Whetstone: Reliable Monitoring of Cloud Services

Heng Zhang, Jesus Luna, and Neeraj Suri
Dept. of Computer Science
TU Darmstadt, Germany
Email: {zhang, jluna, suri}@deeds.informatik.tu-darmstadt.de

Ruben Trapero
Atos Research & Innovation
Calle de Albarracin 25, Madrid, Spain
Email: ruben.trapero@atos.net

***Abstract*—Cloud services have become powerful enablers for a variety of smart computing solutions supporting multimedia, social networking, e-commerce and critical infrastructures among others. Consequently, as we increasingly depend on the cloud, the need exists to ensure its effective role as a trustworthy services platform. Towards this objective, a plethora of cloud monitoring mechanisms have been proposed which typically assume that the collected monitoring information is reliably correct. In reality, the information collected by cloud monitors is often susceptible to reliability issues (e.g., monitor malfunctions, data corruptions, or data tampering), and obtaining reliable cloud monitoring information is still an open issue.**

We propose *Whetstone* as a novel approach to address the gap where an efficient approach of ascertaining reliable values from a set of collected monitoring data is required. To this end, *Whetstone* first introduces a statistical approach to filter defective data from the collected data set. Next, *Whetstone* develops an optimization approach to quantify the reliability of the collected data by leveraging the value deviation of the collected data. Finally, *Whetstone* devises a weighted aggregation approach for generating the reliable value based on the obtained information. We evaluate the proposed approach with different experimental configurations. The experimental results demonstrate the efficacy of our approach for successfully generating the maximum likelihood reliable value for raw data sets.

I. Introduction

Cloud services, by virtue of providing transparent access to backend distributed resources, are increasingly underpinning a variety of smart computing projects. For example, a large-scale Internet of Things (IoT) network may utilize cloud services to process the massive data collected by the attached IoT devices which lack the requisite computational resources to locally process the data [1]. Similarly, a smart community may desire an intelligent carpooling service to reduce carbon dioxide (CO_2) emissions for protecting the environment. To this end, the carpooling service takes advantage of a cloud service to compute optimized taxi dispatching plans [2]. Moreover, a smart grid achieves to offer sustainable and economic electricity distribution by utilizing cloud services to manage the communication of heterogeneous information [3]. As the cloud enabled applications proliferate, the increasingly dependency on the cloud also portends the need to ensure the cloud as a dependable services platform.

In this context, the monitoring of the cloud operations (for functionality, resource optimizations and especially the detection of anomalous behaviors) forms an essential basis for securing cloud services. As a result, a large number of monitoring mechanisms have recently been proposed [4]-[9]. However, virtually all existing monitoring schemas assume that the information collected by security monitors is reliably correct. However, this assumption can be fallacious for a variety of reasons, e.g., security monitor malfunctions, unpredictable data/network corruptions, or malicious data tampering [10]. For instance, security monitors deployed for collecting monitored data might encounter the problem of transient malfunctions or failures. As a result, the collected data is unreliable or even completely flawed. Additionally, the monitored data might be corrupted during recording/transmission phases by unpredictable factors such as communication channel congestion or noise [26]. Moreover, the monitored data might be intentionally tampered by attackers for bypassing security mechanisms or for fabricating necessary preconditions for performing subsequent attacks [25]. Therefore, to ensure that existing monitoring mechanisms generate correct cloud services monitoring results, the key point is to ascertain the reliability of the collected data.

In order to improve data reliability, existing methodologies target obtaining reliable data value in two steps [11][12]. The first step is to measure the value of a given target multiple times. The second step is to collate the multiple recorded values to generate a reference representative value. The commonly adopted aggregation technique is termed *Majority Voting* that procures reliable values by taking advantage of a voting process [13]. Specifically, the value with more votes (i.e., occurrence frequency) contributes more to the final procured value in the aggregation process. The value generated by using this technique is the reference "reliable" value with respect to the monitored target. Naturally, the value occurrence directly affects the reliability of the generated mean value. However, a major drawback of this technique is in overlooking the important fact that the reliability of the collected values is distinctive [14]. Supposing that an unreliable value (e.g., erroneous data) occurs in the collected value set for many times, the occurrence-based technique fails to generate reliable value. In other words, this technique is only applicable when every collected value has the same level of reliability. Therefore, an approach that is able to generate reliable values in the presence of raw values with differing reliability degrees is needed.

