IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 10, 2019, accepted December 21, 2019, date of publication December 25, 2019,
date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962152

Time Series Data Cleaning: A Survey

XI WANG AND CHEN WANG
School of Software, Tsinghua University, Beijing 100084, China
Corresponding author: Chen Wang (wang_chen@tsinghua.edu.cn)

This work was supported in part by the National Key Research and Development Plan under Grant 2017YFC0804307 and Grant 2019YFB1705301, and in part by the National Natural Science Foundation of China under Grant 61572272 and Grant 71690231.

ABSTRACT Errors are prevalent in time series data, which is particularly common in the industrial field. Data with errors could not be stored in the database, which results in the loss of data assets. At present, to deal with these time series containing errors, besides keeping original erroneous data, discarding erroneous data and manually checking erroneous data, we can also use the cleaning algorithm widely used in the database to automatically clean the time series data. This survey provides a classification of time series data cleaning techniques and comprehensively reviews the state-of-the-art methods of each type. Besides we summarize data cleaning tools, systems and evaluation criteria from research and industry. Finally, we highlight possible directions time series data cleaning.

INDEX TERMS Data cleaning, data quality, time series.

I. INTRODUCTION

Time series data can be defined [128] as a sequence of random variables, $x_1, x_2, \ldots, x_n$, where the random variable x_1 denotes the value taken by the series at the first time point, the variable x_2 denotes the value for the second time period, x_n denotes the value for the n-th time period, and so on. Time series have been widely used in many fields [11], [16], [49] such as financial economy, meteorology and hydrology, signal processing, industrial manufacturing, and so on. Time series data are important in industry, where there are all kinds of sensor devices capturing data from the industrial environment uninterruptedly. Owing to the fact that data of the sensor devices are often unreliable [53], time series data are often large and dirty. In the financial field, the most important application of time series data is to predict future commodity (stock) price movements. However, time series errors in the financial field are also very prevalent, even some data sets, which are considered quite accurate, still have erroneous data. For instance, the correct rate of stock information on Yahoo Finance is 93%. The costs and risks of errors, conflicts, and inconsistencies in time series have drawn widespread attention from businesses and government agencies. In recent work, the data quality issues in time series data are studied, since they pose unique data quality challenges due to the presence of autocorrelations, trends, seasonality, and gaps in the time series data [25]. According to Shilakes and Tylman [78], the relevant market growth rate of data quality is about 17%, which is much higher than the 7% annual growth rate of the IT industry. For instance, approximately 30% to 80% of the time and cost are spent on data cleaning in data warehousing project development. The time series errors can be either timestamp errors or observed value errors. For possible timestamp errors, Song *et al.* [80] propose a method for cleaning timestamps. In this survey, we focus on the existing methods of dealing with observed value errors, thereby, the time series errors mentioned in the following are observation errors. There are two types of processing methods commonly used in the industry when dealing with time series data errors:

(1) Discarding erroneous data. First, the time series is detected via using an anomaly detection algorithm, and then the detected abnormal data are discarded.

(2) Cleaning data. Data cleaning is divided into manual cleaning and automatic cleaning. There is no doubt that manual cleaning has a high accuracy rate, but it is difficult to implement because it takes more time and effort.

The existing surveys of data cleaning [23], [50] mainly summarize the methods of dealing with data missing, data inconsistence, data integration and erroneous data in the database. Karkouch *et al.* [58] review the generation of sensor data, the reasons for the formation of data quality problems, and the techniques for improving data quality. However, Karkouch *et al.* [58] do not provide a detailed overview of the existing state-of-the-art of erroneous data cleaning. Thereby, **we review the state-of-the-art of time series data error value cleaning**, which may provide a tutorial for others.

The associate editor coordinating the review of this manuscript and approving it for publication was Feng Xia.

A. PROBLEM STATEMENT

In this study, enlightened by related research [48], [86] and [101] on the classification of time series error types, we summarize the common error in time series into three categories, namely, single point big error, single point small error and continuous errors. This article takes the stock price of a stock for 30 consecutive trading days as an example. As shown in Figure 1, the characteristics of these three types of errors are described in detail. The red line in the figure indicates the true price of the stock in 30 consecutive trading days, and the blue line indicates the price of the stock crawled by a website. For various reasons, the observed value may not be the same as the actual value. It can be seen that in the four consecutive trading days of 8-11, the observed values are all 0, and the true values are 1.3, 1.2, 1.1 and 1.15, respectively, on the 20th trading day, the observed value is 2.4 and the true value is 1.3, on the 25th trading day, the observed value is 1 and the true value is 1.3.

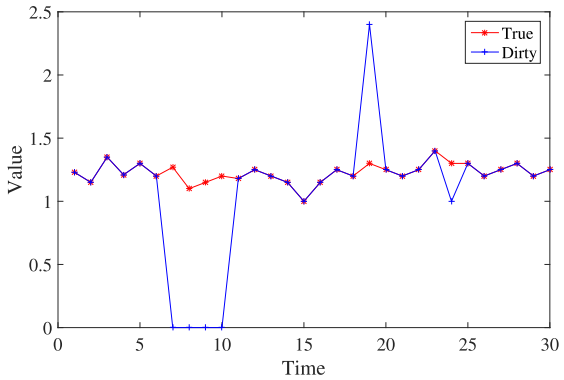

FIGURE 1. An example error type.

(1) Continuous errors. The so-called continuous errors, that is, in the time series, errors occur in several consecutive time points. Specifically, continuous errors can continue to be subdivided into several types [86], but no longer detailed here. The observed values from the 8th to 11th trading days in Figure 1 are all 0, that is, continuous errors occur here. Continuous errors are common in real life. For instance, when someone is holding a smartphone and walking on the road, nearby tall buildings may have a lasting impact on the collected GPS information. Besides, system errors can also cause continuous errors.

(2) A single big error. A single point error is an error that occurs discontinuously in a time series and only occurs on a single data point at intervals. A big error means that the observed value of the data point is far from the true value. Remarkably, the size of the error is relative and closely related to the real situation of the data set. As shown on the 20th trading day in Figure 1, the observed value differs from the true value by 1.1. Compared with the 25th trading day when the observed value differs from the true value by 0.3, the error of this data point is large, so this data point error is a single big error. The single point big error is also very common in daily life. For instance, the data of motor vehicle oil level recorded by cursors may cause a single big error when bumping on the road.

(3) A single small error. Similar to a single point big error, that is, errors do not occur consecutively, only on a single data point at intervals. When the observed value of the data point differs from the true value by a small distance, on the 25th trading day in Figure 1, it is a single point small error. As stated in [10], the rationale behind single-point small errors is that people or systems always try to minimize possible errors. For instance, people may only have some small omissions when copying files.

(4) Translational error. As shown in Figure 2, where x axis represents time and y axis represents the value of the corresponding time, the red line represents true value, and the blue line represents error value after the translation, the solution to this type of error is not as much as mentioned above.

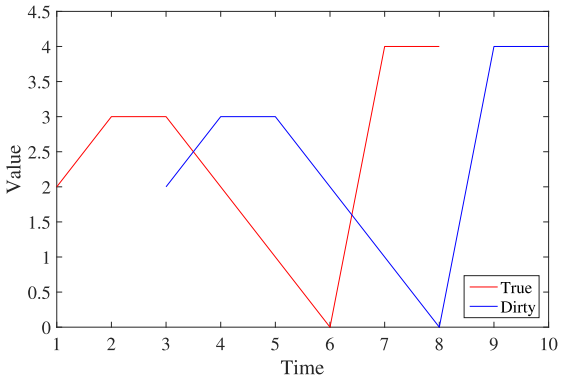

FIGURE 2. An example of translational error.

Ignoring time series errors often results in unpredictable consequences for a series of applications such as query analysis. Thereby, time series cleaning algorithms are very important for mining the potential value of data. This paper reviews the cleaning algorithm and anomaly detection algorithm of time series data. By summarizing the existing methods, a reference or guidance is given to scholars interested in time series data cleaning and based on this, the possible challenges and future work of time series cleaning topics are discussed.

B. PROBLEM CHALLENGE

For the problem of time series data cleaning, the following four difficulties have been discovered through the survey:

(1) The amount of data is large and the error rate is high. The main source of time series data is sensor acquisition. Especially in the industrial field, sensors distributed throughout the machine are constantly monitoring the operation of the machine in real-time. These sensors often collect data at a frequency of seconds, and the amount of data collected is quite large. For instance, the sensor collection interval of a wind power company equipment is 7 seconds, each machine has more than 2000 working status data, and more than 30 million pieces of data are collected every day, so the

working status data of one day could exceed 60 billion. However, the data collected by the sensor are often not accurate enough, and some because of the physical quantity of the observation is difficult to measure accurately. For instance, in a steel mill, with affecting by environmental disturbances the surface temperature of the continuous casting slab cannot be accurately measured or may cause distortion due to the power of the sensor itself.

(2) The reasons for generating time series data errors are complicated. People always try to avoid the generation of time series erroneous data, however, there are various time series errors. Besides the observed errors that we mentioned above for various reasons, Karkouch *et al.* [58] also explain in detail the IoT data errors generated by various complex environments. IoT data is a common time series of data, and its widespread existence is really in the world. The complex reasons of time series errors also are challenges we face in cleaning and analyzing data that is different from traditional relational data.

(3) Time series data are continuously generated and stored. The biggest difference between time series data and relational data is that the time series is continuous. Thereby, for time series data, it is important that the cleaning algorithm supports online operations (real-time operations). The online anomaly detection or cleaning algorithm can monitor the physical quantity in real-time, detect the problem and then promptly alarm or perform a reasonable cleaning. Thereby, the time series cleaning algorithm is not only required to support online calculation or streaming calculation but also has good throughput.

(4) Minimum modification principle [1], [22], [35]. Time series data often contain many errors. Most of the widely used time series cleaning methods utilize the principle of smooth filtering. Such methods may change the original data too much, and result in the loss of the information contained in the original data. Data cleaning needs to avoid changing the original correct data. It should be based on the principle of minimum modification, that is, the smaller the change, the better.

