting that the condition $|\phi_1^{(k)}| < 1$ in Proposition 7 for the parameter to converge could be commonly observed in real data. First, referring to [5], most time series in practice are stationary, which is guaranteed to have $|\phi_1^{(k)}| < 1$ for $p = 1$. Moreover, for non-stationary cases, a typical processing way is to transform it to stationary via differencing [5].

Proposition 8. *For IMR(1) with first* ℓ *data points labeled in* $y^{(0)}$*, if the parameter converges, having* $\lim_{k\to+\infty}\phi_1^{(k)} = \phi_1$*, then the converged repair result is*

$$\lim_{k\to+\infty} y_t^{(k)} = y_t = \begin{cases} y_i^{(0)} & \text{if } i \in [1, \ell] \\ \phi_1^{i-\ell}(y_\ell^{(0)} - x_\ell) + x_i & \text{if } i > \ell \end{cases} \tag{10}$$

where the converge parameter can be directly calculated by

$$\phi_1 = \frac{(y_1^{(0)} - x_1)(y_2^{(0)} - x_2) + \dots + (y_{\ell-1}^{(0)} - x_{\ell-1})(y_\ell^{(0)} - x_\ell)}{(y_1^{(0)} - x_1)^2 + \dots + (y_{\ell-1}^{(0)} - x_{\ell-1})^2}. \tag{11}$$

It is not surprising that the converged parameter ϕ_1 in Equation 11 in Proposition 8 is exactly the solution of Equation 9 in Proposition 5 with $m = 1$ for the special case of one labeled segment.

5. EFFICIENT PARAMETER ESTIMATION

Among the three major steps in Algorithm 1, while the repair candidate generation and evaluation (in Sections 3.3 and 3.4) are inevitable for the minimum repair, we show in this section that the costly parameter $\phi^{(k)}$ estimation (in Section 3.2) is optimizable in our iterative repairing scenario. First, we identify that the matrices $\boldsymbol{Z}^{(k)}$, $\boldsymbol{V}^{(k)}$ for parameter estimation could be pruned by simply removing rows with value 0, in Section 5.1. Moreover, incremental computation could be designed such that the complexity of parameter estimation is reduced from $O(n)$ to $O(1)$, in Section 5.2.

5.1 Matrix Pruning

Intuition. Recall that when estimating the parameter $\phi^{(k)}$ in Equation 5, we need to consider two large matrices $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$ with sizes $(n-p)\times p$ and $(n-p)\times 1$, respectively. The value $y_i^{(k)} - x_i$ in $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$ denotes the

difference between the labeled/repaired value $y_i^{(k)}$ and the input value x_i of point i. In practice, the labeled data is often limited, while the repaired data should not be significantly changed referring to the minimum change principle of repairing. That is, most values in $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$ equal to 0. We show (in Proposition 9 below) that the sparse matrices could be pruned by removing rows with value 0.

Let $z_i^{(k)} = y_i^{(k)} - x_i$ for simplicity. We rewrite $\boldsymbol{V}^{(k)}, \boldsymbol{Z}^{(k)}$ in Equation 5 for parameter estimation,

$$\boldsymbol{V}^{(k)} = \begin{pmatrix} z_{p+1}^{(k)} \\ z_{p+2}^{(k)} \\ \vdots \\ z_n^{(k)} \end{pmatrix}, \quad \boldsymbol{Z}^{(k)} = \begin{pmatrix} z_p^{(k)} & \cdots & z_1^{(k)} \\ z_{p+1}^{(k)} & \cdots & z_2^{(k)} \\ \vdots & \ddots & \vdots \\ z_{n-1}^{(k)} & \cdots & z_{n-p}^{(k)} \end{pmatrix}.$$

The following conclusion states that the same parameter $\phi^{(k)}$ could still be computed by Equation 5, after removing the rows in $\boldsymbol{Z}^{(k)}$, whose values equal to 0, and the corresponding rows in $\boldsymbol{V}^{(k)}$.

Proposition 9. *For any row in $\boldsymbol{Z}^{(k)}$, denoted by $\boldsymbol{Z}_r^{(k)}$, if the entire row are all with value 0, i.e., $z_{r+p-1}^{(k)} = z_{r+p-2}^{(k)} = \ldots = z_r^{(k)} = 0$, then it is safe to remove the row $\boldsymbol{Z}_r^{(k)}$ and the corresponding row $\boldsymbol{V}_r^{(k)} = \left(z_{p+r}^{(k)}\right)$ from matrices $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$, respectively, which still compute that the same $\phi^{(k)}$.*

Example 9 (Parameter estimation with matrix pruning, Example 6 continued)**.** *Consider again $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ and $y^{(0)} = \{6, 5.6, 5.4, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ in Example 6. Given order $p = 1$, we have $\boldsymbol{V}^{(0)} = \{-4.4, -4.2, 0, 0, 0, 0, 0, 0, 0, 0, 0\}'$ with size 11×1, and $\boldsymbol{Z}^{(0)} = \{0, -4.4, -4.2, 0, 0, 0, 0, 0, 0, 0, 0\}'$ with size 11×1. Referring to Proposition 9, all the rows except the fourth and fifth rows in $\boldsymbol{Z}^{(0)}$ and the corresponding rows in $\boldsymbol{V}^{(0)}$ can be removed. After matrix pruning, we have $\boldsymbol{V}^{(0)} = \{-4.2, 0\}'$ with size 2×1, and $\boldsymbol{Z}^{(0)} = \{-4.4, -4.2\}'$ with size 2×1. Referring to Equation 5, the parameter is estimated by*

$$\phi_1^{(0)} = \frac{(-4.4) * (-4.2)}{(-4.4)^2 + (-4.2)^2} = 0.5.$$

The computed parameter is the same as in Example 6.