To address this gap, we propose a novel methodology

Research supported in part by EC H2020 CIPSEC GA #700378 and BMBF TUD-CRISP.

termed *Whetstone* for obtaining the reliable monitoring information from the collected data set in this paper. To this end, we first adopt a data cleansing approach to filter the unreliable data by making use of statistical properties of the monitored data. Then, we propose a data reliability quantification approach by leveraging the relationship of data reliability and value deviation. Finally, we develop a novel weighted aggregation approach to generate reliable values based on the reliability of collected values.

To the best of our knowledge, our approach is the first work proposed for deriving reliable data to support monitoring cloud services. In summary, we make the following contributions:

1) We propose *Whetstone* as a novel methodology to generate reliable monitoring values from collected data by considering the reliability degree of collected data individually.
2) We propose a quantification approach for ascertaining the reliability of the monitored data based on an optimization model and theoretically prove the correctness of the determined reliability results.
3) Our experimental results demonstrate the effectiveness of the proposed approach to generate the reliable value via a tunable optimization coefficient.

The remainder of this paper is organized as follows. Section II describes the considerations of obtaining reliable value from collected data. Section III models the reliable value generation problem. Section IV details the design of our proposed approach. Section V evaluates the effectiveness of our proposition with experiments and makes discussions. Section VI reviews the related work.

II. Background

We first review the challenges of developing an effective methodology to obtain reliable monitoring values. Next, we present the main observations that underpin the development of our proposed methodology.

A. Challenges

Monitored data collected by security monitors contains a variety of useful information (e.g., abnormal workload variation, unusual service customer logins, or occasional virtual machine exceptions) for monitoring cloud services. With reliable monitoring data, cloud monitoring mechanisms are expected to generate correct monitoring results. In reality, the monitored data suffers from various reliability problems, such as failures of security hardware (e.g., monitoring devices), errors of data flows (e.g., communication channels), or manipulations of data sources (e.g., system log tampering). To address these problems, a common solution is to repeatedly measure the value of a given target and aggregate the values for deriving a reliable value. As mentioned in the introduction, a widely adopted aggregation technique is developed based on the *majority voting* principle of the value receiving more votes also more important for the aggregation [13]. In practice, the occurrence of the collected value is generally utilized as the vote and the mean of collected values is generally regarded as the reliable value derived by this technique. The major problem of the technique is that every collected value is considered as uniformly reliable in the aggregation process. However, erroneous values and normal values have completely different significance from a reliability perspective. As a result, it is questionable to simply aggregate the collected values for the reliable value generation by merely considering the value occurrence.

The reliability of monitoring data is the value that represents the degree of the collected data free from errors during a monitoring process. From a reliability perspective, the distance from the collected values to the true value varies. As depicted in Figure 1, the collected values of a monitored data (denoted by green square / red parallelogram / diamond / triangle) are all deviated from the real value (denoted by blue point) to some extent. If the collected values have tiny deviation (e.g., locating inside the small radius (r_1) dashed circle), the aggregated value (denoted by magenta point v_1) is closer to the real value. While the collected values have greater deviation (e.g., locating inside the greater radius (r_2) dashed circle), the aggregated value (denoted by magenta point v_2) is less closer to the real value. These two cases highlight that the reliability of the generated value is directly affected by the reliability of the collected data. As a result, two main problems need to be solved for deriving a reliable value from the collected data. The first problem is

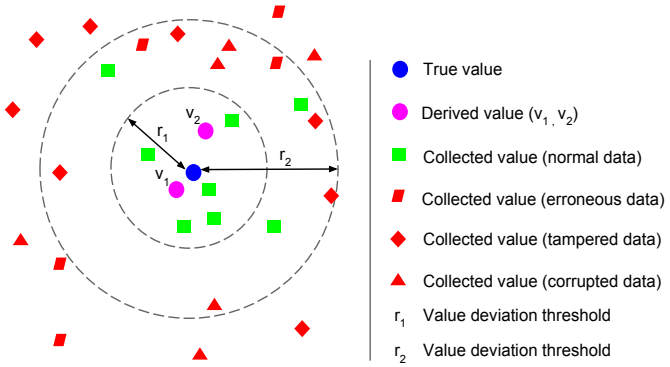

Fig. 1. The insight of the collected value regarding monitored data

that the collected data might contain defective values which involve various reliability issues as aforementioned. Hence, a data cleansing approach that supports filtering defective data from the raw data set is needed for obtaining a cleansed data set. Even with the cleansed data, another challenge is to develop a proper aggregation approach that can utilize the cleansed data for generating reliable values.

B. Observations

Two issues are frequently observed in cloud monitoring activities which also constitute the basis for our proposed methodology.

The first observation is that the population of the monitored data value approximately follows a Gaussian distribution given that the number of collected values is sufficiently large [15].

This high volume data collection is typical for cloud security monitors and thus the relevance of Gaussian distributions. According to the statistical property of Gaussian distribution, if a value is significantly deviated from the mean value (i.e., the true value), it has a high likelihood to be defective.

The second observation is that there is an inverse relationship between the data reliability and the value deviation regarding a monitored data [10]. Namely, the value of reliable monitored data is closer to the true value while the value of unreliable monitored data is distant from the real value. As an example, system event logs contain valuable information for securing cloud services. In normal situation, security auditors/experts can utilize the reliable information extracted from the event logs (e.g., the log of failed login events) to discover potential attack behaviors. However, if the log has been tampered by attackers, the critical information indicating brute-force password attacks against customer accounts can be deliberately removed. Thus, the tampered data contains the information that is quite deviated from the real situation reflected by the reliable data.

These observations highlight the typical characteristics of the monitored data. In the next section, we introduce our proposed approach for generating the reliable value of the monitored data by taking advantage of these observations.

III. Problem Statement

We now present the problem model of generating the reliable values of monitored data, and also outline the relevant notations and terminologies adopted in the paper.

A. Problem Model

In this paper, we particularly consider the problem of generating reliable value of monitored data for performing security monitoring on cloud services. Structurally, we describe the problem with an input-output problem model as follows:

- *Input*
  To obtain a reliable value with respect to a monitored target T in a Cloud service, a set of monitored data $D^T = \{d_1^T, d_2^T, \cdots, d_m^T\}$ $(m \in \mathbb{N})$ is collected by a deployed security monitor for m times. The collected data d_m^T denotes the m^{th} measured value of target T. The size of the collected data set D^T is m.
- *Output*
  Based on the obtained data set D^T, the output is data $\widehat{d^T}$ that represents the reliable value of target T by properly aggregating all collected data $d_1^T, d_2^T, \cdots, d_m^T$.

B. Solution Approach

To obtain the reliable value of monitored data $\widehat{d^T}$, our proposed methodology needs to:

- Obtain a set of cleansed data D'^T by precluding defective data (e.g., corrupted data, erroneous data, or tampered data) from the raw data set D^T.
- Measure the cleansed data's value deviation for taking advantage of the inverse relationship between the value deviation and the data reliability.
- Determine the reliability degree of every cleansed data in D'^T as a weight for generating $\widehat{d^T}$.
- Aggregate the cleansed data and its corresponding weights to generate the reliable value $\widehat{d^T}$.

IV. Proposed Methodology: Whetstone

We first overview the *Whetstone*'s framework prior to detailing the design methodology.

A. System Overview

To obtain reliable monitored data, we propose a multi-step methodology termed *Whetstone*. Prior to discussing the design details, we present the framework of *Whetstone* in Figure 2. The framework of our proposed methodology consists of four major steps summarized as follows.

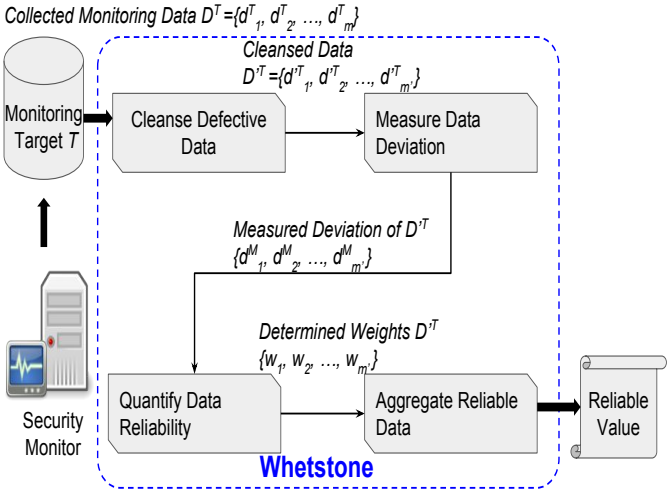

Fig. 2. Framework of the Proposed Methodology

1) *Cleanse defective data:* The proposed methodology starts by conducting a statistical preprocessing process (i.e., Grubbs' test) to filter defective data from the raw monitored data set in order to obtain a cleansed data set as the foundation for supporting subsequent procedures (Section IV-B1).
2) *Measure data deviation:* The methodology measures the cleansed data so as to obtain the value deviation from a reference base (Section IV-B2).
3) *Quantify data reliability:* The methodology introduces an optimization approach to quantify the reliability of every cleansed data by taking advantage of the obtained deviations (Section IV-B3).
4) *Aggregate reliable data: Whetstone* finally aggregates the obtained information (i.e., the cleansed data and its reliability) to generate a reliable value with respect to a specific monitored target (Section IV-B4).

B. Design Methodology

In this part, we explain the design of the proposed methodology that enables us to generate a reliable value from the collected data regarding a monitored target.

*1) **Cleanse defective data**:* To obtain a reliable value, *Whetstone* first requires to cleanse all collected value so as to obtain a cleansed set for aggregating a reliable value. Considering collected values follows a Gaussian distribution [15], we propose to cleanse the collected values by utilizing the Grubbs' test which is a statistical method for identifying far-deviated data in a data set complying with the Gaussian distribution [16].

For a set of collected data, the defective data denotes the most deviated data (i.e., either the greatest or the smallest one) in the m-element data population D^T. Hence, *Whetstone* first proposes two hypotheses of the data state as follows.

- H_0 : *There is no defective data in the collected data set* D^T.
- H_a : *There is exactly one defective data in the collected data set* D^T.

To test the above hypotheses, *Whetstone* applies the following two-sided Grubbs' test formula [16].

$$G = \frac{\max\limits_{i=1,2,\cdots,m} \left| d_i^T - \bar{d^T} \right|}{s} = \frac{\max\limits_{i=1,2,\cdots,m} \left| d_i^T - \bar{d^T} \right|}{\sqrt{\frac{\sum_{i=1}^{m} \left(d_i^T - \bar{d^T} \right)^2}{m-1}}} \tag{1}$$

where $i \in \{1, 2, \cdots, m\}$, G denotes the value of the Grubbs' test, $\bar{d^T}$ represents the mean value of all collected data in D_T, and s is the standard deviation of D^T.

After determining the value of G by Formula (1), *Whetstone* can test the hypothesis as follows. Specifically, H_0 is rejected at the significance level α if

$$G > \frac{m-1}{\sqrt{m}} \sqrt{\frac{(t_{\alpha/(2m),m-2})^2}{m-2+(t_{\alpha/(2m),m-2})^2}} \tag{2}$$

In (2), $t_{\alpha/(2m),m-2})$ represents the t-distribution with $m-2$ degrees of freedom at the $\alpha/(2m)$ significance level. We set the value of α to 0.05 in our paper.

Whetstone executes Grubbs' testing processes in an iterative manner until H_0 is no longer rejected at the specified confidence level α by Formula (2). In consequence, the data that results in the rejections of H_0 in the testing process will be removed from the raw data set. We denote the cleansed data set by $D^{'T} = \{d_1^{'T}, d_2^{'T}, \cdots, d_{m'}^{'T}\}$ $(m^{'} \leqslant m; m, m^{'} \in \mathbb{N})$ which is a subset of the raw data set D^T (i.e., $D^{'T} \subseteq D^T$). Accordingly, the size of the cleansed set $D^{'T}$ is $m^{'}$ and the number of removed data is $m - m^{'}$.

*2) **Measure data deviation**:* After obtaining the cleansed data set $D^{'T}$, *Whetstone* proceeds to quantify the data reliability. As discussed in Section II-B, high reliability data demonstrates the small deviation from the true value, while low reliability data demonstrates the high deviation from the true value. Therefore, the inverse relationship between data reliability and value deviation can be utilized as an effective leverage for ascertaining data reliability. *Whetstone* takes advantage of the inverse relationship to quantify the reliability of cleansed data in $D^{'T}$.

To this end, *Whetstone* needs to tackle a major problem of obtaining the value deviation, as it is challenging to make use of the inverse relationship for data reliability quantification without knowing the deviation value.

The prerequisite for obtaining a value deviation is to have a reference base that is, in theory, the true value with respect to a monitored target. Ideally, the deviation can be measured by applying distance formulas (e.g., Euclidean distance) with the known reference base. However, the ideal reference base is impossible to acquire due to the fact that the true value is unknown in practice. To deal with this problem, it is necessary to introduce an estimated value which functions as the reference base for value deviation measurement. Considering that the cleansed data in $D^{'T}$ is distributed around the true value, the mean value of $D^{'T}$ is thus meaningful in estimating the true value. Hence, we introduce an approximated reference base $\bar{d^{'T}}$ for measuring value deviations as follows.

$$\bar{d^{'T}} = \frac{\sum_{j=1}^{m^{'}} d_j^{'T}}{m^{'}} \tag{3}$$

where $j \in \{1, 2, \cdots, m^{'}\}$, $\bar{d^{'T}}$ denotes the approximated reference base, and $d_j^{'T}$ denotes the data in set $D^{'T}$.

Based on Equation (3), the deviation of the monitored data can be measured by computing the distance between $d_j^{'T}$ and $\bar{d^{'T}}$ as follows.

$$d_j^M = |d_j^{'T} - \bar{d^{'T}}| \tag{4}$$

where d_j^M denotes the measured value deviation between the monitored data $d_j^{'T}$ and the base $\bar{d^{'T}}$.

*3) **Quantify data reliability**:* Besides the deviation obtained by using Equation (4), *Whetstone* also needs a proper method to leverage the inverse relationship for quantifying the reliability of collected data in $D^{'T}$. To facilitate the data reliability quantification, *Whetstone* introduces the weight w for representing the data reliability. The definition of weight is presented as follows.

Definition 1. Weight $w_j \in [0, +\infty]$ is a positive value that is used to proportionally represent the reliability of collected data $d_j^{'T}$ (for some $j = 1, 2, \cdots, m^{'}$).

Noticeably, if the weight is close to the lower bound $w = 0$, it means that the data does not contain much valid monitoring information (i.e., the defective data). If the weight is close to the upper bound $w = +\infty$, it means that the data is absolutely reliable (i.e., the true value).

With the help of weights, the data reliability quantification problem now can be addressed by solving an optimization problem based on the inverse relationship. Specifically, the optimization problem targets finding a particular weight assignment of the cleansed data so as to yield the minimum sum of the product of data weights and data deviations. For

this purpose, a data with great given deviation needs to be assigned with the most possible small weight. The correctness of such an weight assignment is supported by Theorem 1 as follows.

Theorem 1. For a finite set of data pairs (d_j^M, w_j) where d_j^M is constant and w_j is bounded (for some $j \in \{1, 2, \cdots, m^{'}\}$), the minimum sum of $d_j^M \cdot w_j$ can only be achieved on condition that a great w_j is paired with the most possible small d_j^M.

Proof: Let the deviation set D be sorted as $d_1^M < \cdots < d_p^M < \cdots < d_q^M < \cdots < d_j^M$ and the weight set W also be sorted as $w_1 < \cdots < w_p < \cdots < w_q < \cdots < w_j$ $(1 < p < q < j)$, the minimum value M is $M = d_1^M w_j + \cdots + d_p^M w_{j+1-p} + \cdots + d_q^M w_{j+1-q} + \cdots + d_j^M w_1$.

Supposing there exists a value $M^{'} = d_1^M w_j + \cdots + d_p^M w_{j+1-q} + \cdots + d_q^M w_{j+1-p} + \cdots + d_j^M w_1$ smaller than M, then it gives the following inequality as

$M - M^{'} = (d_p^M - d_q^M)(w_{j+1-p} - w_{j+1-q}) > 0$

$\because d_p^M < d_q^M$ and $w_{j+1-p} > w_{j+1-q}$

$\therefore (d_p^M - d_q^M)(w_{j+1-p} - w_{j+1-q}) = M - M^{'} < 0$

It contradicts to the given inequality. Therefore, there is no other value smaller than $M = d_1^M w_j + \cdots + d_p^M w_{j+1-p} + \cdots + d_q^M w_{j+1-q} + \cdots + d_j^M w_1$. ∎

Based on the above consideration, *Whetstone* proposes the following optimization problem for determining the data reliability.

Definition 2. Given a set of measured value deviations $D = \{d_1^M, d_2^M, \cdots, d_{m'}^M\}$ and a set of weights $w = \{w_1, w_2, \cdots, w_{m'}\}$,

$$\begin{aligned} \underset{w}{minimize} \quad & f(w,d) = \sum_{j=1}^{m^{'}} w_j d_j^M \\ s.t. \quad & f_0(w) = \sum_{j=1}^{m^{'}} \alpha^{-w_j} = C \end{aligned} \tag{5}$$

where $\alpha > 1$ and $C \in \mathbb{R}^+$ is a positive coefficient. The optimization problem consists of two proposed functions. Namely, $f(w,d)$ is the proposed objective function that can be optimized by finding the particular weight assignment specified in Theorem 1. $f_0(w)$ is the constraint function that ensures the optimization of $f(w,d)$ is feasible. Given weight w is a variable varying within the range $[0, +\infty]$, we introduce an adjustable coefficient C as the bound of the sum of α^{-w_j} for making the optimization process valid.

To make the objective function optimizable, we introduce a new variable β_j to represent α^{-w_j}. As a result, weight w_j can be represented by,

$$w_j = -log_\alpha^{\beta_j} \tag{6}$$

To determine the optimal value, the Lagrange function of the proposed optimization problem thus can be represented based on Equation (5)(6) as

$$L(\beta_j, \lambda) = \sum_{j=1}^{m^{'}} (-log_\alpha^{\beta_j} \cdot d_j^M) + \lambda (\sum_{j=1}^{m^{'}} \beta_j - C) \tag{7}$$

Given that the sum of the equality constraint function is subject to coefficient C, we can compute the Lagrange multiplier λ of the Lagrange function (7) on the condition that the partiality derivative of β_j is zero.

$$\lambda = \frac{1}{C} \sum_{j=1}^{m^{'}} d_j^M \tag{8}$$

Based on Equation (5)−(8), we determine the value of weight w_j as follows.

$$w_j = -\log_\alpha \frac{C \cdot d_j^M}{\sum_{j=1}^{m^{'}} d_j^M} \tag{9}$$

*4) **Aggregate reliable data**:* With the quantified data reliability, *Whetstone* is able to aggregate the cleansed data with respect to a monitored target. According to our problem model described in Section III-A, the aggregated data can be denoted by $\widehat{d^T}$. To get rid of the potential reliability bias, *Whetstone* proposes a weight-based approach to generate $\widehat{d^T}$. In specific, *Whetstone* determines $\widehat{d^T}$ by aggregating all the data in cleansed set $D^{'T}$ based on the respective reliability degree derived by Equation (9) as follows.

$$\widehat{d^T} = \frac{\sum_{j=1}^{m^{'}} w_j d_j^{'T}}{\sum_{j=1}^{m^{'}} w_j} \tag{10}$$

In Equation (10), the aggregated data d^T considers the reliability contribution of every collected data $d_j^{'T}$ in an uneven manner. