IEEE Open Journal of the Computer Society

Received 15 July 2023; accepted 15 October 2023. Date of publication 19 October 2023;
date of current version 1 November 2023. The review of this article was arranged by Associate Editor Celimuge Wu.

Digital Object Identifier 10.1109/OJCS.2023.3326150

Finding the Truth From Uncertain Time Series by Differencing

JIZHOU SUN [1], DELIN ZHOU [1], AND BO JIANG [1]

Huaiyin Institute of Technology, Huaian 223003, China

CORRESPONDING AUTHOR: BO JIANG (e-mail: jiangbo@hyit.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 62002131, and in part by the Shuangchuang Ph.D award (from World Prestigious Universities) of Jiangsu Province under Grant JSSCBS20211179.
This article has supplementary downloadable material available at https://doi.org/10.1109/OJCS.2023.3326150, provided by the authors.

ABSTRACT Time series data is ubiquitous and of great importance in real applications. But due to poor qualities and bad working conditions of sensors, time series reported by them contain more or less noises. To reduce noise, multiple sensors are usually deployed to measure an identical time series and from these observations the truth can be estimated, which derives the problem of truth discovery for uncertain time series data. Several algorithms have been proposed, but they mainly focus on minimizing the error between the estimated truth and the observations. In our study, we aim at minimizing the noise in the estimated truth. To solve this optimization problem, we first find out the level of noise produced by each sensor based on differenced time series, which can help estimating the truth wisely. Then, we propose a quadratic optimization model to minimize the noise of the estimated truth. Further, a post process is introduced to refine the result by iteration. Experimental results on both real world and synthetic data sets verify the effectiveness and efficiency of our proposed methods, respectively.

INDEX TERMS Differencing, optimization, time series, truth discovery.

I. INTRODUCTION

In recent years, the Internet of Things (IoT) and wireless sensor networks [1], [2], [3] have emerged and are playing a key role in many fields such as smart cities, smart farms, healthcare systems, environmental monitoring projects and so on. As one of the main types of data in IoT, time series data becomes commonplace in these areas. For example, in the application of a smart farm, a batch of sensors would be deployed throughout the farmland to monitor temperature, soil moisture, the concentration of CO_2, etc. These measurements can be collected in the form of time series to analyze the effect of the environment on crop growth. As another example, wearable sensors can be integrated onto the bodies of players to monitor their motion and physiological conditions. This can help the coach mastering the real-time situations of the players and making decisions timely. Researchers have being developing various algorithms over time series [4], [5], [6]. These algorithms can be used for analyzing, mining, predicting and visualization. In the smart farm application, data analyzing and data mining algorithms can extract knowledge such as association rules between environment measurements and crop growth from the abundant time series. In the second example, visualization algorithms can present the tedious and abundant data in a more intuitive manner to the coach.

Although it is relatively easy to collect data using sensors, in the real world, time series reported by sensors always deviate from true data, which we refer to as the uncertainty of time series. There are many reasons that can cause uncertainty in time series, such as imprecision of sensors, privacy-preserving requirements, and low confidence in some data-producing methods. Among them, imprecise sensors contribute the largest share. In the smart farming setting, deployed sensors may have low precision due to budget limitations. In healthcare systems, wearable sensors are often designed to be as lightweight as possible to avoid affecting the movement of the wearers, which can result in lower data precision. In real applications, using inaccurate data directly for time series data analysis may lead to incorrect decision-making. To cope with uncertainty of data, two mainstream classes of methods have been proposed: uncertain data

algorithms [7] and truth discovery methods [8]. Uncertain data algorithms typically employ probabilistic models or statistical techniques to represent and reason about uncertain data. They take into account the probability distributions associated with the data values and incorporate uncertainty propagation and inference mechanisms to make decisions or perform computations. But uncertain data algorithms are usually more complex and inefficient. Truth discovery methods, on the other hand, focus on recovering the true data from multiple imprecise information. For example, in a smart farm application, although individual sensors may have errors, deploying multiple sensors allows us to design algorithms that can accurately recover the correct data as much as possible from multiple sensors' measurements. Although it cannot guarantee the acquisition of absolutely correct data, truth discovery methods have significant advantages of low development cost and high efficiency. By reducing uncertainty, regular algorithms can be directly applied on the recovered data, thus saving the efforts of developing complex uncertain data algorithms for various of problems.

Many truth discovery methods working on uncertain time series have been proposed. Some complex algorithms assume that different data sources (sensors) have varying reliabilities, so their contributions to true discovery are different. These algorithms combine truth discovery with the data source reliability estimation into a joint framework, and seek the optimal solution by iteration or optimization. Although existing methods have obtained promising results, they don't make full use of the inherent information within the observations in terms of estimating data source reliability.

In this paper, we approach the problem from a different perspective. We consider the sequence reported by a sensor as a combination of the ground truth sequence and a noise sequence. The presence of more noise indicates that the data source is less reliable. Therefore, the data source reliability can be represented by the noise level of the observations. So, the motivation of this study is to estimate the data source reliability by analyzing the noise intensity in the observation values. In the real world, time series are used to represent the information of an actual object within a given time range. The values in a time series generally do not fluctuate dramatically in a very short time period. Therefore, it is reasonable to consider the ground truth series as continuous and smooth. But random noise series are not continuous in essence. Based on the difference of smoothness, we propose a novel method to estimate noise level of data source by performing differencing on the observation sequence. Through differencing, the truth series component in the observation can be significantly attenuated, making the noise estimation more accurate.

After estimating the noise of data source, the reliability or weight of data source can be obtained by solving a noise minimization model. Then, the weighted sum of observations is taken as the estimated truth. As an alternative, we propose a quadratic noise minimization model to estimate data source weights by representing the noise using the differenced series of the estimated truth.

The smoothness of time series has been utilized in some existing truth discovery methods, primarily as a constraint in optimizing the objective function [9]. However, this differs from the purpose of our method. Meanwhile, differencing operation is also utilized in many time series data analysis tasks, but its main purpose is to transform unstable sequences into stable ones, which is different from our motivation either.

To further enhance the accuracy of the proposed noise minimization model, we incorporate an iterative process into it. After estimating the ground truth, the weights of data source can be updated, and subsequently, the ground truth can be re-estimated. These two processes can be performed in an iterative manner.

In summary, our main contributions are listed as follows:

1) We propose a new noise minimization framework to perform truth discovery from uncertain time series.
2) We propose a new method to estimate noise in time series based on differencing operation and provide theoretical support.
3) We improve our optimization model by iteration and propose a quadratic optimization model as an alternative.
4) Experimental results show that our algorithms obtain state of the art performance in both accuracy and time cost.

In the rest of this paper, related works are surveyed in Section II, and the problem definition is formalized in Section III. We give our optimization model based on the weighted noise minimization and an alternative quadratic model in Section IV. A refining model by iteration is introduced in Section V. Experiments are conducted in Section VI. Advantages and disadvantages of the proposed methods are discussed in Section VII.

II. RELATED WORKS

A. TIME SERIES TRUTH DISCOVERY

Truth discovery aims to resolve conflicts from multi-source noisy information. This problem has been extensively studied over the years. Truth discovery method can be designed for categorical [10], [11], [12], [13], [14], [15] or continuous numerical data [9], [16], [17], [18], [19], [20], [21], [22], [23], [24]. In this section, we only focus on algorithms working for continuous numerical data because time series in real applications are seldom made of categorical data.

The simplest truth discovery method for numerical data is using the average value of observations to be the estimated truth. But this method doesn't consider the qualities of data sources and the estimation is not robust to data outliers. Another simple method uses the median value of observations as the estimated truth. It is robust to data outliers but the median value is just the middle value of the observations and it may not be close to the truth.

Many existing algorithms share a common underlying principle, that is, if a data source provides more trustworthy information, it is more reliable, and if a piece of information is

provided by a more reliable data source, it is more trustworthy. In light of the high relevance between data source reliability and the ground truth, reliability estimation and truth discovery are often combined into a joint framework, which can be solved by iteration, optimization or probability methods [8].

In [16], the authors proposed a probability graph model GTM, in which the variance of data source is modeled by a gamma distribution, the ground truth and the observation values are modeled by Gaussian distributions. Truth discovery is equivalent to getting the maximum a posterior estimate(MAP) for the truth. Gaussian distribution is also used to represent the source variance in [17]. To tackle the long tail phenomenon of data sources, variance is estimated using the upper bound of the confidence interval. Bootstrapping technology is integrated into truth discovery procedure to decrease the effect of the outlying claims [18]. The authors in [19] built a randomized Gaussian mixed model, in which each data source be considered as a component. The source bias is modeled by an uniform distribution, and observations are considered fitting a multi-variate Gaussian distribution. The estimated truth is found by maximizing the likelihood of observing the multi-source input. Li et al. [20] proposed a truth discovery method which works on heterogeneous data types by designing various distance functions for categorical and continuous numerical data.

There are some methods taking advantages of the correlations of the related objects. If two objects are similar, their observations should be similar too. Such a principle is formulated as a regularization term in the objective function of the optimization model [21]. The authors in [22] proposed a chain graph probability model, in which the dependency of the related objects is modeled by a Markov random field. The optimal inferred truth and the source weights are found by maximizing the posterior distribution conditioned on the observations.

Dynamic truth discovery problem has also received a lot of attention in recent years. Zhi et al. [23] proposed using first-order Markov process to represent the temporal dynamics of time series. Observations are represented by hidden Markov model, in which the truths are considered as the latent variables. EM algorithm is used to infer model parameters. Kalman filter and smoother are used to estimate the ground truth. In [24], truth discovery component and time series analysis component are combined together. Temporal patterns in time series are modeled by SARIMA model, by which data prediction is performed. Estimated truth is encouraged to be close to the prediction to fit the temporal pattern. In [9], the relationship between different object observations is represented by a random local regression model. In addition, the algorithm utilizes the smoothness property of time series by setting a constraint that the values at adjacent time points should be similar.

B. TIME SERIES DATA CLEANING

A related topic to time series truth discovery is data cleaning, which mainly deals with data missing, data inconsistence, data integration and so on [25]. Many algorithms have been proposed to tackle these problems,which mainly include smoothing-based methods, constraint based methods and statistics based methods. The typical smoothing-based methods include Moving average (MA) method, Autoregressive (AR) model, and Kalman filtering model. They can be used not only for data smoothing but also for data prediction. Constraint based methods use dependency between data to check and remedy error [26]. Statistics based algorithms use probabilistic models to learn data statistical characteristics and make inference to data. Hidden markov model (HMM) is a commonly used statistical model, which has been used to predict stocks price and clean RFID data [27], [28].

Data cleaning is often integrated into truth discovery method or it can be used as an independent preprocess step before conducting truth discovery. In [16], [22], data outliers are detected based on the relative and absolute errors.

III. PROBLEM DEFINITION

In this paper, time series is not treated as discrete, but a discrete sampling from a continuous time function $v(t)$, which is usually smooth. We use $V = \langle v_1, v_2, \ldots, v_n \rangle$ to denote the ground truth time series of length n. However, V is unknown, but the approximate values can be measured by sensors. Suppose there are m sensors, observations from them are denoted by $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$,where $S_i = \langle s_{i,1}, s_{i,2}, \ldots, s_{i,n} \rangle$ and $s_{i,t}$ denotes the measurement at time t by the i'th sensor. Due to the inaccuracy of sensors, $s_{i,t} = v_j + e_{i,t}$, where $e_{i,t}$ is the noise introduced by the i'th sensor at time t, which is an observation of $N_{i,t}$, the noise random variable of this sensor at time t. With respect to $N_{i,t}$, we make some reasonable assumptions:

1) For each sensor, noises at different timestamps are independent identically distributed. Then the noise of the i'th sensor can be denoted by N_i.
2) There is no systematic error in sensors, i.e., $E[N_i] = 0$.
3) N_i is independent with the true time series.
4) For any two different sensors, their noises are independent with each other.

The main target of truth discovery on uncertain time series is to find out the estimated ground truth: $\hat{V} = \langle \hat{v_1}, \hat{v_2}, \ldots, \hat{v_n} \rangle$ from observations $\mathcal{S}$, so that the distance $Dist(V, \widehat{V})$ between the estimated series and the ground truth is as small as possible.

IV. OPTIMIZATION MODEL BASED ON NOISE MINIMIZATION AND DIFFERENCING OPERATION

As introduced in section II, true time series can be estimated by the average of the observations:

$$\hat{V} = \frac{1}{m} \sum_{i=1}^{m} S_i$$

But this method does not consider the reliability of the data source. A better strategy is to assign different weights to the data sources based on their reliability. The less noise

the observation contains, the more reliable the corresponding data source is, and a higher weight should be assigned to the source. According to Assumption 2, where the expectation of noise is 0, the noise intensity can be represented by the variance. Therefore, an improved estimation formula becomes:

$$\begin{aligned} &\hat{V} = \sum_{i=1}^{m} w\left(\sigma_i^2\right) S_i \\ &\text{s.t.} \sum_{i=1}^{m} w\left(\sigma_i^2\right) = 1 \\ &w\left(\sigma_i^2\right) \geq 0, i = 1, 2, \ldots, m \end{aligned} \tag{1}$$

Where ${\sigma_i}^2 = D[N_i]$, and $w(\sigma_i^2)$ is a weight function of noise variance, which should monotonically decrease as σ_i^2 increases. Then truth discovery can be decomposed into two phases. In the first phase, the noise of each data source $\hat{\Sigma} = \langle \hat{\sigma_1^2}, \hat{\sigma_2^2}, \ldots \hat{\sigma_m^2} \rangle$ should be accurately estimated. In the second phase, the weight function $w(\sigma_i^2)$ should be determined such that the weighted sum of the observations is as close to the ground truth as possible.

A. ESTIMATING σ_i^2 WITH S_i

From Assumption 2, $E[N_i] = 0$, thus $\sigma_i^2 = D[N_i] = E[N_i^2] - E^2[N_i] = E[N_i^2]$. To estimate $E[N_i^2]$, we can make full use of the observation S_i, by the concept of energy.

Definition 1: The energy of a time series S_i is:

$$P(S_i) = \frac{1}{n} \sum_{t=1}^{n} S_{i,t}^2 \tag{2}$$

$S_{i,t}$ denotes the random variable reported by the i'th sensor at time t, then $S_{i,t} = v_t + N_{i,t}$, we can get:

$$P(S_i) = \frac{1}{n} \sum_{t=1}^{n} (v_t + N_{i,t})^2$$

Then the expectation of the observation series energy is:

$$\begin{aligned} E(P(S_i)) = E\left(\frac{1}{n} \sum_{t=1}^{n} v_t^2\right) + E\left(\frac{1}{n} \sum_{t=1}^{n} N_{i,t}^2\right) \\ + E\left(\frac{2}{n} \sum_{t=1}^{n} v_t \times N_{i,t}\right) \end{aligned}$$

The first term on the right side is constant, the second term on the right side is the expectation of N_i^2 and equals σ_i^2, and the third term on the right side equals 0 because $E(N_{i,t})$ is 0 according to Assumption 2.

Thus, the above equation can be simplified as:

$$E(P(S_i)) = \frac{1}{n} \sum_{t=1}^{n} v_t^2 + \sigma_i^2 \tag{3}$$

If v_t is near 0, σ_i^2 can be estimated with $E(P(S_i))$:

$$\hat{\sigma_i^2} = E(P(S_i)) \tag{4}$$

$E(P(S_i))$ can be approximated by $P(S_i)$ using (2). It is obvious that $E(P(S_i))$ is a biased estimation of σ_i^2. Because V is unknown, the bias can not be remedied.

B. REDUCING THE BIAS BY DIFFERENCING

From (3) and (4), the variance estimation bias is equal to $\frac{1}{n} \sum_{t=1}^{n} v_t^2$. If v_t approaches 0, the bias will be very small, resulting in a more accurate variance estimation. Considering that true time series are typically smooth, the values of adjacent elements are likely to be similar. Therefore, we consider using first-order differenced series of the observation to estimate the noise variance of data source.

Definition 2: Differenced series is defined recursively: zero-order differenced series $S_i^{(0)}$ of S_i is itself, i.e., $S_i^{(0)} = S_i$, k-order differenced series $S_i^{(k)}$ of S_i is $\langle S_{i,2}^{(k-1)} - S_{i,1}^{(k-1)}, S_{i,3}^{(k-1)} - S_{i,2}^{(k-1)}, \ldots, S_{i,n-k+1}^{(k-1)} - S_{i,n-k}^{(k-1)} \rangle$, where $k \in \{1, 2, \ldots, n-1\}$.

Then, the energy of first-order differenced series $S_i^{(1)}$ is:

$$P\left(S_i^{(1)}\right) = \frac{1}{n-1} \sum_{t=1}^{n-1} \left(S_{i,t}^{(1)}\right)^2 \tag{5}$$

The expectation of $P(S_i^{(1)})$ can be obtained:

$$E\left(P\left(S_i^{(1)}\right)\right) = \frac{1}{n-1} \sum_{t=1}^{n-1} \left(v_t^{(1)}\right)^2 + 2\sigma_i^2 \tag{6}$$

$E(P(S_i^{(1)}))$ can be approximated by $P(S_i^{(1)})$ using (5). Because $v_t^{(1)}$ is small compared to v_t, σ_i^2 can be more accurately estimated with:

$$\hat{\sigma_i^2} = \frac{1}{2(n-1)} \sum_{t=1}^{n-1} \left(S_{i,t}^{(1)}\right)^2 \tag{7}$$

And the expectation of $\hat{\sigma_i^2}$ is:

$$E\left(\hat{\sigma_i^2}\right) = \frac{1}{2(n-1)} \sum_{t=1}^{n-1} \left(v_t^{(1)}\right)^2 + \sigma_i^2 \tag{8}$$

For a smooth time series, three adjacent values are also likely to lie on a straight line. That is to say, change of difference of two adjacent values is small. Therefore, we can use second-order difference of the observation to estimate σ_i^2.

It's not hard to get the following equation:

$$\hat{\sigma_i^2} = \frac{1}{6(n-2)} \sum_{t=1}^{n-2} \left(S_{i,t}^{(2)}\right)^2 \tag{9}$$

And the expectation of $\hat{\sigma_i^2}$ is:

$$E\left(\hat{\sigma_i^2}\right) = \frac{1}{6(n-2)} \sum_{t=1}^{n-2} \left(v_t^{(2)}\right)^2 + \sigma_i^2 \tag{10}$$

To be more general, d-order difference of the observation can be used to estimate σ_i^2:

$$\hat{\sigma_i^2} = \frac{1}{(n-d)\binom{2d}{d}} \sum_{t=1}^{n-d} \left(S_{i,t}^{(d)}\right)^2 \tag{11}$$

And the expectation of $\hat{\sigma_i^2}$ is:

$$E\left(\hat{\sigma_i^2}\right) = \frac{1}{(n-d)\binom{2d}{d}} \sum_{t=1}^{n-d} \left(v_t^{(d)}\right)^2 + \sigma_i^2 \tag{12}$$

Proof:

From Definition 1 and 2,

$$\begin{aligned} P\left(S_i^{(d)}\right) &= \frac{1}{n-d} \sum_{k=1}^{n-d} \left[\sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} S_{i,t+k}\right]^2 \\ &= \frac{1}{n-d} \sum_{k=1}^{n-d} \left[\sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} \times (v_{t+k} + N_{i,t+k})\right]^2 \\ &= P\left(V^{(d)}\right) + P\left(N_i^{(d)}\right) \\ &\quad + \frac{2}{n-d} \sum_{k=1}^{(n-d)} \left[\sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} v_{t+k} \right. \\ &\quad \left. \times \sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} N_{i,t+k}\right] \end{aligned}$$

Then,

$$\begin{aligned} E\left(P\left(S_i^{(d)}\right)\right) &= E\left(P\left(V^{(d)}\right)\right) + E\left(P\left(N_i^{(d)}\right)\right) \\ &\quad + \frac{2}{n-d} \sum_{k=1}^{(n-d)} \left[\sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} v_{t+k} \right. \\ &\quad \left. \times E\left(\sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} N_{i,t+k}\right)\right] \end{aligned}$$

Because $E(N_{i,t+k}) = 0$,

$$E\left(P\left(S_i^{(d)}\right)\right) = E\left(P\left(V^{(d)}\right)\right) + E\left(P\left(N_i^{(d)}\right)\right) \tag{13}$$

As V is smooth, $V^{(d)}$ is small comparing to $N_i^{(d)}$ and can be ignored. So:

$$E\left(P\left(S_i^{(d)}\right)\right) \approx E\left(P\left(N_i^{(d)}\right)\right) \tag{14}$$

Since

$$\begin{aligned} E\left(P\left(N_i^{(d)}\right)\right) &= \frac{1}{n-d} \sum_{k=1}^{n-d} E\left(\sum_{t=0}^{d} (-1)^{(d+t)} \binom{d}{t} N_{i,t+k}\right)^2 \\ &= \frac{1}{n-d} \sum_{k=1}^{n-d} E\left[\sum_{t=0}^{d} \binom{d}{t}^2 N_{i,t+k}^2 \right. \\ &\quad \left. + 2 \sum_{t=0}^{d} \sum_{t1>t}^{d} (-1)^{(2d+t+t1)} \binom{d}{t} \binom{d}{t1} N_{i,t+k} N_{i,t1+k}\right] \end{aligned}$$

$N_{i,t+k}$ and $N_{i,t1+k}$ are independent, so,

$$E\left(P\left(N_i^{(d)}\right)\right) = \frac{1}{n-d} \sum_{k=1}^{n-d} \sum_{t=0}^{d} \binom{d}{t}^2 E\left(N_{i,t+k}^2\right)$$

Since

$$\sum_{t=0}^{d} \binom{d}{t}^2 = \binom{2d}{d}$$

,so

$$E\left(P\left(N_i^{(d)}\right)\right) = \frac{1}{n-d}(n-d)\binom{2d}{d} E\left(N_i^2\right) = \binom{2d}{d}\sigma_i^2 \tag{15}$$

From (14),

$$\sigma_i^2 \approx \frac{E\left(P\left(S_i^{(d)}\right)\right)}{\binom{2d}{d}}$$

We use $P(S_i^{(d)})$ to approximate $E(P(S_i^{(d)}))$ and get:

$$\hat{\sigma_i^2} = \frac{P\left(S_i^{(d)}\right)}{\binom{2d}{d}} = \frac{1}{(n-d)\binom{2d}{d}} \sum_{t=1}^{n-d} \left(S_{i,t}^{(d)}\right)^2 \tag{16}$$

From (13), (15), and (16),

$$E\left(\hat{\sigma_i^2}\right) = \frac{E\left(P\left(V^{(d)}\right)\right)}{\binom{2d}{d}} + \sigma_i^2 = \frac{1}{(n-d)\binom{2d}{d}} \sum_{t=1}^{n-d} \left(v_t^{(d)}\right)^2 + \sigma_i^2$$

The proof completes.

From (12), variance estimation bias is:

$$bias = \frac{1}{(n-d)\binom{2d}{d}} \sum_{t=1}^{n-d} \left(v_t^{(d)}\right)^2 \tag{17}$$

It can be seen that as d increases, the bias declines exponentially. Therefore, $\hat{\sigma_i^2}$ in (11) can be seen as the unbiased estimation of σ_i^2 under high order difference.

To gain a clear understanding of the relationship between difference order and the accuracy of noise estimation, we conduct the following experiment. We randomly choose 5 true time series from UCR dataset [29] to calculate noise estimation biases using (17), and then generate 5 random noise series following a standard normal distribution to estimate noise variances using (11). From Fig. 1, we can see that as d increases, the noise estimation bias decreases rapidly and approaches zero, while the noise estimation remains relatively stable.

Although experiment shows that the value of d has not much impact on noise estimation based on a small amount of samples, it affects efficiency when d is large.

Besides unbiasedness, the stability of the estimation is also should be considered, i.e., the variance of $\hat{\sigma_i^2}$. Further investigation reveals that as d increases, the variance of the noise

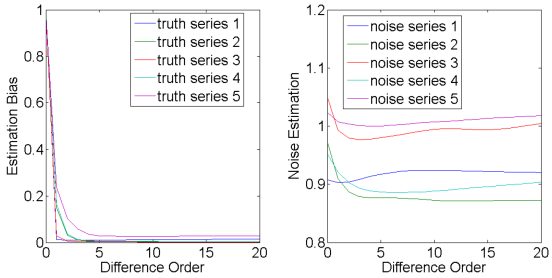

FIGURE 1. Relationship between noise estimation accuracy and difference order.

estimation also increases, indicating that the noise estimation becomes less stable.

We derive the following theorem regarding the relationship between the difference order and the stability of noise estimation:

Theorem 1: Suppose $\mu_4 = R\sigma_s^4$, where μ_4 is fourth order central moment of noise random variable, R is a constant related to the noise, σ_s is the standard variance of noise from the s'th data source, then the standard variance of the estimated σ_s^2 is:

$$\sigma\left(\hat{\sigma_s^2}\right) \approx \frac{\sigma_s^2}{\binom{2d}{d}(n-d)} \times \sqrt{4\sum_{n\geq j>i\geq 1}\left[\sum_{t=1}^{n-d}\binom{d}{i-t}\binom{d}{j-t}\right]^2 + R - 1} \tag{18}$$

The proof is given in appendix.

The theorem can guide us to choose an appropriate difference order d. As d increases, the variance of the estimation will also increases. If $\sigma(\hat{\sigma_s^2}) \leq \epsilon\sigma_s^2$ is required, d should be chosen to satisfy the following inequality:

$$\frac{1}{\binom{2d}{d}(n-d)}\sqrt{4\sum_{n\geq j>i\geq 1}\left[\sum_{t=1}^{n-d}\binom{d}{i-t}\binom{d}{j-t}\right]^2 + R - 1} \leq \epsilon$$

C. DETERMINING THE WEIGHT FUNCTION $w(\cdot)$

Due to the smoothness property, it is reasonable to assume that the truth series contains no noise. Therefore, weight functions should be determined in a way that minimizes the noise present in the estimated truth. According to (1), noise in the estimated truth can be represented by:

$$\begin{aligned}\sigma^2\left(\hat{V}_t\right) &= \sigma^2\left(\sum_{i=1}^{m} w\left(\sigma_i^2\right) \times S_{i,t}\right) \\ &= \sigma^2\left(\sum_{i=1}^{m} w\left(\sigma_i^2\right) \times (v_t + N_{i,t})\right) \\ &= \sum_{i=1}^{m} w^2\left(\sigma_i^2\right) \times \sigma_i^2 \end{aligned} \tag{19}$$

Then, weight functions can be determined by the following noise minimization model:

$$\begin{aligned}&\min \sum_{i=1}^{m} w^2\left(\sigma_i^2\right) \times \sigma_i^2 \\ &\text{s.t.} \sum_{i=1}^{m} w\left(\sigma_i^2\right) = 1 \\ &w\left(\sigma_i^2\right) \geq 0, i = 1, 2, \ldots, m \end{aligned} \tag{20}$$

Because σ_i^2 is unknown, we use $\hat{\sigma_i^2}$ to approximate σ_i^2 using (11), and $w(\hat{\sigma_i^2})$ can be calculated with Lagrange Multiplier method:

$$w\left(\hat{\sigma_i^2}\right) = \frac{\frac{1}{\hat{\sigma_i^2}}}{\sum_{l=1}^{m}\frac{1}{\hat{\sigma_i^2}}} \tag{21}$$

Finally, the estimated ground truth is:

$$\hat{V} = \sum_{i=1}^{m} w\left(\hat{\sigma_i^2}\right) \times S_i \tag{22}$$

In the above optimization model, the noise of the estimated truth is transformed into a weighted sum of the noise in the observations, as can be seen from (19). Alternatively, we can directly represent the noise with the d-order difference of the estimated truth using (11). Then, the optimization model can be formulated as:

$$\begin{aligned}&\min\ \sigma^2\left(\hat{V}^{(d)}\right) = \frac{1}{(n-d)\binom{2d}{d}}\sum_{j=1}^{n-d}\left(\sum_{i=1}^{m} w_i \times S_{i,j}^{(d)}\right)^2 \\ &\text{s.t.} \sum_{i=1}^{m} w_i = 1 \\ &w_i \geq 0, i = 1, 2, \ldots, m \end{aligned} \tag{23}$$

Define

$$D_j = \left[S_{1,j}^{(d)}, S_{2,j}^{(k)}, \ldots, S_{m,j}^{(d)}\right]^T$$
$$w = [w_1, w_2, \ldots, w_m]^T$$

Then, (23) converts to

$$\begin{aligned}&\min\ w^T\left(\sum_{j=1}^{n-d}\left(D_j \times D_j^T\right)\right)w \\ &\text{s.t.} \sum_{i=1}^{m} w_i = 1 \\ &w_i \geq 0, i = 1, 2, \ldots, m \end{aligned} \tag{24}$$

The optimization is quadratic and convex, so it can be easily solved. We use QuadProg++ library [30] to solve this optimization problem.

Algorithm 1: Iterative Optimization Truth Discovery.

Input: difference order d; observations $S = \{S_1, S_2, \ldots, S_m\}$ from m data sources
Output: truth estimation $\hat{V}$

1: **for** $i = 1$ to m **do**
2: calculate $\hat{\sigma_i^2}$ using (11)
3: **end for**
4: **for** $i = 1$ to m **do**
5: calculate $w(\hat{\sigma_i^2})$ using (21)
6: **end for**
7: calculate initial estimation $\hat{V}^{(0)}$ using (22)
8: **while** energy of the estimated $\hat{V}^{(k)}$ decreases **do**
9: **for** i=1 to m **do**
10: update $w_i^{(k+1)}$ according to (26) and (21)
11: **end for**
12: update $\hat{V}^{(k+1)}$ by (22)
13: **end while**
14: **return** $\hat{V}^{(k)}$

Finally, the estimated ground truth can be calculated using:

$$\hat{V} = \sum_{i=1}^{m} w_i \times S_i \tag{25}$$

V. REFINING

A. ITERATIVE OPTIMIZATION MODEL BASED ON DIFFERENCED SERIES ENERGY

In noise minimization model (20), σ_i^2 is estimated with (11). But due to randomness of the observation, noise estimation cannot be equal to its actual value. In addition, performing differencing operation on the observation may not eliminate the estimation bias completely. So, the estimated truth with (22) is not optimal, but it can be considered a good approximation of the ground truth. Based on this estimation, we can re-estimate the noise of data source using (26).

$$\hat{\sigma_i^2} = \frac{1}{n}\sum_{t=1}^{n}(S_{i,t} - \hat{V}_t)^2 \tag{26}$$

Then, we can re-estimate the ground truth using (21) and (22). These two estimation steps can be performed iteratively.

We give the detail steps of the iterative optimization process in Algorithm 1.

Lines 1–7 implement the optimization model from (20) to (22). To start the iteration, the weight of data source $w_i^{(0)}$ can be initialized by $w(\hat{\sigma_i^2})$, and ground truth estimation $\hat{V}^{(0)}$ be initialized by $\hat{V}$. Then, in the $(k+1)$th iteration, $w_i^{(k+1)}$ can be calculated by $\hat{\sigma_i^2}$, which is derived from $\hat{V}^{(k)}$ using (26). Based on $w_i^{(k+1)}$, $\hat{V}^{(k+1)}$ can be calculated with (22). $w_i^{(k)}$ and $\hat{V}^{(k)}$ can be updated iteratively until convergence. In our setting, iteration continues until d-order difference energy of $\hat{V}^{(k)}$ doesn't decrease. Some other iteration stop conditions can be used, for example, iteration number reaches a specified number. However, doing so may lead to the estimated ground truth being trapped by the nearest neighboring sample, thus preventing effective algorithm improvement.

It is not hard to obtain that the time complexity of Algorithm 1 is $O(dmn + T \times (mn + dn))$, where T is the number of iterations, m is the number of data sources, n is the length of time series, and d is the difference order. It is obvious that the complexity is linear with the input data size $m \times n$.

VI. EXPERIMENTAL EVALUATION

In our research, all experiments are conducted on a desktop computer equipped with 16 GB of RAM, a 2.90 GHz Intel Core i7 CPU, and running Windows 10 operating system. The programming environment used is Dev-C++ 5.11.

A. DATASET

We perform all experiments on UCR time series classification dataset [29]. UCR contains 85 data sets in total. Each data set consists of two parts: TRAIN and TEST, each containing a set of time series data. Within each time series, the first data point represents a category label used for classification tasks, which is not utilized in our experiments. The subsequent data points represent the actual time series data, which serve as the ground truth. For each data set, we combine TRAIN and TEST parts and use all time series to evaluate algorithms. To simulate observations from data sources, we add Gaussian noise to the ground truth at each timestamp in every time series. We generate m observations for each time series. For each observation, the expectation of the Gaussian distribution is set to 0 and the standard variance of the distribution is randomly selected from the range (0,1].

B. EVALUATION METRICS

Mean of Absolute Error (MAE) and Root Mean of Square Error (RMSE) are used for evaluating algorithm performance. MAE calculates the mean of absolute difference from the estimated truths to the ground truths and RMSE calculates the square root of the mean of square distance between the estimated truths to the ground truths. The less MAE and RMSE are, the better the algorithm is.

C. BASELINE METHODS

We compare our algorithms with eight commonly used methods: Mean, Median, OTD [24], RelSen [9], GTM [16], CRH [20], PTDCorr [22] and CATD [17]. CRH minimizes the weighted sum of error between the observations and the ground truths. CATD minimizes the weighted sum of error variance from data sources. GTM and PTDCorr both maximize the log likelihood probability of the ground truth, source qualities and the observations, but PTDCorr considers object correlation. In our implementation, we assume there exists temporal correlation among the truths at adjacent timestamps. Besides considering the weighted error between truths and observations, RelSen also considers object correlation and smoothness of time series, while OTD mines the temporal patterns within the truth series.

TABLE 1. Error Comparison of Different Algorithms on UCR Dataset

Dataset	Metric	Mean	Median	OTD	Relsen	CRH	GTM	CATD	PTDCorr	DIFF	ITER	QUAD
50words	RMSE	0.1815	0.1367	0.1944	0.1205	0.1296	0.0771	0.0895	0.0681	0.0649	**0.0647**	**0.0647**
	MAE	0.145	0.1038	0.1529	0.0963	0.1035	0.0616	0.0715	0.0543	0.0519	**0.0517**	**0.0517**
Adiac	RMSE	0.1795	0.134	0.1889	0.1183	0.1266	0.0721	0.0883	0.0705	0.062	0.0619	**0.0618**
	MAE	0.1433	0.1013	0.1486	0.0944	0.1012	0.0576	0.0706	0.0563	0.0496	0.0495	**0.0494**
ArrowHead	RMSE	0.1797	0.134	0.1775	0.1182	0.1254	0.0732	0.0868	0.0675	0.0607	**0.0606**	0.0608
	MAE	0.1432	0.1013	0.139	0.0943	0.1001	0.0585	0.0691	0.0539	0.0484	**0.0483**	0.0485
Beef	RMSE	0.185	0.1427	0.1838	0.125	0.1232	0.0814	0.0841	**0.0586**	0.0651	0.065	0.0649
	MAE	0.1473	0.1078	0.1453	0.0995	0.0984	0.0651	0.0672	**0.0468**	0.0521	0.0521	0.0521
BeetleFly	RMSE	0.1889	0.1395	0.1562	0.1244	0.1314	0.0532	0.0926	0.0748	0.0439	**0.0437**	**0.0437**
	MAE	0.151	0.1038	0.1223	0.0995	0.1049	0.0424	0.074	0.0597	0.0351	**0.0349**	**0.0349**
BirdChicken	RMSE	0.1816	0.1356	0.2086	0.1202	0.1327	0.0795	0.0895	**0.06**	0.0617	0.0616	0.0616
	MAE	0.1454	0.1029	0.1634	0.0962	0.1059	0.0636	0.0715	**0.048**	0.0493	0.0492	0.0492
Car	RMSE	0.1853	0.1415	0.1815	0.1237	0.1231	0.0875	0.0835	**0.0604**	0.0691	0.0691	0.0691
	MAE	0.148	0.1078	0.1428	0.0988	0.0983	0.07	0.0667	**0.0482**	0.0552	0.0552	0.0552
CBF	RMSE	0.1808	0.1346	0.2766	0.1245	0.1271	**0.0685**	0.0886	0.084	0.1181	0.0802	0.1271
	MAE	0.1445	0.102	0.2142	0.0995	0.1017	**0.0548**	0.0709	0.0671	0.0943	0.0641	0.1016
Chlorine Concentration	RMSE	0.1805	0.1346	0.2009	0.1235	0.1283	**0.071**	0.0894	0.0798	0.1126	0.0747	0.1066
	MAE	0.1443	0.1019	0.157	0.0987	0.1025	**0.0568**	0.0714	0.0633	0.09	0.0597	0.0852
CinC_ECG_torso	RMSE	0.1795	0.1332	0.178	0.1179	0.1276	0.0791	0.0889	0.0767	0.0617	**0.0615**	**0.0615**
	MAE	0.1432	0.1007	0.1409	0.094	0.1018	0.0631	0.0709	0.0612	0.0492	0.0491	**0.049**
Coffee	RMSE	0.1812	0.1356	0.1746	0.12	0.1277	0.071	0.0894	0.0693	0.0604	**0.0597**	0.0598
	MAE	0.1445	0.1019	0.1368	0.0957	0.1020	0.0569	0.0715	0.0555	0.0483	0.0478	**0.0477**
Computers	RMSE	0.1793	0.1335	0.1846	0.1214	0.1289	0.0721	0.0895	0.073	0.0898	**0.061**	0.0697
	MAE	0.1431	0.1004	0.1459	0.0967	0.1028	0.0576	0.0714	0.0581	0.0717	**0.0487**	0.0557
Cricket_X	RMSE	0.1805	0.1354	0.2088	0.1217	0.1261	0.077	0.0879	0.0693	0.0909	**0.0682**	0.0793
	MAE	0.1442	0.1027	0.164	0.0969	0.1008	0.0615	0.0702	0.0548	0.0726	**0.0545**	0.0634
Cricket_Y	RMSE	0.1811	0.134	0.1929	0.1202	0.1264	0.0732	0.0871	0.0674	0.0837	**0.063**	0.0715
	MAE	0.1447	0.1014	0.1515	0.0959	0.1009	0.0583	0.0695	0.0535	0.0668	**0.0503**	0.0571
Cricket_Z	RMSE	0.1801	0.1338	0.1781	0.1206	0.1289	0.0744	0.0886	0.0695	0.089	**0.0642**	0.0748
	MAE	0.1438	0.1013	0.1394	0.0959	0.1030	0.0594	0.0708	0.0548	0.0711	**0.0513**	0.0597
DiatomSize Reduction	RMSE	0.1825	0.1393	0.201	0.1219	0.1285	0.0743	0.0896	0.0671	0.0624	0.0624	**0.0623**
	MAE	0.1457	0.1052	0.1582	0.0972	0.1025	0.0594	0.0715	0.0535	0.0498	0.0498	**0.0497**
DistalPhalanx OutlineAgeGroup	RMSE	0.178	0.1318	0.216	0.117	0.1270	0.0702	0.0894	0.079	0.0646	**0.0614**	0.0645
	MAE	0.1428	0.1003	0.1679	0.0939	0.1017	0.0562	0.0716	0.0633	0.0517	**0.0491**	0.0517
DistalPhalanx OutlineCorrect	RMSE	0.1793	0.1337	0.1959	0.1188	0.1296	0.0711	0.0912	0.0797	0.0653	**0.0616**	0.065
	MAE	0.1435	0.101	0.1518	0.095	0.1037	0.0568	0.073	0.0638	0.0521	**0.0491**	0.0519
DistalPhalanxTW	RMSE	0.1808	0.1358	0.208	0.1203	0.1275	0.0728	0.0894	0.0784	0.0668	**0.0633**	0.0662
	MAE	0.1447	0.1032	0.1607	0.0963	0.1019	0.0583	0.0713	0.0625	0.0535	**0.0507**	0.053
Earthquakes	RMSE	0.1805	0.1351	0.2067	0.1435	0.1261	**0.0781**	0.0881	0.1006	0.1522	0.0836	0.1373
	MAE	0.1443	0.102	0.1629	0.1139	0.1008	**0.0623**	0.0704	0.0775	0.1218	0.0668	0.1095

Bold entities in each row mean the minimal error of all compared methods on a corresponding dataset in the leftmost column.

Parameters for the baseline methods are set as follows. For OTD, parameters are set as ($p = 5, P = 5, E = 0, d = 5, D = 5, M = 5, \eta = 10, \delta = 0.5, \lambda = 0.001, g = 1$). For RelSen, parameters are set as ($l = 2709, \gamma = 1.0$). For GTM, parameters are set as ($\alpha = 1, \beta = 1, \mu_0 = 0, \sigma_0^2 = 1$). For CATD, parameter α is set as 0.05. For PTDCorr, parameters are set as ($\alpha = 5, \beta = 1, \theta = 0.2$). The number of data sources is set to 10 for all methods. If not specified otherwise, difference order d is set to 2.

D. ACCURACY EVALUATION

We compute the average RMSE and MAE for each data set in UCR. Due to space limitations, we present the results of the first 20 data sets in Table.1. In this table, DIFF represents our first optimization model in (20), ITER represents our refining model incorporating an iterative process, which is shown in algorithm 1, and QUAD represents our quadratic optimization model in (24). From Table.1, we can observe that our methods are highly effective, achieving the best results in 14 out of 20 data sets. This demonstrates the effectiveness of truth discovery framework based on differencing operation and noise minimization. In comparison, ITER exhibits higher accuracy, indicating that the iterative process has a significant impact on the algorithm's performance.

MAE and RMSE of different methods on all data sets in UCR are plotted in Figs. 2 and 3. Among our methods, we

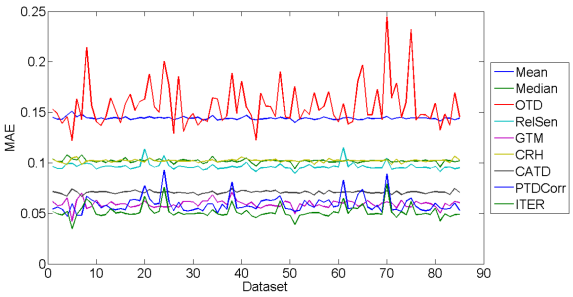

FIGURE 2. MAE comparison on UCR dataset.

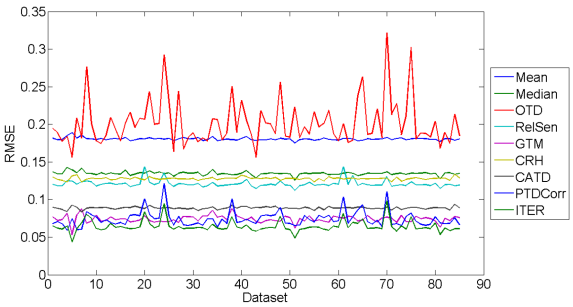

FIGURE 3. RMSE comparison on UCR dataset.

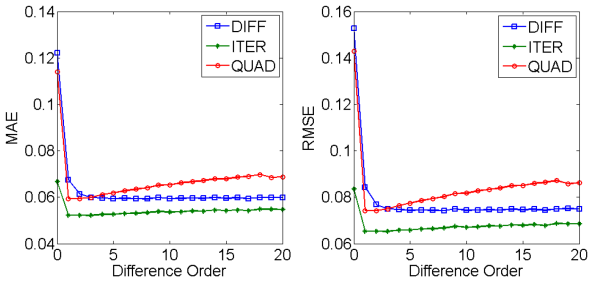

FIGURE 4. Relationship between difference order and MAE/RMSE for our three methods.

choose ITER to compare with the baselines because it performs best in most cases. From the results, we can see that ITER outperforms all the baseline methods and gets the best average accuracy in 73 out of all 85 data sets. Among the baseline methods, Mean and Median methods don't consider the relationship between the data source reliability and truth discovery, so the accuracy of these two methods is not satisfactory. OTD performs worst in most data sets. A possible reason may be that SARIMA model cannot fit the time series in UCR dataset well. GTM and PTDCorr obtain comparable performance with ITER. Because PTDCorr considers object correlation, it performs slightly better than GTM overall.

To explore the impact of difference order on the proposed methods, we range difference order d from 0 to 20 and calculate average MAE and RMSE of all time series for each difference order. Fig. 4 shows the relationship between difference order and MAE/RMSE. From the results, we can

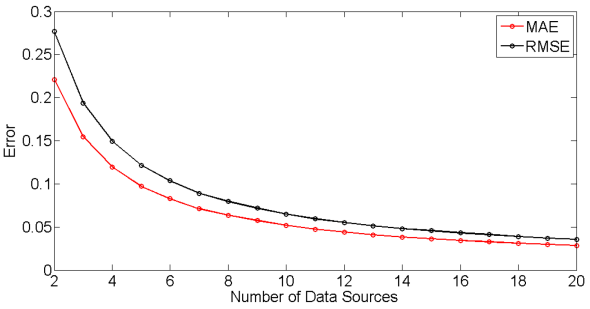

FIGURE 5. Relationship between number of data sources and error for ITER method.

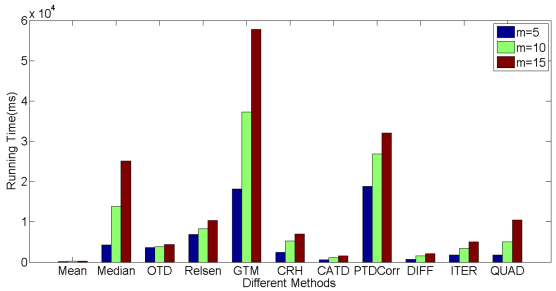

FIGURE 6. Time performance comparison for different methods.

observe that when the difference order equals zero, i.e., when no differencing operation is performed on the observations, the accuracy of each algorithm is worse compared to when the differencing operation is performed. The phenomenon shows the differencing operation can reduce the bias of noise estimation. Meanwhile, it can be observed that the difference order doesn't have a significant impact on the accuracy of the algorithms. When the difference order ranges from 1 to 20, the relative estimation errors are very small, which is a significant advantage compared to other baseline methods. Many existing methods require setting various parameters, and different parameter settings often heavily impact performance. In addition, we can observe that ITER is always better than DIFF. However, as the difference order increases, the performance of QUAD is not as good as that of the other two methods.

To explore the impact of the number of data sources on accuracy, we vary m from 2 to 20 and compute the average MAE and RMSE for each m. Results are shown in Fig. 5. We can observe that as m increases, the algorithm can acquire more information related to the truth, resulting in a decrease in the estimation error.

E. TIME PERFORMANCE EVALUATION

Besides the accuracy, we also compare time performance of different methods. We measure the total execution time of different methods separately when the number of data sources is 5, 10, and 15. We use *qsort* function in *C* standard library to implement Median method. Results are shown in Fig. 6.

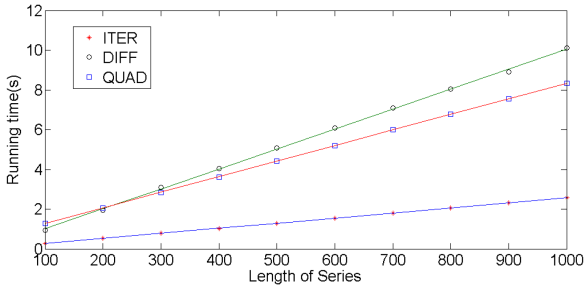

FIGURE 7. Relationship between running time and length of series for our three methods.

From the results, we can observe that our methods exhibit excellent time performance. Although GTM and PTDCorr get less estimation error in a small portion of data, they need much more execution time. For QUAD, the time cost increases more rapidly than DIFF and ITER because it requires performing complex matrix decomposition operations. In addition, as the number of data sources increases, the size of the Hessian matrix in quadratic optimization also increases.

Further, we explore the relationship of the time cost with the length of series for our three methods. We synthesize truth series of lengths ranging from 100 to 1000, increasing by 100 each time. The first element is set to 0, and each subsequent element is set to a Gaussian random variable with an expectation equal to the value at the previous timestamp and a variance equal to 1. Observations of the data sources are simulated by the same method as before. For each length, we build 10 0000 truth series. We test each algorithm 10 times and take the average as the execution time. Test results are shown in Fig. 7. It can be found that there are strong linear relationship between the length of series and the time cost. Pearson correlation coefficients are 1, 0.99996 and 0.99999 for DIFF, ITER and QUAD respectively.

VII. DISCUSSION

Compared to the baseline methods, the benefits of our algorithms lay in the following aspects:

1) Many existing methods combine truth discovery and estimation of data source weights into a joint framework, but the source weights are often initialized equally or set based on the error between the observations and the initial truths before iteration [9], [17], [20], [22]. In contrast, in our paper, weights of data source are computed based on estimating the noise of data source by performing differencing operations on the observations. Through differencing, the smooth signal in the time series can be greatly attenuated while preserving the noise signal effectively, thereby reducing interference from the true values in noise estimation. To the best of our knowledge, this method is novel and can be applied to other time series analysis tasks.
2) Our methods only require setting one parameter, which is the difference order, and different parameter setting has a small impact on the performance of the algorithm. In contrast, many existing algorithms [9], [22], [24] have several parameters that need to be set. Different parameter setting has heavily impact on the performance of the algorithm, making these algorithms more complex to use.
3) We use probabilistic methods to theoretically demonstrate the relationship between parameter setting and stability of the noise estimation. Therefore, stability of our algorithms can be theoretically guaranteed.

Among the proposed three models, DIFF performs most efficiently and the algorithm's accuracy is also satisfactory. ITER incorporates an iterative process into DIFF, obtaining higher accuracy with only a slight increase in execution time. Therefore, it can be seen as an improved version of DIFF. QUAD can be seen as an alternative to DIFF. It shares the same optimization objective as DIFF, but the form of their objective functions is different. QUAD is a quadratic optimization model that incurs higher computational costs when the difference order is large. It involves matrix decomposition operations during the algorithm execution. In addition, as the difference order increases, the computational stability may decrease. However, it is worth noting that in cases with lower orders (e.g., order 1,2), QUAD can achieve better performance than DIFF. Therefore, in low-order scenarios, QUAD can be chosen, while ITER can be chosen in most cases. If there is a high requirement for time performance, DIFF can be used.

The main drawback of our methods is they can't perform truth discovery in online manner. On the other hand, our methods are designed for single object, not considering multi object correlations. In future work, we will integrate data preprocessing into the truth discovery framework and improve algorithm performance further.

REFERENCES

[1] Q. Chen, Z. Cai, L. Cheng, H. Gao, and J. Li, "Low-latency concurrent broadcast scheduling in duty-cycled multihop wireless networks," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 851–860.
[2] Q. Chen, H. Gao, S. Cheng, X. Fang, Z. Cai, and J. Li, "Centralized and distributed delay-bounded scheduling algorithms for multicast in duty-cycled wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3573–3586, Dec. 2017.
[3] Q. Chen, Z. Cai, L. Cheng, H. Gao, and J. Li, "Energy-collision-aware minimum latency aggregation scheduling for energy-harvesting sensor networks," *ACM Trans. Sensor Netw.*, vol. 17, no. 4, pp. 1–34, 2021.
[4] M. Yadav, S. Jain, and K. Seeja, "Prediction of air quality using time series data mining," in *Proc. Int. Conf. Innov. Comput. Commun.*, 2019, pp. 13–20.
[5] Y. Shi, T. Yu, Q. Liu, H. Zhu, F. Li, and Y. Wu, "An approach of electrical load profile analysis based on time series data mining," *IEEE Access*, vol. 8, pp. 209915–209925, 2020.
[6] S. Huang, D. Wang, X. Wu, and A. Tang, "DSANet: Dual self-attention network for multivariate time series forecasting," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 2129–2132.
[7] C. C. Aggarwal and S. Y. Philip, "A survey of uncertain data algorithms and applications," *IEEE Trans. Knowl. data Eng.*, vol. 21, no. 5, pp. 609–623, May 2009.
[8] Y. Li et al., "A survey on truth discovery," *ACM SIGKDD Explorations Newslett.*, vol. 17, no. 2, pp. 1–16, 2016.

[9] C. Feng, X. Liang, D. Schneegass, and P. Tian, "RelSen: An optimization-based framework for simultaneously sensor reliability monitoring and data cleaning," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 345–354.
[10] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 1048–1052.
[11] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. 11th Int. Conf. Inf. Process. Sensor Netw.*, 2012, pp. 233–244.
[12] B. Aydin, Y. S. Y. Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas, "Crowdsourcing for multiple-choice question answering," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2946–2953.
[13] D. Wang, T. Abdelzaher, L. Kaplan, and C. C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst.*, 2013, pp. 530–539.
[14] C. Ye, H. Wang, T. Ma, J. Gao, H. Zhang, and J. Li, "PatternFinder: Pattern discovery for truth discovery," *Knowl.-Based Syst.*, vol. 176, pp. 97–109, 2019.
[15] J. Yang and W. P. Tay, "An unsupervised Bayesian neural network for truth discovery in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5182–5195, Nov. 2021.
[16] B. Zhao and J. Han, "A probabilistic model for estimating real-valued truth from conflicting sources," *Proc. QDB*, vol. 1817, 2012.
[17] Q. Li et al., "A confidence-aware approach for truth discovery on long-tail data," *Proc. VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.
[18] H. Xiao et al., "Towards confidence in the truth: A bootstrapping based truth discovery approach," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1935–1944.
[19] H. Xiao, J. Gao, Z. Wang, S. Wang, L. Su, and H. Liu, "A truth discovery approach with theoretical guarantee," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1925–1934.
[20] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1187–1198.
[21] C. Meng et al., "Truth discovery on crowd sensing of correlated entities," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 169–182.
[22] Y. Yang, Q. Bai, and Q. Liu, "A probabilistic model for truth discovery with object correlations," *Knowl.-Based Syst.*, vol. 165, pp. 360–373, 2019.
[23] S. Zhi, F. Yang, Z. Zhu, Q. Li, Z. Wang, and J. Han, "Dynamic truth discovery on numerical data," in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 817–826.
[24] L. Yao et al., "Online truth discovery on time series data," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 162–170.
[25] X. Wang and C. Wang, "Time series data cleaning: A survey," *IEEE Access*, vol. 8, pp. 1866–1881, 2019.
[26] Z. Liang, H. Wang, X. Ding, and T. Mu, "Industrial time series determinative anomaly detection based on constraint hypergraph," *Knowl.-Based Syst.*, vol. 233, 2021, Art. no. 107548.
[27] A. Gupta and B. Dhingra, "Stock market prediction using hidden Markov models," in *Proc. Students Conf. Eng. Syst.*, 2012, pp. 1–4.
[28] A. I. Baba, M. Jaeger, H. Lu, T. B. Pedersen, W.-S. Ku, and X. Xie, "Learning-based cleansing for indoor RFID data," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 925–936.
[29] "UCR time series classification archive," 2015. [Online]. Available: https://www.cs.ucr.edu/eamonn/time_series_data/
[30] "Github - liuq/quadprogpp: A C++ library for quadratic programming which implements the goldfarb-idnani active-set dual method," 2016. [Online]. Available: https://github.com/liuq/QuadProgpp

JIZHOU SUN received the bachelor's degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009, the master's degree from China Aerospace Science and Industry Corporation, Beijing, China, in 2012, and the Doctoral of Philosophy degree in computer software and theory from the Harbin Institute of Technology, Harbin, China, in 2020. He is currently a Lecture with Huaiyin Institute of Technology, Huaian, China. His research interests mainly include massive data computing, data cleaning, data management, and time series data.

DELIN ZHOU was born in Zhengzhou, Henan Province, China, in 2000. He received the bachelor's degree from the Henan University of Science and Technology, Luoyang, China, in 2022. He is currently working toward the Postgraduation degree with the Huaiyin Institute of Technology, Huaian, China. His research interests include massive data computing, machine learning, and time series data.

BO JIANG received the bachelor's and master's degrees in computer science from Jiangsu University, Zhenjiang, China, in 2001 and 2004, respectively. He is currently a Lecture with the Huaiyin Institute of Technology, Huaian, China. His research interests include image processing, pattern recognition, data cleaning, and time series data.