5.2 Incremental Computation

Intuition. In each iteration in Algorithm 1, the parameter $\phi^{(k)}$ is estimated by Equation 5 w.r.t. $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$ over all the n points. However, referring to the minimum change principle in Section 3.4, there is only one point, say r, which is changed in each iteration, having $y_r^{(k)} \neq y_r^{(k-1)}$. That is, most values in $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$ are the same to $\boldsymbol{Z}^{(k-1)}$ and $\boldsymbol{V}^{(k-1)}$. We show (in Proposition 10 below) that $\phi^{(k)}$ can be incrementally computed by considering only the changed values instead of the entire $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$. The time complexity of parameter estimation in each iteration is thus reduced from linear time to constant time.

5.2.1 Recursive Formula

To enable the incremental computation, we rewrite Equation 5 for parameter estimation as follows,

$$\phi^{(k)} = (\boldsymbol{A}^{(k)})^{-1}\boldsymbol{B}^{(k)} \tag{12}$$

$$\boldsymbol{A}^{(k)} = (\boldsymbol{Z}^{(k)})'\boldsymbol{Z}^{(k)} = \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1p}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} & \cdots & a_{2p}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}^{(k)} & a_{p2}^{(k)} & \cdots & a_{pp}^{(k)} \end{pmatrix} \tag{13}$$

$$\boldsymbol{B}^{(k)} = (\boldsymbol{Z}^{(k)})'\boldsymbol{V}^{(k)} = \begin{pmatrix} b_1^{(k)} \\ b_2^{(k)} \\ \vdots \\ b_p^{(k)} \end{pmatrix} \tag{14}$$

where

$$a_{ii}^{(k)} = \sum_{l=p+1-i}^{n-i} z_l^{(k)} z_l^{(k)}, \qquad 1 \leq i \leq p \tag{15}$$

$$a_{ij}^{(k)} = a_{ji}^{(k)} = \sum_{l=p+1-i}^{n-i} z_l^{(k)} z_{l-j+i}^{(k)}, \quad 1 \leq i \leq p, 1 \leq j \leq p, j > i \tag{16}$$

$$b_i^{(k)} = \sum_{l=p+1}^{n} z_l^{(k)} z_{l-i}^{(k)}, \qquad 1 \leq i \leq p \tag{17}$$

The following conclusion illustrates that $\boldsymbol{A}^{(k)}$ and $\boldsymbol{B}^{(k)}$, with sizes $p \times p$ and $p \times 1$, respectively, could be recursively computed from $\boldsymbol{A}^{(k-1)}$ and $\boldsymbol{B}^{(k-1)}$. The pre-defined p in the AR(p)/ARX(p) model is a fixed value in the algorithm and has $p \ll n$. The computational cost of Equation 12 over $\boldsymbol{A}^{(k)}$ and $\boldsymbol{B}^{(k)}$ with sizes on p will be significantly lower than Equation 5 w.r.t. $\boldsymbol{Z}^{(k)}$ and $\boldsymbol{V}^{(k)}$ over all the n points.

Proposition 10. *Let r be the changed point in the current repair iteration, having $y_r^{(k)} \neq y_r^{(k-1)}$ or equivalently $z_r^{(k)} \neq z_r^{(k-1)}$. $\boldsymbol{A}^{(k)}, \boldsymbol{B}^{(k)}$ could be recursively computed from $\boldsymbol{A}^{(k-1)}, \boldsymbol{B}^{(k-1)}$.*

That is, for $1 \leq i \leq p$, we have

$$a_{ii}^{(k)} = a_{ii}^{(k-1)} + \begin{cases} 0 & \text{if } r < p+1-i \text{ or } r > n-i \\ z_r^{(k)} z_r^{(k)} - z_r^{(k-1)} z_r^{(k-1)} & \text{if } p+1-i \leq r \leq n-i \end{cases} \tag{18}$$

For $1 \leq i \leq p, 1 \leq j \leq p, i < j$, we have

$$a_{ij}^{(k)} = a_{ji}^{(k)} = a_{ij}^{(k-1)} + (z_r^{(k)} - z_r^{(k-1)})\times \begin{cases} 0 & \text{if } r < p+1-j \text{ or } r > n-i \\ z_{r+j-i}^{(k-1)} & \text{if } p+1-j \leq r < p+1-i \\ z_{r-j+i}^{(k-1)} & \text{if } n-j < r \leq n-i \\ (z_{r+j-i}^{(k-1)} + z_{r-j+i}^{(k-1)}) & \text{if } p+1-i \leq r \leq n-j \end{cases} \tag{19}$$

For $1 \leq i \leq p$, we have

$$b_i^{(k)} = b_i^{(k-1)} + (z_r^{(k)} - z_r^{(k-1)})\times \begin{cases} 0 & \text{if } r < p+1-i \\ z_{r+i}^{(k-1)} & \text{if } p+1-i \leq r < p+1 \\ z_{r-i}^{(k-1)} & \text{if } r > n-i \\ (z_{r+i}^{(k-1)} + z_{r-i}^{(k-1)}) & \text{if } p+1 \leq r \leq n-i \end{cases} \tag{20}$$

5.2.2 Recursive Algorithm

Algorithm 2 shows the procedure of incrementally estimating the parameter $\phi^{(k)}$. For $k = 0$ in the first iteration, computing $\boldsymbol{A}^{(0)}, \boldsymbol{B}^{(0)}$ by Equations 13 and 14 w.r.t. matrices $\boldsymbol{Z}^{(0)}$, $\boldsymbol{V}^{(0)}$, however, is inevitable. Nevertheless, the efficient pruning of rows with value 0 in Proposition 9 can be applied as presented in Line 2 in Algorithm 2.

Algorithm 2: Estimate$(x, y^{(k)})$

Input: time series x and intermediate repair result $y^{(k)}$
Output: estimated parameter $\phi^{(k)}$

1 **if** $k = 0$ **then**
2 Initialize $\boldsymbol{A}^{(0)}, \boldsymbol{B}^{(0)}$ in Equations 13 and 14 by using the pruned matrices $\boldsymbol{Z}^{(0)}$, $\boldsymbol{V}^{(0)}$ in Proposition 9 ;
3 **else**
4 Let r be the changed point in the k-th iteration having $y_r^{(k)} \neq y_r^{(k-1)}$;
5 Compute $\boldsymbol{A}^{(k)}, \boldsymbol{B}^{(k)}$ according to $\boldsymbol{A}^{(k-1)}, \boldsymbol{B}^{(k-1)}$ by using the recursive Equations 18-20 ;
6 $\phi^{(k)} \leftarrow (\boldsymbol{A}^{(k)})^{-1}\boldsymbol{B}^{(k)}$ according to Equation 12 ;
7 **return** $\phi^{(k)}$

For the following iterations $k > 0$, the recursive computing of $\boldsymbol{A}^{(k)}, \boldsymbol{B}^{(k)}$ from $\boldsymbol{A}^{(k-1)}, \boldsymbol{B}^{(k-1)}$ performs. As presented in Proposition 10, all the $p^2 + p$ values in $\boldsymbol{A}^{(k)}, \boldsymbol{B}^{(k)}$ can be recursively updated in constant time. Consequently, the complexity of parameter estimation is reduced from $O(n)$ (referring to Equations 15-17) to $O(1)$ in Equations 18-20.

Example 10 (Parameter estimation using incremental computation, Example 6 continued)**.** *Consider again* $x = \{6, 10, 9.6, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *and* $y^{(0)} = \{6, 5.6, 5.4, 8.3, 7.7, 5.4, 5.6, 5.9, 6.3, 6.8, 7.5, 8.5\}$ *in Example 6. We have* $\boldsymbol{V}^{(0)} = \{-4.4, -4.2, 0, 0, 0, 0, 0, 0, 0, 0, 0\}'$ *and* $\boldsymbol{Z}^{(0)} = \{0, -4.4, -4.2, 0, 0, 0, 0, 0, 0, 0, 0\}'$. *Given* $p = 1$*, the matrices* $\boldsymbol{A}^{(0)}, \boldsymbol{B}^{(0)}$ *have only one element, with* $a_{11}^{(0)} = (-4.4)^2 + (-4.2)^2 = 37$ *and* $b_1^{(0)} = (-4.4) * (-4.2) = 18.48$ *initialized by Line 2 in Algorithm 2.*

According to Examples 7 and 8, the repaired point is $y_4^{(1)} = 6.2$*. We have* $z_4^{(1)} = 6.2 - 8.3 = -2.1$*, while* $z_4^{(0)} = 0$*. Line 5 in Algorithm 2 incrementally computes* $\boldsymbol{A}^{(1)}, \boldsymbol{B}^{(1)}$ *from* $\boldsymbol{A}^{(0)}, \boldsymbol{B}^{(0)}$*, i.e.,* $a_{11}^{(1)} = a_{11}^{(0)} + (-2.1)^2 = 41.41$ *by using the incremental update in Equation 18, and* $b_1^{(1)} = b_1^{(0)} + (-2.1 - 0) * (-4.2 + 0) = 27.3$ *referring to Equation 20. Finally, the parameter* $\phi_1^{(1)}$ *is computed according to* $\boldsymbol{A}^{(1)}, \boldsymbol{B}^{(1)}$ *by using Equation 12, i.e.,* $\phi_1^{(1)} = 27.3/41.41 = 0.66$.

6. EXPERIMENT

In this section, we experimentally compare our proposed methods IMR with the state-of-the-art approaches, including the anomaly detection methods using (1) AR [3], (2) ARX [3], (3) ARIMA [18, 3], (4) Tsay [23] as models, (5) the smoothing-based method EWMA [13], and (6) the constraint-based approach SCREEN [22].

GPS data with real errors. In the GPS dataset, real errors are naturally embedded and the corresponding ground truths are manually labeled. It collects GPS readings by a person carrying a smartphone and walking around at campus. Since we know exactly the path of walking, a number of 186 dirty points out of 742 points in trajectory are manually identified. True locations of dirty points are also manually labeled, as ground truth. (See major results in Section 6.1.)

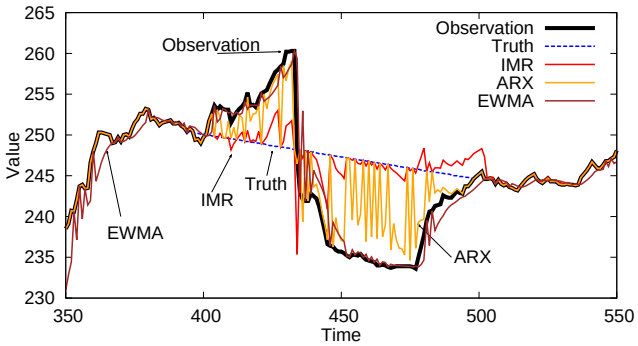

Figure 3: GPS example

ILD data with synthetic errors. The Intel Lab Data (`http://db.csail.mit.edu/labdata/labdata.html`, ILD) includes a number of measurements taken from 54 sensors for every 31 seconds in about 38 days. Taking 31 seconds as one epoch and omitting the missing data, a dataset of 4912 points is obtained in sensor 1 from Feb 29th to Mar 1st. We synthetically inject errors into the data, by shifting the values for an amount of 3 with variance 0.1 under Gaussian distribution (see some examples in Figure 8). Such "shifting" errors are very common in practice, for example the sensor is stuck for a short while, or unit error in collection in a period. (See major results in Section 6.2.)

Criteria. RMS error [15] is employed to evaluate the repair. Let x^{truth} be the ground truth of clean sequence, x^{dirty} be the observation sequence with faults embedded, and x^{repair} be the repaired sequence. The RMS error [15] is given by:

$$\Delta(x^{\text{truth}}, x^{\text{repair}}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^{\text{truth}} - x_i^{\text{repair}})^2}.$$

The measure evaluates the distance between the ground truth and its repair result. Low RMS error is preferred.

6.1 Experiments on Real Errors

The experiments on real errors over GPS data consider various algorithm settings, including (1) order p, (2) convergence threshold τ, (3) max-num-iterations, and (4) labeling rate. Similar results are also observed in ILD and omitted.

6.1.1 Example Results

Figure 3 illustrates an example part (in latitude, after transformed) of the GPS dataset, including the collected observations with errors, the labeled truth, and the repair results by different methods. Owing to various influences such as buildings, GPS readings may deviate from the truth. For instance, the data between time points 400 and 500 are collected from a place near a high building, where significant errors are observed. As shown, the proposed IMR shows a repair closest to the truth, compared to other methods.

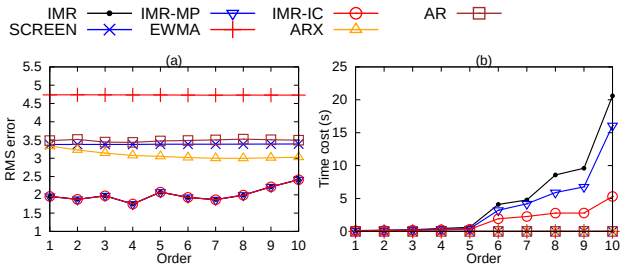

Figure 4: Varying order p, over GPS with $\tau = 0.2$, data size 750, and labeling rate 0.2

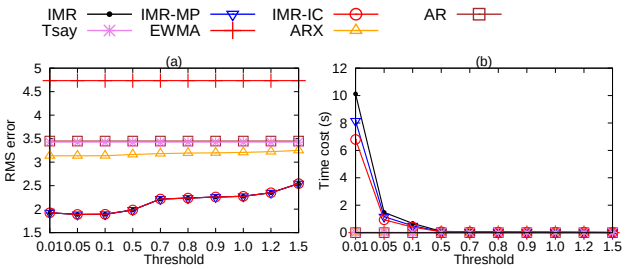

Figure 5: Varying threshold τ, over GPS with $p = 3$, data size 750, and labeling rate 0.2

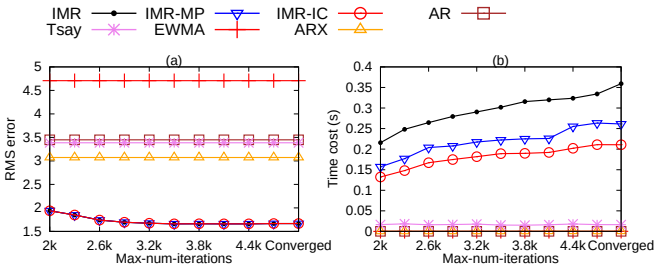

Figure 6: Varying maximum number of iterations, over GPS with $\tau = 0.2, p = 3$ and data size 750

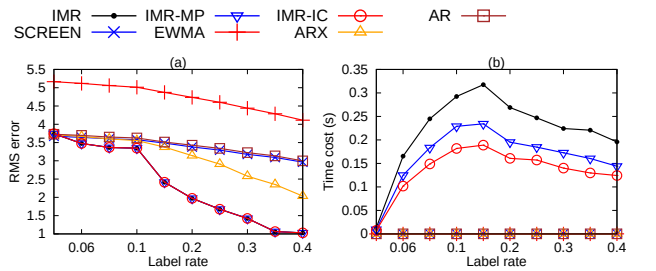

Figure 7: Varying labeling rate, over GPS with $\tau = 0.2, p = 3$ and data size 750

6.1.2 *Varying Order p*

Figure 4 presents the results on varying order p, for AR(p), ARX(p) and IMR(p). First, as shown in Figure 4(a), AR-based method shows better performance with the increase of order p, where more historical values take effect in the predication of each value. An excessively large p, however, does not show further improvement, since this simple model may not be able to capture the complicated semantics in a large window. Similar results are also observed in ARX for the same reason. Remarkably, owing to the iterative strategy with minimum repair in each iteration, our proposed IMR method already achieves low RMS error of repairing even with $p = 1$. The results verify the necessity of analyzing the special case of IMR(1) with $p = 1$ in Section 4.

It is not surprising that the iterative IMR needs higher time costs in Figure 4(b) than other existing methods with only one pass through the data. In addition to the original IMR in Algorithm 1, we also present the results of IMR with matrix pruning (IMR-MP) in Section 5.1 and incremental computation (IMR-IC) in Algorithm 2 for efficient parameter estimation. IMR, IMR-MP and IMR-IC show exactly the same accuracy results in Figure 4(a). Both efficient estimation methods improve the time costs in Figure 4(b). In particular, IMR-IC for incremental parameter estimation with constant time significantly reduces time costs.

In summary, as illustrated in Figures 4 and 10 over the GPS and ILD datasets, respectively, our proposed IMR has no clear preference of order p in repairing accuracy, while larger order p leads to higher time cost.

6.1.3 *Varying Convergence Threshold τ*

Figure 5 reports the results by varying the threshold τ (with $p = 3$). By setting a small τ, IMR needs more iterations to converge. The corresponding time costs in Figure 5(b) are higher. Better repairing performance is achieved by IMR with a small τ, as shown in Figure 5(a). However, by further reducing the threshold τ, e.g., from 0.1 to 0.01, the repair accuracy could hardly be further improved, while the corresponding iterations and time costs significantly increase. On the other hand, by increasing the threshold τ, the time costs reduce. Indeed, the threshold τ provides a trade-off between repair accuracy and time costs for IMR.

In summary, a lower threshold indeed leads to better results (lower RMS error) and needs more iterations (higher time costs). See Figure 11 on ILD for more clear impact of the threshold.

6.1.4 *Specifying Maximum Number of Iterations*

In Section 4, we analyze several special cases, where repairing is guaranteed to converge in theory under some conditions. For general cases where such conditions are not met, (although all the experiments converge in Section 6 under various settings with/without theoretical convergence guarantee), one may specify the maximum number of iterations, as a remedy in practice to avoid waiting for convergence. That is, Algorithm 1 terminates when the iteration number reaches *max-num-iterations*, even if the convergence condition in Line 5 is not met.

Figures 6 evaluates various settings of maximum number of iterations (average time costs are reported by repeating each test 10 times).As illustrated, a moderately large number of iterations already achieve good repair results, i.e., close to the (right-most) converged results.

6.1.5 *Varying Labeling Rate*

Figure 7 illustrates the results on various labeling rates. A labeling rate 0.1 denotes that 10% data points are labeled with truth in the dataset. It is not surprising that the higher the labeling rate is, the better the repair performance of IMR and ARX will be, which utilize the labeled truth, as illustrated in Figure 7(a). An interesting result is that with the increase of labeling rate, the corresponding time costs in

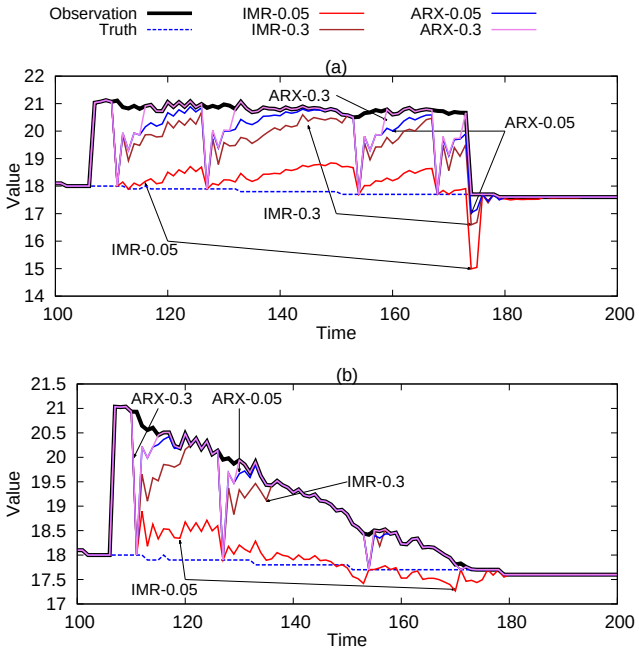

Figure 8: ILD example with (a) Shift and (b) Innovational errors

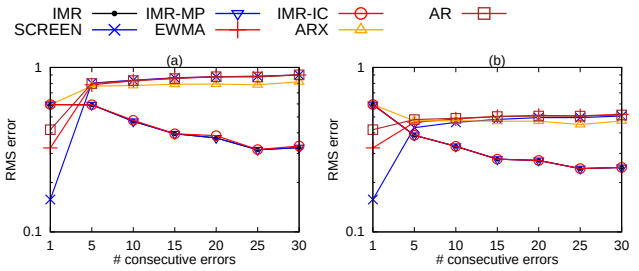

Figure 9: Varying the number of consecutive errors, under (a)Shift and (b) Innovational error types, over ILD with $\tau = 0.1, p = 3$ and data size 3k

Figure 7(b), first increase and then drop. The reason is that for a small labeling rate (say 0.06) with points barely modified, the iterative repair can quickly converge, while leaving most dirty data unchanged. The corresponding RMS error is high in this case as shown in Figure 7(a). With more data labeled in the input, more dirty points will be identified and repaired by the algorithm, leading to higher computation costs. When the labeling rate is large, such as 0.25, a great number of dirty points may be labeled. Thereby, the iterative repair could converge quickly again.

As anomaly detection methods, the results of Tsay and ARIMA are generally similar to those of AR and ARX. SCREEN and EWMA methods are not affected by order p, threshold τ and labeling rate. It is not surprising that SCREEN performs weakly, which verifies our motivation and analysis in the Introduction. Similarly, since EWMA does not utilize the labeled truth, its performance is weak.

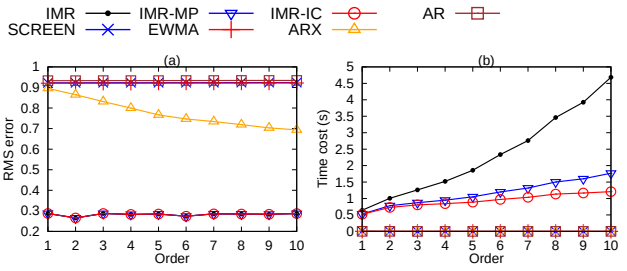

Figure 10: Varying order p, over ILD with $\tau = 0.1$, data size 3k, and labeling rate 0.2

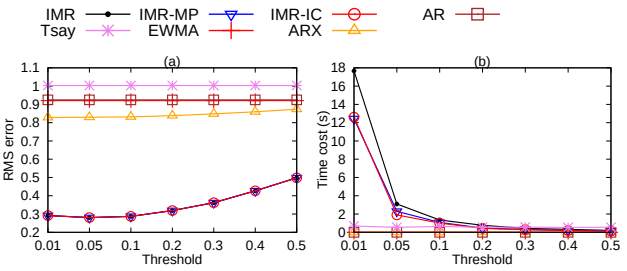

Figure 11: Varying threshold τ, over ILD with $p = 3$, data size 3k, and labeling rate 0.2

6.2 Experiments on Synthetic Errors

The experiments on synthetic errors over ILD data focus on varying the errors. Again, similar results are also observed in the other dataset GPS and thus omitted.

6.2.1 *Evaluation on Various Errors*

We consider Shift and Innovational errors [2, 23], as the example illustrated in Figure 8, where Spike errors are considered as a special case with # consecutive errors = 1. The corresponding accuracy results are reported in Figure 9. In general, similar results are observed over Innovational and Shift errors. That is, while our proposed IMR may not deal with Spike errors (i.e., with # consecutive errors = 1 in Figure 9) as good as SCREEN [22], IMR always shows significantly better results (lower RMS measure) in addressing a large number of consecutive errors, on both Shift and Innovational error patterns. The results demonstrate again that our proposal works well in repairing consecutive errors.

As illustrated in Figure 8(a), not only the proposed IMR but also ARX with a small threshold τ suffers from an over-correction when the time series shifts back to non-anomalous data. The number attached to each method, e.g., IMR-0.05, denotes $\tau = 0.05$ for IMR. As shown, there is a trade-off in both IMR and ARX: a smaller threshold τ shows better results in dealing with consecutive errors, but leads to over-correction when the shift ends. Nevertheless, as presented in Figures 5 and 11, a smaller threshold τ generally has better overall accuracy. It is also worth noting that existing methods, EWMA smoothing and SCREEN, cannot handle well the repairing either when the time series shifts back to non-anomalous data, as the example illustrated in Figure 1.

6.2.2 *Evaluation on Online Computing*

As long as online labeling is available (discussed in Section 1.3), the proposed IMR is applicable. Remarkably, by interpreting all the historical data as one labeled segment,

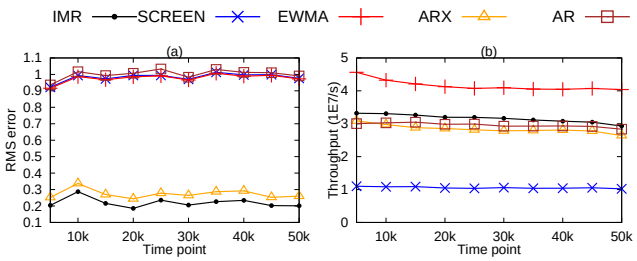

Figure 12: Online computing, over ILD

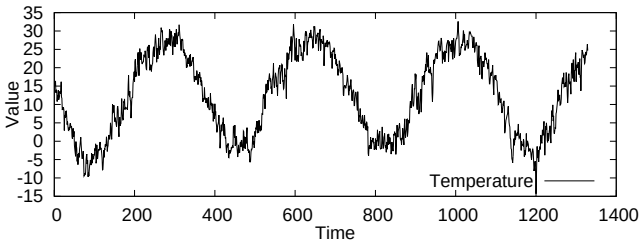

Figure 13: Cyclic Temperature example

the direct calculation of repairs without iterating in Section 4.2.2 can be applied for efficient online computing.

Figure 12 presents the results of online repairing. Since errors are randomly introduced in the dataset, to obtain reliable results, we repeat 10 times for each test with random error introducing and report the average. Troughput of IMR is stable and comparable to others. IMR again shows the best repair. Improvement of IMR compared to ARX is not as significant as in other experiments. It is not surprising, referring to the similar Equation 3 for ARX and Equation 10 for IMR(1). The advantage of IMR is that no threshold parameter is required for IMR(1) with one labeled segment in Section 4.2.2, while ARX needs to set threshold τ.

6.2.3 *Experiments on Temperature*

To evaluate over cyclic time series, we employ another data set on temperature in years (`http://data.cma.cn`, with cyclic patterns as illustrated in Figure 13). Similar to ILD, we inject synthetic errors to the temperature data. Figure 14 presents the results over different error types. Again, the results are generally similar to those on ILD in Figure 9. That is, the proposed IMR shows significantly better results when dealing with a large number of consecutive errors. SCREEN shows no better results in this dataset under Spike errors (i.e., # consecutive errors = 1), since the clean data also contain a large number of Spikes as illustrated in Figure 13 and cannot be distinguished from errors.

7. RELATED WORK

The idea of performing repair in multiple iterations has also been studied [24], where the past repairs could help in recommend more accurate repairs in the future. The continuous data cleaning approach [24] however is not directly applicable in our problem, since it employs FD constraints which is not available over time series data.

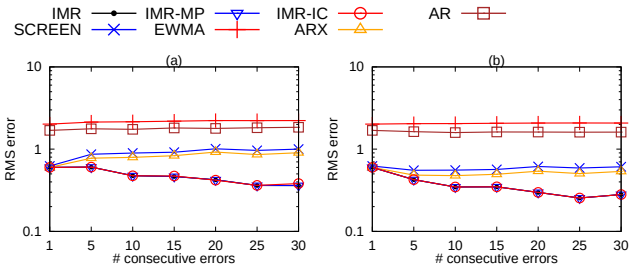

Figure 14: Varying number of consecutive errors, under (a)Shift and (b)Innovational error types, over Temperature with $\tau = 0.2, p = 3$ and data size 1.5k

7.1 Anomaly Detection over Temporal Data

AR and ARX indeed have been widely used for anomaly detection in various areas such as economics and social surveys [3, 6]. We consider ARX [19] in this study, since this approach can utilize the labeled truth and could be adapted to cooperate with the minimum change principle in data repairing (as shown in Section 3).

Hellerstein [13] surveys methods on cleaning errors in quantitative attributes of large databases, where time series data are also discussed as a special scenario. For instance, Tsay [23] presents unified methods for detecting and handling outliers and structure changes in a univariate timeseries. Iterative procedures consist of specification-estimation-detecting-removing cycles to handle one-by-one the most significant disturbance. ARIMA [18, 3] is a general parametric family of time series, consisting of autoregressive process and moving-average process. It can also incorporate a wide range of nonstationary series.

7.2 Smoothing-based Cleaning

Smoothing techniques are often employed to eliminate noisy data. For example, the simple moving average (SMA) [4] smooths time series data by computing the unweighted mean of the last k points. Instead of weighting equally, the exponentially weighted moving average (EWMA) [10] assigns exponentially decreasing weights over time. As indicated in [22], also illustrated in Figure 2 in Example 2 and observed in Figure 3 (EWMA) in the experiments, the smoothing methods may seriously alter the original correct data, and thus have low repair accuracy. In contrast, our minimum change-based approach, applying only high confidence repairs, could preserve most the original values with a better repair accuracy.

7.3 Constraint-based Cleaning

Constraint-based repairing is widely considered in cleaning dirty data, such that the repaired data satisfies some given constraints and the repair modification is minimized [1, 8]. To clean sequential data, existing study [22] employs a class of speed constraints declaring that the speeds of value changes should be bounded. The repairing is thus to modify the sequence towards the satisfaction of such speed constraints. This constraint-based repairing falls short in two aspects: (1) it cannot handle a sequence of continuous errors, and (2) the labeled truth is not utilized.

8. CONCLUSION

In this paper, we study the problem of repairing dirty time series data, given the labeled truth of some data points. (1) While existing anomaly detection techniques could be adapted to repairing, we argue that significant deviation (between observation and predication) based anomaly detection is inconsistent with the minimum change principle in data repairing. Our experiments over real datasets illustrate such inconformity of applying anomaly detection in anomaly repairing. (2) We thereby propose an iterative minimum repairing (IMR) algorithm. By creatively performing one minimum repair in each iteration of error predication, the algorithm bonds the beauty of capturing temporal nature in anomaly detection with the minimum change in data repairing. Again, the experiments demonstrate the superiority of our proposal. Remarkably, in contrast to anomaly detection approaches AR and ARX, our proposed IMR is not sensitive to the setting of order p, i.e., a small p is sufficient to achieve high repair accuracy with low time costs. (3) The convergence of IMR is explicitly analyzed. In particular, we show that the converged repair result could be directly calculated without iterative computing in certain cases, which enables efficient online repairing over streaming data. It is worth noting that unlike the existing ARX, no threshold needs to be specified for IMR in online computing. (4) Finally, we design efficient pruning and incremental computation, which reduce the complexity of parameter estimation from linear time to constant time. Experiments illustrate the significant improvement on time performance by pruning and incremental computation.

Acknowledgment

This work is supported in part by National Key Research Program of China under Grant 2016YFB1001101; China NSFC under Grants 61572272, 61325008, 61370055, 61672313 and 61202008; Tsinghua University Initiative Scientific Research Program. Shaoxu Song is a corresponding author.

9. REFERENCES

[1] P. Bohannon, M. Flaster, W. Fan, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, pages 143–154, 2005.

[2] G. E. Box and G. C. Tiao. Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical association*, 70(349):70–79, 1975.

[3] G. E. P. Box and G. M. Jenkins. Time series analysis: Forecasting and control. 1994.

[4] D. R. Brillinger. *Time series: data analysis and theory*, volume 36. Siam, 2001.

[5] P. J. Brockwell and R. A. Davis. *Introduction to time series and forecasting*. Springer Science & Business Media, 2006.

[6] P. J. Brockwell and R. A. Davis. *Introduction to time series and forecasting*. springer, 2016.

[7] B. Cheng. Yule–walker equations. *Wiley StatsRef: Statistics Reference Online*, 2014.

[8] X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *ICDE*, pages 458–469, 2013.

[9] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *PVLDB*, 3(1):173–184, 2010.

[10] E. S. Gardner Jr. Exponential smoothing: The state of the art–part ii. *International Journal of Forecasting*, 22(4):637–666, 2006.

[11] M. Gupta, J. Gao, C. Aggarwal, and J. Han. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 5(1):1–129, 2014.

[12] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[13] J. M. Hellerstein. Quantitative data cleaning for large databases. *United Nations Economic Commission for Europe (UNECE)*, 2008.

[14] D. J. Hill and B. S. Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software*, 25(9):1014–1022, 2010.

[15] S. R. Jeffery, M. N. Garofalakis, and M. J. Franklin. Adaptive cleaning for RFID data streams. In *VLDB*, pages 163–174, 2006.

[16] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: Is the problem solved? *PVLDB*, 6(2):97–108, 2012.

[17] F. Mörchen. Algorithms for time series knowledge mining. In *KDD*, pages 668–673, 2006.

[18] M. C. Otto and W. R. Bell. Two issues in time series outlier detection using indicator variables. In *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, pages 182–187, 1990.

[19] G. Park, A. C. Rutherford, H. Sohn, and C. R. Farrar. An outlier analysis framework for impedance-based structural health monitoring. *Journal of Sound and Vibration*, 286(1):229–250, 2005.

[20] C. R. Rao. *Linear statistical inference and its applications*, volume 22. John Wiley & Sons, 2009.

[21] S. Song, C. Li, and X. Zhang. Turn waste into wealth: On simultaneous clustering and cleaning over dirty data. In *KDD*, pages 1115–1124, 2015.

[22] S. Song, A. Zhang, J. Wang, and P. S. Yu. SCREEN: stream data cleaning under speed constraints. In *SIGMOD*, pages 827–841, 2015.

[23] R. S. Tsay. Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, 7(1):1–20, 1988.

[24] M. Volkovs, F. Chiang, J. Szlichta, and R. J. Miller. Continuous data cleaning. In *ICDE*, pages 244–255, 2014.

[25] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 7. MIT press Cambridge, MA, 1949.

[26] Z. Xing, J. Pei, and P. S. Yu. Early classification on time series. *Knowl. Inf. Syst.*, 31(1):105–127, 2012.

[27] K. Yamanishi and J.-i. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *KDD*, pages 676–681. ACM, 2002.

[28] Y. Zheng, F. Liu, and H. Hsieh. U-air: when urban air quality inference meets big data. In *KDD*, pages 1436–1444, 2013.