C. ORGANIZATION

Different algorithms tackle these challenges in different ways, which usually include smoothing-based methods, constraint-based methods, and statistical-based methods as shown in Table 1. Besides some time series anomaly detection algorithms can also be effectively used to clean data. The remainder of this paper is organized as follows. The aforesaid four types of algorithms are discussed from Section II to V, respectively. In Section VI we introduce existing time series cleaning tools, systems, and evaluation criteria. Finally, we summarize this paper in Section VII and discuss possible future directions.

II. SMOOTHING BASED CLEANING ALGORITHM

Smoothing techniques are often used to eliminate data noise, especially numerical data noise. Low-pass filtering, which filters out the lower frequency of the data set, is a simple algorithm. The characteristic of this type of technology is that the time overhead is small, but because the original data may be modified much, which makes the data distorted and leads to the uncertainty of the analysis results, there are not many applications used in time series cleaning. The research of smoothing technology mainly focuses on algorithms such as Moving Average (MA) [15], Autoregressive (AR) [11], [51], [97] and Kalman filter model [57], [68], [70]. Thereby, this chapter mainly introduce these three technologies and their extensions.

TABLE 1. The overview of methods.

Type	Method
Smoothing based (Section II)	Moving Average [15] AutoRegressive [11], [51], [97] Autoregressive Moving Average Model [3], [30], [86] Kalman Filter [18], [33], [38], [45], [57], [68], [70], [105] Interpolation [59], [95] State-space Model [56], [68], [87]
Constraint based (Section III)	Order Dependencies [32], [41]–[43] Denial Constraints [65] Sequential Dependencies [46] Speed Constraints [82] Variance Constraints [66] Similarity Rule Constraints [119] Learning Individual Models [120]
Statistics based (Section IV)	Maximum Likelihood [13], [44], [89], [96], [99] Bayesian Model [39], [90] Markov Model [112]–[114] Hidden Markov Model [4], [116]–[118] SMURF [27], [54], [115] Spatio-Temporal Probabilistic Model [124] Expectation-Maximization [79] Relationship-dependent Network [7], [69], [71]
Anomaly detection (Section V	Density-Based Spatial Clustering of Applications with Noise [29], [123] Local Outlier Factor [29] Abnormal Sequence Detection [24], [47], [61] Window-based Anomaly Detection [63] Generative Adversarial Networks [76], [121], [122] Long Short-Term Memory [107]–[109]

A. MOVING AVERAGE

The moving average (MA) series algorithm [15] is widely used in time series for smoothing and time series prediction. A simple moving average (SMA) algorithm: Calculate the average of the most recently observed N time series values, which is used to predict the value at time t. A simple definition as shown in equation (1).

$$\widehat{x_t} = \frac{1}{2n+1} \sum_{i=-n}^{n} x_{i+t} \tag{1}$$

In equation (1), $\widehat{x_t}$ is the predicted value of x_t, x_t represents the true value at time t, $2 * n + 1$ is the window size(count). For a time series $x(t) = 9.8, 8.5, 5.4, 5.6, 5.9, 9.2, 7.4$, for simplicity, we use SMA with $n = 3$ and $k = 4$, then calculate according to equation (2).

$$x_t = \frac{(x_{t-n} + x_{t-2} \ldots + x_{t+n})}{(2n+1)} = 7.4 \tag{2}$$

To eliminate errors or noises, the data $x(t)$ can be considered as a certain time window of sliding window, then continuously calculate the local average over a given interval $2n + 1$ to filter out the noise (erroneous data) and get a smoother measurement.

In the weighted moving average (WMA) algorithm, data points at different relative positions in the window have different weights. Generally defined as:

$$\widehat{x_t} = \sum_{i=-n}^{n} \omega_i x_{i+t} \tag{3}$$

In equation (3), ω_i represents the weight of the influence of the i position data point on the t position data point, other definitions follow the example above. A simple strategy is that the farther away from the two data points, the smaller the mutual influence. For instance, a natural idea is the reciprocal of the distance between two data points as the weight of their mutual influence. Similarly, the weight of each data point in the exponential weighted moving average (EWMA) algorithm [38] decreases exponentially with increasing distance, which is mainly used for unsteady time series [17], [52].

Aiming at the need for the rapid response of sensor data cleaning, Zhuang *et al.* [105] propose an intelligent weighted moving average algorithm, which calculates weighted moving averages via collected confidence data from sensors. Zhang *et al.* [103] propose a method based on multi-threshold control and approximate positive transform to clean the probe vehicle data, and fill the lost data with the weighted average method and exponential smoothing method. Qu *et al.* [74] first use cluster-based methods for anomaly detection and then use exponentially weighted averaging for data repair, which is used to clean power data in a distributed environment.

B. AUTOREGRESSIVE

The Autoregressive (AR) Model is a process that uses itself as a regression variable and uses the linear combination of the previous k random variables to describe the linear regression model of the random variable at the time t. The definition of AR model [51], [88] as shown in equation (4).

$$\widehat{x_t} = \sum_{i=1}^{k} \omega_i x_{t-i} + \epsilon_t + a \tag{4}$$

In equation (4), $\widehat{x_t}$ is the predicted value of x_t, x_t represents the true value at time t, k is the order, μ is mean value of the process, ϵ_t is white noise, ω_i is the parameter of the model, a is a constant.

Park *et al.* [72] use labeled data y to propose an autoregressive with exogenous input (ARX) model based on the AR model:

$$\widehat{y_t} = x_t + \sum_{i=1}^{k} \omega_i (y_{t-i} - x_{t-i}) + \epsilon_t \tag{5}$$

In equation (5), $\widehat{y_t}$ is the possible repair of x_t, and others are the same to the aforesaid AR model. Alengrin and Favier [3] propose Autoregressive moving average (ARMA) model), which is composed of the AR model and MA model. Besides that, the Gaussian Autoregressive Moving Average model is defined as shown in equation (6) [86].

$$\Phi(B) Z_t = \theta_0 + \theta(B) x_t \tag{6}$$

In equation (6), $\Phi(B) = 1 - \Phi_1 B - \ldots - \Phi_p B^p$ and $\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$ are polynomial in B of degrees p and q, respectively, θ_0 is a constant, B is the backshift operator such that $BZ_t = Z_{t-1}$, and x_t is a sequence of independent Gaussian variates with mean $\mu = 0$ and variance σ_x^2. Box and Pierce [12] propose a more complex Autoregressive Integrated Moving Average (ARIMA) model based on the ARMA model, which is not described in detail here. Akouemo and Povinelli [2] propose a method combining ARX and Artificial Neural Network (ANN) model for cleaning time series, which performs a hypothesis test to detect anomalies the extrema of the residuals, and repairs anomalous data points by using the ARX and ANN models. Dilling and MacVicar [30] clean high-frequency velocity profile data with ARMA model and Chen *et al.* [21] use the ARIMA model to clean wind power data.

C. KALMAN FILTER MODEL

Kalman [57] proposes the Kalman filter theory, which can deal with time-varying systems, non-stationary signals, and multi-dimensional signals. Kalman filter creatively incorporates errors (predictive and measurement errors) into the calculation, the errors exist independently and are not affected by measured data. The Kalman model involves probability, random variable, Gaussian Distribution, and State-space Model, etc. Consider that the Kalman model involves too much content, no specific description is given here, and only a simple definition is given. First, we introduce a system of discrete control processes which can be described by a Linear Stochastic Difference equation as shown in equation (7).

$$x_t = m x_{t-1} + n v_t + p_t \tag{7}$$

Also, the measured values of the system are expressed as shown in equation (8).

$$y_t = r x_{t-1} + q_t \tag{8}$$

In equation (7) and (8), x_t is the system state value at time t, and v_t is the control variable value for the system at time t. m and n are system parameters, and for multi-model systems, they are matrices, y_t is the measured value at time t, r is the parameter of the measurement system,

TABLE 2. Summary of smoothing.

Reference	Method
[15]	MA
[103], [105]	WMA
[38], [74], [103]	EWMA
[51], [88], [97]	AR
[2], [72]	ARX
[3], [30], [86]	ARMA
[12], [21]	ARIMA
[57], [68], [70]	Kalman Filter Model
[18], [33], [45]	
[38], [105]	
[67]	The Unscented Kalman Filter
[56], [68], [87]	The State-space Model
[59], [95]	Interpolation
[20], [73], [83]	To Estimate the Parameters of Model

and for multi-measurement systems, r is a matrix, $p(k)$ and $q(k)$ represent the noises of the process and measurement, respectively, and they are assumed to be white Gaussian Noise.

The extended Kalman filter is the most widely used estimation for a recursive nonlinear system because it simply linearizes the nonlinear system models. However, the extended Kalman filter has two drawbacks: linearization can produce unstable filters and it is hard to implement the derivation of the Jacobian matrices. Thereby, Ma and Teng [67] present a new method of predicting the Mackey-Glass equation based on the unscented Kalman filter to solve these problems. In the field of signal processing, there are many works [18], [33], [45] based on Kalman filtering, but these techniques have not been widely used in the field of time series cleaning. Gardner [38] propose a new model, which is based on the Kernel Kalman Filter, to perform various nonlinear time series processing. Zhuang *et al.* [105] use the Kalman filter model to predict sensor data and smoothed it with WMA.

D. SUMMARY AND DISCUSSION

As shown in Table 2, there are many methods based on smoothing, such as the state-space model [56], [68], [87] and Interpolation [59], [95]. The state-space model assumes that the system's change over time can be determined by an unobservable vector sequence, the relationship between the time series and the observable sequence can be determined by the state-space model. By establishing state equations and observation equations, the state-space model provides a model framework to fully describe the temporal characteristics of dynamic systems. To make this kind of smoothing algorithm have a better effect, many studies [20], [73], [83] have also proposed various techniques to estimate the parameters in the above methods. Most smoothing techniques, when cleaning time series, have a small-time overhead, but it is very easy to change the original correct data points, which greatly affects the accuracy of cleaning. In other words, correct data are altered, which can distort the results of the analysis and lead to uncertainty in the results.

III. CONSTRAINT BASED CLEANING ALGORITHM

In this section, we introduce several typical algorithms, which include order dependencies (ODs) [32], sequential dependencies (SDs) [46] and speed constraints [82], for repairing time series errors.

A. ORDER DEPENDENCIES

In relational databases, Order Dependencies (ODs) are simple and effective methods, which have been widely studied [32], [41]–[43]. We find that ODs are also suitable for solving some time series data cleaning problems. The specific explanation is as follows: Let $x(t) = x_1, x_2 \ldots x_t$ be a time series, ODs can be expressed by $<, \leq, >, \geq$. For the number of miles traveled by the car $x(t)$, the mileage should increase over time. Formal representation is as follows: $t_1 < t_2$ then $x_{t_1} \leq x_{t_2}$ where $x(t)$ is mileage, t is timestamp. For instance, consider an example relation instance in Table 3. The tuples are sorted on attribute sequence number, which identifies sea level that rapidly increase from hour to hour.

TABLE 3. An example of order dependencies.

	sequence number	time
t_1	1	1
t_2	2	4
t_3	3	10
t_4	4	13
t_5	5	17
t_6	6	21
t_7	7	23

Generally, ODs in the form of equation (9) states that N is strictly increasing with M. Such as equation (10).

$$M \rightarrow_{(0,\infty)} N \tag{9}$$

$$\text{hour} \rightarrow_{(0,\infty)} \text{height} \tag{10}$$

ODs and DCs can also be used as an integrity constraint for error detection and data repairing in databases. Wijsen [92], [93] extends ODs with a time dimension for temporal databases. Let $I = \{I_1, I_2, I_3, \ldots\}$ be a temporal relation, which can be viewed as a time series of conventional "snapshot" relations, all over the same set of attributes. A trend dependency (TD) allows attributes with linearly ordered domains to be compared over time by using any operator of $\{<, =, >, \leq, \geq, \neq\}$. Consider the constraint is specified over (I_i, I_{i+1}) in I. For each time point i, it requires comparing employee records at time i with records at the next time $i + 1$, such that salaries of employees should never decrease. Lopatenko and Bravo [65] propose a numerical type data cleaning method based on Denial Constraints (DCs) as constraints, whose principle is similar to this one.

B. SEQUENTIAL DEPENDENCIES

The sequential dependency algorithm proposed by Golab *et al.* [46] focuses on the difference in values between two consecutive data points in a time series. Golab *et al.* [46]

define the CSD Tableau Discovery Problem as given a relation instance and an embedded SD $M \rightarrow_g N$, to find a tableau t_r of minimum size such that the CSD $(M \rightarrow_g N, t_r)$ has confidence at least a given threshold. A CSD can be

$$(\mathsf{hour} \rightarrow_{(0,\infty)} \mathsf{height}, [1961.01.01\ 00{:}00\text{–}2016.01.01\ 00{:}00]).$$

It states that for any two consecutive hours in [1961.01.01 00:00–2016.01.01 00:00], their distance should always be > 0.

Generally, a sequential dependency (SD) is in the form of

$$M \rightarrow_g N. \tag{11}$$

In equation (11), $M \subseteq R$ are ordered attributes, $N \subseteq R$ can be measured by certain distance metrics, and g is an interval. It states that when tuples are sorted on M, the distance between the N-values of any two consecutive tuples are within interval g. Casado-Vara *et al.* [19] propose the concept of streaming mode to represent the structure and semantic constraints of data streams. The concept contains a variety of semantic information, including not only numeric values, but also attributes between order. The sequential dependency algorithm can be used not only for traditional relational database cleaning, but also for time series cleaning. In fact, there are many dependency-based cleaning algorithms designed for relational databases that are not suitable for time series data cleaning, such as: Functional Dependencies [6] (FDs) and Conditional Functional Dependencies [36], [37] (CFDs). The sequential dependency is one of the few algorithms based on dependency that can be used for time series cleaning.

C. SPEED CONSTRAINTS

To clean time series data, speed constraint-based method proposed by Song *et al.* [82] consider the restrictions of speed on value changes in a given interval. As we have learned some common sense, e.g., the maximum flying speed of a bird, temperatures in a day, car mileage, etc. Consider with time window T is a pair of minimum speed $S_{\min}$ and maximum speed $S_{\max}$ over the time series $x = x_1, x_2, \ldots, x_t$, where each x_i is the value of the i-th data point, with a timestamp i.

For instance, consider time series:

$$x(t) = \{150, 160, 170, 180, 110, 200, 210, 220, 230\}$$

where timestamps $t = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The value attribute corresponds to x, while the time attribute in Table 4 denotes the timestamps.

Suppose a window size $T = 2$, $S_{\min} = -50$, and $S_{\max} = 50$ in the speed constraints, for data points x_5 and x_4, $\frac{110-180}{5-4} = -80 < -50$. Similarly, x_5 and x_6 with speed $\frac{200-110}{6-5} = 90 > 50$ are violations to $s_{\max} = 50$. To remedy the violations (denoted by red lines), a repair on x_5 can be performed, i.e., $x_5' = 190$, which is represented by the blue "$*$" symbol in Figure 3. As illustrated in Figure 3, the repaired sequence satisfies both the maximum and minimum speed constraints.

TABLE 4. An example relation instance of time series.

	time	value
t_1	1	150
t_2	2	160
t_3	3	170
t_4	4	180
t_5	5	110
t_6	6	200
t_7	7	210
t_8	8	220
t_9	9	230

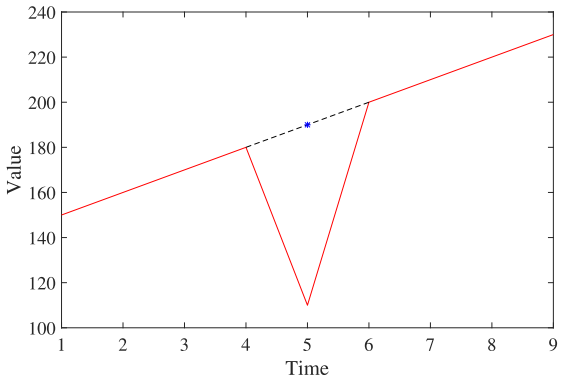

FIGURE 3. An example of speed constraints.

Generally, a speed constraint is in the form of equation (12).

$$S = (S_{\min}, S_{\max}) \tag{12}$$

If time series data x satisfies the speed constraint S, then for any x_i, x_j in a time window T, it has $S_{\min} < \frac{x_j - x_i}{j-i} < S_{\max}$. In practical applications, speed constraints are often valid for a specific period of time. For instance, when considering the fastest speed of a car, the time period of the reference is often in hours, and two data points in different years are not considered. The value of the speed constraints S may be positive (the growth rate of the constraint value) or negative (the rate of decline of the constraint value). Speed constraints are less effective when dealing with small errors, and Yin *et al.* [66] propose a further study of variance constraints, which use the variance threshold V to measure the degree of dispersion of the time series in a given W window.

D. SUMMARY AND DISCUSSION

In the field of relational databases, there are many cleaning algorithms based on integrity constraints, which are difficult to apply in the field of time series in which the observed values are substantially numerical, because they follow a strict equality relationship. A few methods, which we summarize in Table (5), can be used for time series data cleaning, for instance ODs and SDs can be used to solve problems in some scenarios, such as the number of miles in a car is non-decreasing. Further speed-based constraints can be used to process data such as GPS and stock prices, but only with

TABLE 5. Summary of constraints.

Reference	Method
[32], [41]–[43]	ODs
[92], [93]	Extend ODs
[65]	DCs
[46]	SDs
[6]	FDs
[36], [37]	CFDs
[82]	Speed Constraints
[66]	Variance Constraints
[119]	Similarity Rule Constraints
[120]	Learning Individual Models

relevant domain knowledge can give a reasonable constraint. Therefore, the constraint-based cleaning algorithm needs to be further improved to have better robustness. The Similarity Rule Constraints proposed by Song *et al.* [119] and the Learning Individual Models proposed by Zhang *et al.* [120] are suitable for repairing missing data. One possible future direction is to use anomaly detection methods to detect anomalies first, and then treat outliers as missing to repair. We will discuss anomaly detection in Section V.

IV. STATISTICS BASED CLEANING ALGORITHM

Statistical-based cleaning algorithms occupy an important position in the field of data cleaning. Such algorithms use models, which learned from data, to clean data. The statistical-based approach involves a lot of statistical knowledge, but this article focuses on statistical-based data cleaning methods, so we won't cover statistical-related knowledge in detail.

A. MAXIMUM LIKELIHOOD

The intuitive idea of the maximum likelihood principle is a random test, if there are several possible outcomes $x_1, x_2 \ldots x_t$, if the result x_i occurs in one test, it is generally considered that the test conditions are favorable for x_i, or think that x_i has the highest probability of occurrence.

Notation: For a given time series data $x(t)$, which consistents with a probability distribution d, and assume that its probability aggregation function (discrete distribution) is F_d; consider a distribution parameter θ, sampling $x_1, x_2 \ldots x_n$ from this distribution, then use F_d to calculate its probability [13] as shown in equation (13).

$$P = (x_1, x_2 \ldots x_n) = F_d(x_1, x_2 \ldots x_n|\theta) \tag{13}$$

Gogacz and S. Toruńczyk [44] use the maximum likelihood technique to clean Radio Frequency Identification (RFID) data. Wang *et al.* [89] propose the first maximum likelihood solution to address the challenge of truth discovery from noisy social sensing data. Yakout *et al.* [96] argue a new data repairing approach that is based on maximizing the likelihood of replacement data in the given data distribution, which can be modeled using statistical machine learning techniques, but this technology is used to repair the data of the database. For the repairing of time series data errors, Zhang *et al.* [99] propose a better solution based on maximum likelihood, which solves the problem from the perspective of probability. According to the probability distribution of the speed change of adjacent data points in the time series, the time series cleaning problem can be converted to find a cleaned time series, which is based on the probability of speed change that has the greatest likelihood.

B. MARKOV MODEL

Markov process is a class of stochastic processes, which means that the transition of each state in the process depends only on the previous n states. This process is called a $n-order$ model, where n is the number that affects the transition state. The simplest Markov process is the $first-order$ process, and the transition of each state depends only on the state before it. Time and state are discrete Markov processes called Markov chains, abbreviated as $X_n = X(n), n = 0, 1, 2 \ldots$. The Markov chain [112] is a sequence of random variables $X_1, X_2, X_3 \ldots$. The range of these variables, that is, the set of all their possible values, is called the "state space", and the value of X_n is the state of time n.

The Markov Model [113], [114] is a statistical model based on Markov chain, which is widely used in speech recognition, part-of-speech automatic annotation, phonetic conversion, probabilistic grammar and other natural language processing applications. In order to find patterns that change over time, the Markov model attempts to build a process model that can generate patterns. References [114] and [113] use specific time steps, states, and make Markov assumptions. With these assumptions, this ability to generate a pattern system is a Markov process. A Markov process consists of an initial vector and a state transition matrix. One thing to note about this assumption is that the state transition probability does not change over time.

Hidden Markov Model (HMM) [116], [117] is a statistical model based on Markov Model, which is used to describe a Markov process with implicit unknown parameters. The difficulty is to determine the implicit parameters of the process from observable parameters, and then use these parameters for further analysis, such as prediction of time series data. For instance, after rolling the dice 10 times, we could get a string of numbers, for example we might get such a string of numbers:1, 4, 5, 3, 3, 1, 6, 2, 4, 5 as shown in Figure 4. This string of numbers is called the visible state chain. But in HMM, we not only have such a string of visible state chains, but also a chain of implied state chains. In this example, the implicit state chain might be: $D5, D3, D2, D3, D4, D6, D1, D5, D1, D2$.

Gupta and Dhingra [118] use HMM to predict the price of stocks. Baba *et al.* [4] argue a data cleaning method based on the HMM, which used to clean RFID data related to geographic location information. In multi-dimensional time series cleaning, HMM has more application space than the single-dimensional cleaning algorithm, because of the correlation between the dimensions.

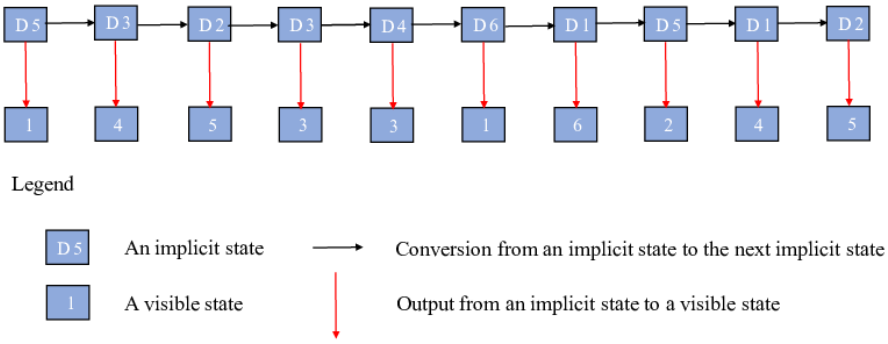

FIGURE 4. An example of hidden markov.

C. BINOMIAL SAMPLING

Jeffery *et al.* [54] propose an adaptive method for cleaning RFID data, which exploits techniques based on sampling and smoothing theory to improve the quality of RFID data. Tag transition detection: tag transition detection refers to the fact that when the position of the tag changes, the cleaning result should reflect the fact that the tag leaves. Let's introduce a few RFID-related concepts.

(1) Interrogation cycles: the reader's question-and-answer process for the tag is the basic unit of the reader's detection tag.

(2) Reader read cycle (epoch, 0.2 sec - 0.25 sec): a collection of multiple interrogation cycles.

Based on the above concept, the following definition: W_i is smooth window of tag i and is composed of ω_i epoch, S_i is the window that tag i is actually detected in the W_i window, $Count_t$ indicates the number of inquiry cycles of t, and R is the corresponding number of t epoch tag i. For a given time window, suppose the probability that the tag i may be read in each epoch is $p_i = \frac{R}{Count_t}$, and the Statistical Soothing for Unreliable RFID Data (SMURF) [54] algorithm treats each epoch's reading of the tag as a Bernoulli experiment with probability p_i. Therefore, p_i conforms to the binomial distribution $B(\omega_i, p_i)$. $p_{i,avg}$ is the average read rate in S_i.

Using the model based on the Bernoulli experiment to observe the tag i, if the average reading rate of the tags in ω_i epoch is $(1-p_{i,avg})^{\omega_i}$. To ensure the dynamic nature of the tag the size of the sliding window W_i needs to be satisfied as shown in equation (14).

$$||S_i| - \omega_i p_{i,avg}| > 2\sqrt{\omega_i p_{i,avg}(1-p_{i,avg})} \tag{14}$$

The SMURF algorithm first sets the initial window size to 1, and then dynamically adjusts the window length based on the actual situation of the read. If the current window meets the integrity requirement [54], the SMURF algorithm will detect the status of the tag. When the detection result indicates that the tag status changes, SMURF will adjust the current window length to 1/2 of the original window to react to the tag's transition. If the calculated window size that satisfies the integrity constraint is greater than the current window size, the algorithm linearly increases the current window size by 2 steps and outputs the point data in the current window. If it is detected that the label does not move, the algorithm outputs the current window midpoint as the output point, and then continues to slide an epoch for the next processing.

SMURF algorithm is widely used to clean RFID data, and many studies [27], [115] improve it. Leema *et al.* [27] study the effect of tag movement speed on data removal results and Xu *et al.* [115] consider the impact of data redundancy on setting up sliding windows.

D. SPATIO-TEMPORAL PROBABILISTIC MODEL

Besides data cleaning, Milani *et al.* [124] propose Spatio-Temporal Probabilistic Model (STPM), this method learns more detailed data patterns from historical data, and then cleans the current data. STPM not only gives joint probability distributions that are updated on the data set at different times, but also distinguishes association updates from association values. STPM based on Dynamic Probabilistic Relational Models (DRPMs), so we need to state DRPMs model first. The DRPMs is a graph model used to represent the relationship between dynamic data sets, its models based on the dependency relationship between attributes, and generally uses conditional probability distribution to calculate the probability of each attribute value in a given parent node value and forms a relationship chain. For instance, when we need to estimate the data at time T, we can only use the data before time T to infer, namely, the current state depends only on the previous state, which is similar to the Markov Model. STPM extends DRPMs to model update pattern between different time data, and captures spatial and temporal update patterns by modeling updates events to provide update relationships of possible existence, finally detect and repair data.

E. OTHERS

Firstly, we summarize the methods described above in Table (6). In fact, Bayesian prediction model is a technique based on Bayesian statistics. The Bayesian prediction model utilizes model information, data information, and prior information, so the prediction effect is good, there this model is widely used, including in the field of time series data cleaning. Wang *et al.* [90] establish a cost model

TABLE 6. Summary of statistics.

Reference	Method
[13], [44], [89] [96], [99]	Maximum Likelihood
[112]–[114]	Markov Model
[116]–[118]	HMM
[4]	
[27], [54], [115]	SMURF
[124]	STPM
[39], [90]	Bayesian
[7], [69], [71]	RDN
[85]	GMM
[79]	EM

for Bayesian analysis which is used to analyze errors in the data. Bergman *et al.* [7] consider the user's participation and use the user's feedback on the query results to clean the data. Mayfield *et al.* [69] propose a more complex relationship-dependent network (RDN [71]) model to model the probability relationships between attributes. The difference between RDN and traditional relational dependencies (such as Bayesian networks [39]) is that RDNs can contain ring structures. The method iteratively cleans the data set and observes the change in the probability distribution of the data set before and after each wash. When the probability distribution of the data set converges, the cleaning process is aborted. Zhou and Tung [104] argue a technique for accelerating the learning of Gaussian models via using GPU. The article believes that in the case of excessive data, it is not necessary to use all the data to learn the model. Also, the author provides a method of automatic tuning. In order to clean and repair fuel level data, Tian *et al.* [85] propose a modified Gaussian mixture model (GMM) based on the synchronous iteration method, which uses the particle swarm optimization algorithm and the steepest descent algorithm to optimize the parameters of GMM and uses linear interpolation-based algorithm to correct data errors. Shumway and Stoffer [79] use the EM [28] algorithm combined with the spatial state model [56], [70] to predict and smooth the time series.

V. TIME SERIES ANOMALY DETECTION

Gupta *et al.* [48] investigate the anomaly detection methods for time series data: for a given time series data, there may be two types of outliers, namely single-point anomalies and subsequence anomalies (continuous anomalies). In this section, we first discuss the detection methods of abnormal points and abnormal sequences, next introduce the application of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm in data cleaning, and then review the abnormal detection methods related to machine learning.

A. ABNORMAL POINT DETECTION

For single-point anomalies, the most common idea is to use predictive models for detection. That is, the predicted value of the established model and the observed value for each data point is compared, and if the difference between the two values is greater than a certain threshold, the observed value is considered to be an abnormal value. Specifically, Basu and Meckesheimer [5] select all data points with timestamps $t-k$ to $t+k$ with the timestamp t as the center point, and the median of these data points is considered to be the predicted value of data points with timestamp t value. Hill and Minsker [51] first cluster the data points and take the average of the clusters as the predicted value of the point. The AR model and the ARX model are widely used for anomaly detection in various fields, such as economics, social surveys [11], [16], and so on. The ARX model takes advantage of manually labeled information, so it is more accurate than the AR model when cleaning data. The ARIMA model [102] represents a type of time series model consisting of AR and MA mentioned above, which can be used for data cleaning of non-stationary time series. Kontaki *et al.* [63] propose continuous monitoring of distance-based outliers over data streams. One of the most widely used definitions is the one based on distance as shown in Figure 5: an object p is marked as an outlier, if there are less than k objects in given distance. Here $k=4$, q is the normal point and p is the abnormal point.

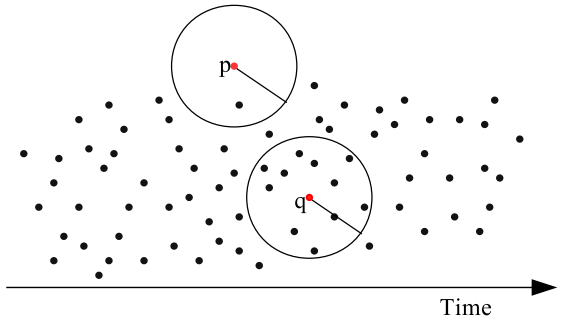

FIGURE 5. An example abnormal point detection.

B. ABNORMAL SEQUENCE DETECTION

Different studies have different definitions of subsequence anomalies. Keogh *et al.* [61] proposed that a subsequence anomaly, that is, a subsequence has the largest distance from its nearest non-overlapping match. With this definition, the simplest calculation method is to calculate the distance between each subsequence with length n and other subsequences. Of course, the time complexity of this calculation method is very high. In the later studies, Keogh *et al.* [60] propose a heuristic algorithm by reordering candidate subsequences and Wei *et al.* [91] argue an acceleration algorithm using local sensitive hash values. In calculating the distance, the Euclidean distance is usually used, and Keogh *et al.* [62] further proposes a method using the compression-based similarity measure as the distance function. As shown in Figure 6, the data is divided into multiple sub-sequences that overlap each other. First, calculate the abnormal score of each window, and then calculate the abnormal score (AS) of the whole test sequence according to the abnormal score (AS) of each window. Window-based techniques can better locate

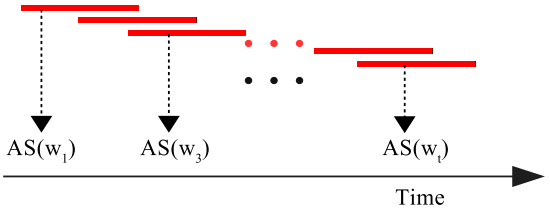

FIGURE 6. An example abnormal sequence detection.

anomalies compared to direct output of the entire time series as outliers. There are two main types of methods based on this technique. One is to maintain a normal database [34], [40], and then compare the test sequence with the sequence in the normal database to determine whether it is abnormal; the other is to build an anomalous database [24], [47] and then compare the test sequence with the sequence in the database to detect if it is anomalous.

C. DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE

The DBSCAN [111] algorithm is a clustering method based on density-reachable relationship, which divides the region with sufficient density into clusters and finds clusters of arbitrary shape in the spatial database with noise and defines the cluster as the largest set of points connected by density. Then the algorithm defines the cluster according to the set density threshold as the basis for dividing the cluster, that is, when the threshold is satisfied, it can be considered as a cluster.

The principle of DBSCAN algorithm: (1) DBSCAN searches for clusters by checking the *Eps* neighborhood of each point in the data set. If the *Eps* neighborhood of point *p* contains more points than *MinPts*, create a cluster with *p* as the core object; (2) Then, DBSCAN iteratively aggregates objects that are directly reachable from these core objects. This process may involve the consolidation of some density-reachable clusters; (3) When no new points are added to any cluster, the process ends.

Where *MinPts* is the minimum number of neighbor points that a given point becomes the core object in the neighborhood, *Eps* is the neighborhood radius. For instance, *Eps* is 0.5 and *MinPts* is 3, for a given data set, the effect of clustering is as shown in Figure 7. Some noise points can be repaired and clustered into classes adjacent to them. Recent research [81] has shown that after repairing erroneous data. They also perform cleaning experiments on GPS data based on DBSCAN, the accuracy of clustering on spatial data can be improved. But this method cannot solve continuous errors and needs further improvement.

D. GENERATIVE ADVERSARIAL NETWORKS

With the rapid development of machine learning technology, more and more problems are solved using machine learning. Li *et al.* [76] use the GANs network to effectively detect anomalies in time series data. GANs trains two models at the same time, which are the generation model for capturing

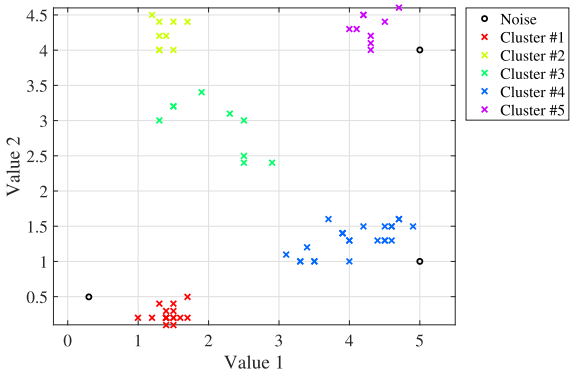

FIGURE 7. An example of DBSCAN clustering.

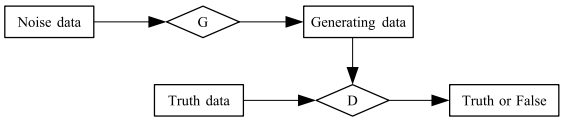

FIGURE 8. A simple flow chart of generative adversarial networks.

data distribution and the discriminant model for discriminating whether the data are real data or pseudo data as shown in Figure 8.

Given a random variable with a probability of uniform distribution as input, we want to generate a probability distribution of the output as "dog probability distribution". The philosophy of Generative Matching Networks (GMNs), which idea is to train the generative network by directly comparing the generated distribution with the true distribution, is to optimize the network by repeating the following steps:

(1) Generate some evenly distributed input;

(2) Let these inputs go through the network and collect the generated output;

(3) Compare the true "dog probability distribution" with the generated "dog probability distribution" based on the available samples (e.g. calculate the MMD distance between the real dog image sample and the generated image sample);

(4) Use backpropagation and gradient descent to compute the errors and update the weights. The purpose of this process is to minimize the loss of the generation model and discriminant.

Li *et al.* [76] use GANs to detect abnormalities in time series and a natural idea is to use GANs network to repair missing values of time series data. Perhaps more machine learning algorithms are waiting for the cleaning of time series error values. A simple idea is to treat the detected anomaly data as missing data and then repair it. Sun *et al.* [121] first analyze the similarity between parking space data and parking data, and then use Recurrent GANs to generate parking data as repair data, which provide a new idea for solving the problem of time series data repair. Fang *et al.* [122] propose FuelNet which is based on Convolutional Neural Networks (CNNs) and GANs. FuelNet is used to repair the inconsistent and impute the incomplete fuel consumption rates over time.

E. LONG SHORT-TERM MEMORY

Since Recurrent Neural Network (RNN) also has the problem of gradient disappearance, it is difficult to process long-sequence data. Gers *et al.* [106] improve RNN and got the RNN special case Long Short-Term Memory (LSTM), which can avoid the disappearance of the regular RNN gradient. It has been widely used in industry and [107]–[109] use LSTM to perform anomaly detection on time series data.

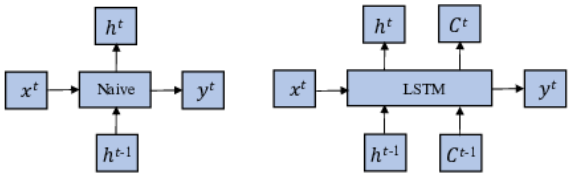

FIGURE 9. Simple RNN structure and simple LSTM structure.

The left picture is a simple RNN structure diagram, and the right picture is a simple LSTM structure diagram in Figure 9, where given function as shown in equation (15).

$$F : h, y = f(h, x) \tag{15}$$

In equation (15), x^t is the input of data in the current state, h^{t-1} (hidden state) indicates the input of the previous node received, y^t is the output in the current state and h^t is the output passed to the next node. As can be seen from the Figure 9, the output h^t is related to the values of x^t and h^{t-1}. y^t is often used to invest in a linear layer (mainly for dimension mapping), and then use *softmax* to classify the required data. As shown in Figure 9, RNN has only one delivery state h^t, LSTM also has a delivery status c^t (cell state). There are three main stages within LSTM:

(1) Forgotten phase. The forgetting phase is mainly to forget the input that is passed in from the previous node. A simple principle is: forget the unimportant, remember the important one. More specifically, z^f is calculated as a forgotten gate, which is used to control the previous state c^{t-1}, and then decide whether to retain the data or forget it.

(2) Selective memory phase. At this stage, the input is selectively memorized. Mainly to remember the input x, the more important the data needs to be more reserved.

(3) Output phase. This phase determines which outputs would be treated as current states. Similar to the normal RNN, the output y^t is often also obtained by h^t change.

Filonov *et al.* [107] and Pankaj *et al.* [110] provide recurrent neural networks by providing network time series data. The recurrent neural network understands what the normal expected network activity is. When an unfamiliar activity from the network is provided to a trained network, it can distinguish whether the activity is expected or invaded.

F. SUMMARY AND DISCUSSION

In addition to the methods described above, we also summarize some common methods in Table (7). As shown in Table (7), Xing *et al.* [94] show that the cleaned sequence can improve the accuracy of time series classification. Diao *et al.* [29] design LOF [14] based online anomaly detection and cleaning algorithm. Zhang *et al.* [100] propose an iterative minimum cleaning algorithm based on the timing correlation of error time series in continuous errors and keep the principle of minimum modification in data cleaning. The algorithm is effective in cleaning continuous errors in time series data. Qu *et al.* [74] first use cluster-based methods for anomaly detection and then use exponentially weighted averaging for data repair, which is used to clean power data in a distributed environment. Corizzo *et al.* [123] use detect anomalous geographic data by distance-based method, and then use Gradient-boosted tree (GBT) to repair the anomalous data. We can conclude that anomaly detection algorithms play an important role in time series data cleaning. It is also becoming more and more important to design anomaly detection algorithms for time series repair, and we discuss future directions in Section VII.

TABLE 7. Summary of detection.

Reference	Method
[5], [60], [61]	Distance-based
[29], [123]	
[34], [40]	Maintain a Normal Database
[24], [47]	Build an Anomalous Database
[74], [81], [111]	Clustering
[76], [121], [122]	GANs
[107]–[109]	LSTM

VI. TOOLS AND EVALUATION CRITERIA

In this section, we first give an overview of tools to clean time series and then summarize evaluation criteria related to time series cleaning methods.

A. TOOLS

There are many tools or systems for data cleaning, but they are not effective on time series cleaning problems. In Table 8 we investigate some tools that might be used for time series cleaning because they [64], [75], [84], [98] are originally used to solve traditional database cleaning problems. Ding *et al.* [31] present Cleanits, which is an industrial time series cleaning system and implements an integrated cleaning strategy for detecting and repairing in industrial time series. Cleanits provides a user-friendly interface so users can use results and logging visualization over every cleaning process. Besides, the algorithm design of Cleanits also considers the characteristics of industrial time series and domain knowledge. The ASPA proposed by Rong and Bailis [77] violates the principle of minimum modification and distort the data, which is not suitable for being used widely. EDCleaner proposed by Wang *et al.* [125] is designed for social network data, detection and cleaning are performed through the characteristics of statistical data fields. Huang *et al.* [126] propose PACAS which is a framework for data cleaning between service providers and customers. Huang *et al.* [127] present TsOutlier, a new framework for detecting outliers with explanations over IoT data. TsOutlier uses multiple algorithms to

TABLE 8. Some examples of tools or systems.

	Method	Detail
PIClean [98]	Based on statistics	Produce probabilistic errors and probabilistic fixes using low-rank approximation, which implicitly discovers and uses relationships between columns of a dataset for cleaning.
HoloClean [75]	Based on statistics	Learn the probability model and then select the final data cleaning plan based on the probability distribution.
ActiveClean [64]	Based on statistics	Allow for progressive and iterative cleaning in statistical modeling problems while preserving convergence guarantees.
Cleanits [31]	Anomaly detection	Develop reliable data cleaning algorithms by considering features of both industrial time series and domain knowledge.
MLClean [84]	Anomaly detection	The combination of data cleaning technology and machine learning methods is designed to generate unbiased cleaning data, which is used to train accurate models.
ASAP [77]	Smoothing based	Develop a new analytics operator called ASAP that automatically smooths streaming time series by adaptively optimizing the trade-off between noise reduction and trend retention.
EDCleaner [125]	Based on statistics	For social network data, detection and cleaning are performed through the characteristics of statistical data fields.
PACAS [126]	Based on statistics	Design a framework for data cleaning between service providers and customers.
TsOutlier [127]	Anomaly detection	Use multiple algorithms to detect anomalies in time series data, and support both batch and streaming processing.

detect anomalies in time series data, and supports both batch and streaming processing. There is not much research on time series cleaning tools or systems, and we discuss further in Future Directions in Section VII.

B. EVALUATION CRITERIA

The Root Mean Square (RMS) error [54] is used to evaluate the effectiveness of the cleaning algorithm. Let x denotes the sequence consisting of the true values of the time series, $\overline{x}$ denotes the sequence consisting of the observations after the error is added, and $\widehat{x}$ denotes the sequence consisting of the repaired values after the cleaning. Here the RMS error [54] is represented as shown in equation (16).

$$\Delta(x, \widehat{x}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{x}_i)^2} \tag{16}$$

The equation (16) measures the distance between the true value and the cleaned value. The smaller the RMS error, the better the cleaning effect.

Other criteria include error distance between incorrect data and correct data, repaired distance between erroneous data and cleaned results (as shown in equation (17) referring to the minimum modification principle in data repairing).

$$\Delta(\overline{x}, \widehat{x}) = \sum_{i=1}^{n} |\overline{x_i} - \widehat{x}_i| \tag{17}$$

Dasu and Loh [26] propose a statistical distortion method to evaluate the quality of cleaning methods. The proposed method directly observes the numerical distribution in the data set and evaluates the quality according to the variation of the distribution caused by different cleaning methods.

VII. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we review four types of time series cleaning algorithms, cleaning tools or systems and related research on evaluation criteria. Next, we summarize the full text in Section VII-A and list some advice of future directions in Section VII-B.

A. CONCLUSION

With the development of technology, people gradually realize the value contained in the data. Owing to companies want to derive valuable knowledge from these data, and data analysis has played an increasingly important role in finance, healthcare, natural sciences, and industry. Time series data, as an important data type, is widely found in industrial manufacturing. For instance, a wind power enterprise analyzes sensor data, which are located throughout the wind turbine, to determine whether the fan is in a normal state; transport companies also want to optimize vehicle fleet travel by analyzing vehicle GPS information. However, due to external environmental interference, sensor accuracy, and other issues, time series data often contain many errors that can interfere with subsequent data analysis and cause unpredictable effects.

B. FUTURE DIRECTIONS

As mentioned above, data is an intangible asset and advanced technology helps to fully exploit the potential value of data. Thereby, time series data cleaning methods provide very important technical support for the discovery of these values in processing time series error data. Next we list some advice of future directions based on [101].

The error type illustrates handbook of time series data. At present, data scientists have a very detailed analysis of the errors in the traditional relational database. However, there is still much work to be further studied in the analysis of

time series data error types. For instance, this paper roughly divides the types of time series errors into three types, namely single point big errors, single point small errors and continuous errors. In fact, in continuous errors, there are also a lot of meticulous types of errors, such as additive errors and the innovational errors [86]. How to systematically analyze these error types and form time series data error type illustrated handbook is very important. The clear error type helps to develop targeted cleaning algorithms to solve the problem of ''GIGO (Garbage in, garbage out.)'' that exists in the current field.

The design of time series data cleaning algorithm. Each chapter of this paper reviews some time series error cleaning algorithms, but further optimizations are possible. The existing methods are mostly for a single-dimensional time series (even the GPS data exists two dimensions' information), but each dimension is cleaned separately during cleaning [82], [99], [100]. To further improve the practicability of the algorithm, it is imperative to consider the cleaning of multidimensional time series. Besides, with the development of machine learning technology, more technical learning techniques should be considered for data cleaning algorithms, which may lead to better cleaning results because of the mathematical support behind them.

The implementation of time series cleaning tool. At present, the mainstream data cleaning tools in the industry are still aimed at relational databases, and these tools are not ideal for processing time series data. As time series data cleaning problems become more serious, how to use the fast-developing distributed technology, high performance computing technology and stream processing technology to implement time series cleaning tools (including research tools and commercial tools) and apply them to real-world scenarios such as industry is also the key work of the next stage.

The algorithm design of time series anomaly detection. In real-world scenarios, efficient anomaly detection algorithms play an irreplaceable role in time series repair. It is difficult to judge the difference between the error value and the true value, so it is necessary to specifically design a time series anomaly detection algorithm that can be applied to an industrial scene. It is worth noting that more research is needed on how to perform anomaly detection, cleaning, and analysis in the case of weak domain knowledge or less labeled data.

Design of data cleaning algorithms for specific application scenarios. With the application of various technologies in the industry, the application scenarios are becoming more and more clear. The requirements for cleaning algorithms in different application scenarios have different focuses. For instance, the data stored in the Blockchain network [19] are generally structured data, with the development of Blockchain technology, the design of data cleaning algorithms on Blockchain networks is also particularly important.

REFERENCES

[1] F. N. Afrati and P. G. Kolaitis, ''Repair checking in inconsistent databases: Algorithms and complexity,'' in *Proc. 12th Int. Conf. Database Theory (ICDT)*, Saint Petersburg, Russia, Mar. 2009, pp. 31–41.
[2] H. N. Akouemo and R. J. Povinelli, ''Data improving in time series using ARX and ANN models,'' *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3352–3359, Sep. 2017.
[3] G. Alengrin and G. Favier, ''New stochastic realization algorithms for identification of ARMA models,'' in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Tulsa, OK, USA, Apr. 1978, pp. 208–213.
[4] A. I. Baba, M. Jaeger, H. Lu, T. B. Pedersen, W. Ku, and X. Xie, ''Learning-based cleansing for indoor RFID data,'' in *Proc. Int. Conf. Manage. Data SIGMOD Conf.*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 925–936.
[5] S. Basu and M. Meckesheimer, ''Automatic outlier detection for time series: An application to sensor data,'' *Knowl. Inf. Syst.*, vol. 11, no. 2, pp. 137–154, Feb. 2007.
[6] C. Beeri, M. Dowd, R. Fagin, and R. Statman, ''On the structure of armstrong relations for functional dependencies,'' *J. ACM*, vol. 31, no. 1, pp. 30–46, 1984.
[7] M. Bergman, T. Milo, S. Novgorodov, and W. C. Tan, ''Query-oriented data cleaning with oracles,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Melbourne, VIC, Australia, May/Jun. 2015, pp. 1199–1214.
[8] L. E. Bertossi, L. Bravo, E. Franconi, and A. Lopatenko, ''Complexity and approximation of fixing numerical attributes in databases under integrity constraints,'' in *Database Programming Languages*, vol. 3774. Berlin, Germany: Springer, 2005, pp. 262–278.
[9] L. Bertossi, L. Bravo, E. Franconi, and A. Lopatenko, ''The complexity and approximation of fixing numerical attributes in databases under integrity constraints,'' *Inf. Syst.*, vol. 33, nos. 4–5, pp. 407–434, 2008.
[10] P. Bohannon, M. Flaster, W. Fan, and R. Rastogi, ''A cost-based model and effective heuristic for repairing constraints by value modification,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Baltimore, MD, USA, Jun. 2005, pp. 143–154.
[11] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
[12] G. E. P. Box and D. A. Pierce, ''Distribution of residual autocorrelations in autoregressive-integrated moving average time series models,'' *J. Amer. Statist. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Apr. 1970.
[13] Y. Bresler and A. Macovski, ''Exact maximum likelihood parameter estimation of superimposed exponential signals in noise,'' *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1081–1089, Oct. 1986.
[14] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, ''LOF: Identifying density-based local outliers,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Dallas, TX, USA, May 2000, pp. 93–104.
[15] D. R. Brillinger, *Time Series: Data Analysis and Theory* (Classics in Applied Mathematics), vol. 36. Philadelphia, PA, USA: SIAM, 2001.
[16] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Basel, Switzerland: Springer, 2016.
[17] R. G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*. North Chelmsford, MA, USA: Courier Corporation, 2004.
[18] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, vol. 3. New York, NY, USA: Wiley, 1992.
[19] R. Casado-Vara, F. de la Prieta, J. Prieto, and J. M. Corchado, ''Blockchain framework for IoT data quality via edge computing,'' in *Proc. 1st Workshop Blockchain-Enabled Netw. Sensor Syst.*, Shenzhen, China, Nov. 2018, pp. 19–24.
[20] I. Chang, G. C. Tiao, and C. Chen, ''Estimation of time series parameters in the presence of outliers,'' *Technometrics*, vol. 30, no. 2, pp. 193–204, 1988.
[21] P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen, ''ARIMA-based time series model of stochastic wind power generation,'' *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 667–676, May 2010.
[22] J. Chomicki and J. Marcinkowski, ''Minimal-change integrity maintenance using tuple deletions,'' *Inf. Comput.*, vol. 197, nos. 1–2, pp. 90–121, 2005.
[23] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, ''Data cleaning: Overview and emerging challenges,'' in *Proc. Int. Conf. Manage. Data SIGMOD Conf.*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 2201–2206.
[24] D. Dasgupta and N. S. Majumdar, ''Anomaly detection in multidimensional data using negative selection algorithm,'' in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, May 2002, pp. 1039–1044.

[25] T. Dasu, R. Duan, and D. Srivastava, "Data quality for temporal streams," *IEEE Data Eng. Bull.*, vol. 39, no. 2, pp. 78–92, Jun. 2016.
[26] T. Dasu and J. M. Loh, "Statistical distortion: Consequences of data cleaning," *Proc. PVLDB*, vol. 5, no. 11, pp. 1674–1683, 2012.
[27] A. A. Leema and M. Hemalatha, "An effective and adaptive data cleaning technique for colossal RFID data sets in healthcare," *WSEAS Trans. Inf. Sci. Appl.*, vol. 8, no. 6, pp. 243–252, 2011.
[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc., B Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.
[29] Y. Diao, K. Liu, X. Meng, X. Ye, and K. He, "A big data online cleaning algorithm based on dynamic outlier detection," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Xi'an, China, Sep. 2015, pp. 230–234.
[30] S. Dilling and B. J. MacVicar, "Cleaning high-frequency velocity profile data with autoregressive moving average (ARMA) models," *Flow Meas. Instrum.*, vol. 54, pp. 68–81, Apr. 2017.
[31] X. Ding, H. Wang, J. Su, Z. Li, J. Li, and H. Gao, "Cleanits: A data cleaning system for industrial time series," *Proc. PVLDB*, vol. 12, no. 12, pp. 1786–1789, 2019.
[32] J. Dong and R. Hull, "Applying approximate order dependency to reduce indexing space," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Orlando, FL, USA, Jun. 1982, pp. 119–127.
[33] G. A. Einicke and L. B. White, "Robust extended Kalman filtering," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2596–2599, Sep. 1999.
[34] D. Endler, "Intrusion detection applying machine learning to solaris audit data," in *Proc. 14th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Scottsdale, AZ, USA, Dec. 1998, pp. 268–279.
[35] R. Fagin, B. Kimelfeld, and P. G. Kolaitis, "Dichotomies in the complexity of preferred repairs," in *Proc. 34th ACM Symp. Princ. Database Syst. (PODS)*, Melbourne, VIC, Australia, May/Jun. 2015, pp. 3–15.
[36] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis, "Conditional functional dependencies for capturing data inconsistencies," *ACM Trans. Database Syst.*, vol. 33, no. 2, p. 6, 2008.
[37] W. Fan, F. Geerts, J. Li, and M. Xiong, "Discovering conditional functional dependencies," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 5, pp. 683–698, May 2011.
[38] E. S. Gardner, Jr., "Exponential smoothing: The state of the art—Part II," *Int. J. Forecasting*, vol. 22, no. 4, pp. 637–666, 2006.
[39] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of relational structure," in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*. Williamstown, MA, USA: Williams College, Jun./Jul. 2001, pp. 170–177.
[40] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection," in *Proc. Workshop Intrusion Detection Netw. Monitor.*, Santa Clara, CA, USA, Apr. 1999, pp. 51–62.
[41] S. Ginsburg and R. Hull, "Order dependency in the relational model," *Theor. Comput. Sci.*, vol. 26, nos. 1–2, pp. 149–195, May 1983.
[42] S. Ginsburg and R. Hull, "Sort sets in the relational model," in *Proc. 2nd ACM SIGACT-SIGMOD Symp. Princ. Database Syst.*, Atlanta, GA, USA, Mar. 1983, pp. 332–339.
[43] S. Ginsburg and R. Hull, "Sort sets in the relational model," *J. ACM*, vol. 33, no. 3, pp. 465–488, 1986.
[44] T. Gogacz and S. Toruńczyk, "Entropy bounds for conjunctive queries with functional dependencies," in *Proc. 20th Int. Conf. Database Theory (ICDT)*, Venice, Italy, Mar. 2017, pp. 15:1–15:17.
[45] Z. Goh, K.-C. Tan, and B. T. G. Tan, "Kalman-filtering speech enhancement method based on a voiced-unvoiced speech model," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 5, pp. 510–524, Sep. 1999.
[46] L. Golab, H. J. Karloff, F. Korn, A. Saha, and D. Srivastava, "Sequential dependencies," *Proc. PVLDB*, vol. 2, no. 1, pp. 574–585, 2009.
[47] F. A. González and D. Dasgupta, "Anomaly detection using real-valued negative selection," *Genetic Program. Evolvable Mach.*, vol. 4, no. 4, pp. 383–403, 2003.
[48] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.
[49] J. D. Hamilton, *Time Series Analysis*, vol. 2. Princeton, NJ, USA: Princeton Univ. Press, 1994.
[50] J. M. Hellerstein, "Quantitative data cleaning for large databases," UNECE, Geneva, Switzerland, Tech. Rep. 2008-02-07-41, 2008.
[51] D. J. Hill and B. S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach," *Environ. Model. Softw.*, vol. 25, no. 9, pp. 1014–1022, 2010.
[52] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Int. J. Forecasting*, vol. 20, no. 1, pp. 5–10, 2004.
[53] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, "Declarative support for sensor data cleaning," in *Proc. 4th Int. Conf. Pervasive Comput.*, Dublin, Ireland, May 2006, pp. 83–100.
[54] S. R. Jeffery, M. N. Garofalakis, and M. J. Franklin, "Adaptive cleaning for RFID data streams," in *Proc. 32nd Int. Conf. Very Large Data Bases*, Seoul, South Korea, Sep. 2006, pp. 163–174.
[55] C. S. Jensen and R. T. Snodgrass, "Temporal data management," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 36–44, Jan. 1999.
[56] R. H. Jones, "Exponential smoothing for multivariate time series," *J. Roy. Stat. Soc., B Methodol.*, vol. 28, no. 1, pp. 241–251, 1966.
[57] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, D, J. Basic Eng.*, vol. 82, pp. 35–45, Mar. 1960.
[58] A. Karkouch, H. Mousannif, H. A. Moatassime, and T. Noël, "Data quality in Internet of Things: A state-of-the-art survey," *J. Netw. Comput. Appl.*, vol. 73, pp. 57–81, Sep. 2016.
[59] E. J. Keogh, S. Chu, D. M. Hart, and M. J. Pazzani, "An online algorithm for segmenting time series," in *Proc. IEEE Int. Conf. Data Mining*, San Jose, CA, USA, Nov./Dec. 2001, pp. 289–296.
[60] E. J. Keogh, J. Lin, and A. W. Fu, "HOT SAX: Efficiently finding the most unusual time series subsequence," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Houston, TX, USA, Nov. 2005, pp. 226–233.
[61] E. J. Keogh, J. Lin, S. Lee, and H. V. Herle, "Finding the most unusual time series subsequence: Algorithms and applications," *Knowl. Inf. Syst.*, vol. 11, no. 1, pp. 1–27, 2007.
[62] E. J. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Seattle, WA, USA, Aug. 2004, pp. 206–215.
[63] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos, "Continuous monitoring of distance-based outliers over data streams," in *Proc. 27th Int. Conf. Data Eng. (ICDE)*, Hannover, Germany, Apr. 2011, pp. 135–146.
[64] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "ActiveClean: Interactive data cleaning for statistical modeling," *Proc. PVLDB*, vol. 9, no. 12, pp. 948–959, 2016.
[65] A. Lopatenko and L. Bravo, "Efficient approximation algorithms for repairing inconsistent databases," in *Proc. 23rd Int. Conf. Data Eng. (ICDE)*, Istanbul, Turkey, Apr. 2007, pp. 216–225.
[66] W. Yin, T. Yue, H. Wang, Y. Huang, and Y. Li, "Time series cleaning under variance constraints," in *Proc. Int. Workshops, Database Syst. Adv. Appl. DASFAA, BDMS, BDQM, GDMA, and SeCoP*, Gold Coast, QLD, Australia, May 2018, pp. 108–113.
[67] J. Ma and J.-F. Teng, "Predict chaotic time-series using unscented Kalman filter," in *Proc. Int. Conf. Mach. Learn.*, vol. 2, Aug. 2004, pp. 687–690.
[68] M. Marczak, T. Proietti, and S. Grassi, "A data-cleaning augmented Kalman filter for robust estimation of state space models," *Econometrics Statist.*, vol. 5, pp. 107–123, Feb. 2018.
[69] C. Mayfield, J. Neville, and S. Prabhakar, "ERACER: A database approach for statistical inference and data cleaning," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Indianapolis, IN, USA, Jun. 2010, pp. 75–86.
[70] G. W. Morrison and D. H. Pike, "Kalman filtering applied to statistical forecasting," *Manage. Sci.*, vol. 23, no. 7, pp. 768–774, 1977.
[71] J. Neville and D. Jensen, "Relational dependency networks," *J. Mach. Learn. Res.*, vol. 8, pp. 653–692, Mar. 2007.
[72] G. Park, A. C. Rutherford, H. Sohn, and C. R. Farrar, "An outlier analysis framework for impedance-based structural health monitoring," *J. Sound Vib.*, vol. 286, nos. 1–2, pp. 229–250, 2005.
[73] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation," *J. Power Sources*, vol. 134, no. 2, pp. 277–292, 2004.
[74] Z. Qu, Y. Wang, C. Wang, N. Qu, and J. Yan, "A data cleaning model for electric power big data based on spark framework," *Int. J. Database Theory Appl.*, vol. 9, no. 3, pp. 137–150, 2016.
[75] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, "Holoclean: Holistic data repairs with probabilistic inference," *Proc. PVLDB*, vol. 10, no. 11, pp. 1190–1201, 2017.

[76] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. 28th Int. Conf. Artif. Neural Netw. Artif. Neural Netw. Mach. Learn. (ICANN)*, Munich, Germany, Sep. 2019, pp. 703–716.
[77] K. Rong and P. Bailis, "ASAP: Prioritizing attention via time series smoothing," *Proc. VLDB Endowment*, vol. 10, no. 11, pp. 1358–1369, Aug. 2017.
[78] C. Shilakes and J. Tylman, "Enterprise information portals. Enterprise software team," *Enterprise Inf. Portals*, pp. 354–362, Oct. 1998.
[79] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *J. Time Ser. Anal.*, vol. 3, no. 4, pp. 253–264, Jul. 1982.
[80] S. Song, Y. Cao, and J. Wang, "Cleaning timestamps with temporal constraints," *Proc. PVLDB*, vol. 9, no. 10, pp. 708–719, 2016.
[81] S. Song, C. Li, and X. Zhang, "Turn waste into wealth: On simultaneous clustering and cleaning over dirty data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, Aug. 2015, pp. 1115–1124.
[82] S. Song, A. Zhang, J. Wang, and P. S. Yu, "SCREEN: Stream data cleaning under speed constraints," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Melbourne, VIC, Australia, May/Jun. 2015, pp. 827–841.
[83] A. Swami and J. M. Mendel, "ARMA parameter estimation using only output cumulants," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 38, no. 7, pp. 1257–1265, Jul. 1990.
[84] K. H. Tae, Y. Roh, Y. H. Oh, H. Kim, and S. E. Whang, "Data cleaning for accurate, fair, and robust models: A big data—AI integration approach," in *Proc. 3rd Int. Workshop Data Manage. End-to-End Mach. Learn.*, Amsterdam, The Netherlands, Jun. 2019, pp. 5:1–5:4.
[85] D. Tian, Y. Zhu, X. Duan, J. Hu, Z. Sheng, M. Chen, J. Wang, and Y. Wang, "An effective fuel-level data cleaning and repairing method for vehicle monitor platform," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 410–422, Jan. 2019.
[86] R. S. Tsay, "Outliers, level shifts, and variance changes in time series," *J. Forecasting*, vol. 7, no. 1, pp. 1–20, 1988.
[87] J. Van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transp. Res. C, Emerg. Technol.*, vol. 13, nos. 5–6, pp. 347–369, Oct./Dec. 2005.
[88] M. Volkovs, F. Chiang, J. Szlichta, and R. J. Miller, "Continuous data cleaning," in *Proc. IEEE 30th Int. Conf. Data Eng. (ICDE)*, Chicago, IL, USA, Mar./Apr. 2014, pp. 244–255.
[89] D. Wang, L. M. Kaplan, and T. F. Abdelzaher, "Maximum likelihood analysis of conflicting observations in social sensing," *J. ACM Trans. Sensor Netw.*, vol. 10, no. 2, pp. 30:1–30:27, 2014.
[90] X. Wang, X. L. Dong, and A. Meliou, "Data X-ray: A diagnostic tool for data errors," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Melbourne, VIC, Australia, May/Jun. 2015, pp. 1231–1245.
[91] L. Wei, E. J. Keogh, and X. Xi, "Saxually explicit images: Finding unusual shapes," in *Proc. 6th IEEE Int. Conf. Data Mining (ICDM)*, Hong Kong, Dec. 2006, pp. 711–720.
[92] J. Wijsen, "Reasoning about qualitative trends in databases," *Inf. Syst.*, vol. 23, no. 7, pp. 463–487, 1998.
[93] J. Wijsen, "Trends in databases: Reasoning and mining," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 3, pp. 426–438, May 2001.
[94] Z. Xing, J. Pei, and P. S. Yu, "Early classification on time series," *Knowl. Inf. Syst.*, vol. 31, no. 1, pp. 105–127, Apr. 2011.
[95] S. Xu, B. Lu, M. Baldea, T. F. Edgar, W. Wojsznis, T. Blevins, and M. Nixon, "Data cleaning in the process industries," *Rev. Chem. Eng.*, vol. 31, no. 5, pp. 453–490, 2015.
[96] M. Yakout, L. Berti-Équille, and A. K. Elmagarmid, "Don't be scared: Use scalable automatic repairing with maximal likelihood and bounded changes," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, New York, NY, USA, Jun. 2013, pp. 553–564.
[97] K. Yamanishi and J. Takeuchi, "A unifying framework for detecting outliers and change points from non-stationary time series data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Edmonton, AB, Canada, Jul. 2002, pp. 676–681.
[98] Z. Yu and X. Chu, "Piclean: A probabilistic and interactive data cleaning system," in *Proc. Int. Conf. Manage. Data SIGMOD Conf.*, Amsterdam, The Netherlands, Jun./Jul. 2019, pp. 2021–2024.
[99] A. Zhang, S. Song, and J. Wang, "Sequential data cleaning: A statistical approach," in *Proc. Int. Conf. Manage. Data, SIGMOD Conf.*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 909–924.
[100] A. Zhang, S. Song, J. Wang, and P. S. Yu, "Time series data cleaning: From anomaly detection to anomaly repairing," *Proc. VLDB Endowment*, vol. 10, no. 10, pp. 1046–1057, Jun. 2017.
[101] Z. Aoqian, "Research on time series data cleaning," Ph.D. dissertation, School Softw., Tsinghua Univ., Beijing, China, 2018.
[102] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.
[103] Z. Zhang, D. Yang, T. Zhang, Q. He, and X. Lian, "A study on the method for cleaning and repairing the probe vehicle data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 419–427, Mar. 2013.
[104] J. Zhou and A. K. H. Tung, "Smiler: A semi-lazy time series prediction system for sensors," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Melbourne, VIC, Australia, May/Jun. 2015, pp. 1871–1886.
[105] Y. Zhuang, L. Chen, X. S. Wang, and J. Lian, "A weighted moving average-based approach for cleaning sensor data," in *Proc. 27th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Toronto, ON, Canada, Jun. 2007, p. 38.
[106] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.
[107] P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: Fault detection using an LSTM-based predictive data model," 2016, *arXiv:1612.06676*. [Online]. Available: https://arxiv.org/abs/1612.06676
[108] P. Malhotra, V. Tv, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder," 2016, *arXiv:1608.06154*. [Online]. Available: https://arxiv.org/abs/1608.06154
[109] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. 23rd Eur. Symp. Artif. Neural Netw. (ESANN)*, Bruges, Belgium, Apr. 2015.
[110] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*. [Online]. Available: https://arxiv.org/abs/1607.00148
[111] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD)*, Portland, OR, USA, 1996, pp. 226–231.
[112] A. Dukhovny, "Markov chains with quasitoeplitz transition matrix: Applications," *Int. J. Stochastic Anal.*, vol. 3, no. 2, pp. 141–152, 1900.
[113] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.
[114] J. Cai, "A Markov model of switching-regime ARCH," *J. Bus. Econ. Stat.*, vol. 12, no. 3, pp. 309–316, 1994.
[115] H. Xu, J. Ding, P. Li, D. Sgandurra, and R. Wang, "An improved SMURF scheme for cleaning RFID data," *Int. J. Grid Utility Comput.*, vol. 9, no. 2, pp. 170–178, 2018.
[116] M. Dong, D. Yang, Y. Kuang, D. He, S. Erdal, and D. Kenski, "$Pm_{2.5}$ concentration prediction using hidden semi-Markov model-based times series data mining," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 9046–9055, 2009.
[117] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of hmm, ANN and GA for stock market forecasting," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 171–180, 2007.
[118] A. Gupta and B. Dhingra, "Stock market prediction using hidden Markov models," in *Proc. Students Conf. Eng. Syst.*, Mar. 2012, pp. 1–4.
[119] S. Song, Y. Sun, A. Zhang, L. Chen, and J. Wang, "Enriching data imputation under similarity rule constraints," *IEEE Trans. Knowl. Data Eng.*, to be published.
[120] A. Zhang, S. Song, Y. Sun, and J. Wang, "Learning individual models for imputation," in *Proc. 35th IEEE Int. Conf. Data Eng. (ICDE)*, Macao, China, Apr. 2019, pp. 160–171.
[121] Y. Sun, L. Peng, H. Li, and M. Sun, "Exploration on spatiotemporal data repairing of parking lots based on recurrent gans," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 467–472.
[122] C. Fang, S. Song, Z. Chen, and A. Gui, "Fine-grained fuel consumption prediction," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Beijing, China, Nov. 2019, pp. 2783–2791.

[123] R. Corizzo, M. Ceci, and N. Japkowicz, "Anomaly detection and repair for accurate predictions in geo-distributed big data," *Big Data Res.*, vol. 16, pp. 18–35, Jul. 2019.
[124] M. Milani, Z. Zheng, and F. Chiang, "Currentclean: Spatio-temporal cleaning of stale data," in *Proc. 35th IEEE Int. Conf. Data Eng. (ICDE)*, Macao, China, Apr. 2019, pp. 172–183.
[125] J. Wang, H. Zhang, B. Fang, X. Wang, G. Yin, and X. Yu, "Edcleaner: Data cleaning for entity information in social network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.
[126] Y. Huang, M. Milani, and F. Chiang, "PACAS: Privacy-aware, data cleaning-as-a-service," in *Proc. IEEE Int. Conf. Big Data*, Seattle, WA, USA, Dec. 2018, pp. 1023–1030.
[127] R. Huang, Z. Chen, Z. Liu, S. Song, and J. Wang, "Tsoutlier: Explaining outliers with uniform profiles over iot data," in *Proc. Int. Conf. Big Data*. Los Angeles, CA, USA: Springer, 2019.
[128] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*. Cham, Switzerland: Springer, 2017.

XI WANG is currently pursuing the M.S.E. degree with the School of Software, Tsinghua University, Beijing, China. His current research interests include time series data quality and cleaning.

CHEN WANG received the B.S. and M.S. degrees from the Department of Computer Science, Fudan University, in 2003 and 2006, respectively. He is currently the Chief Scientist with the National Engineering Laboratory for Big Data Software. Before joining Tsinghua University, he was a Research Manager with the Information Management Department, IBM Research, China. He has published more than 20 articles in refereed conferences and journals. He holds 15 issued and pending patents in the U.S. and China. His research interests include the IoT technology, stream computing, and big data systems.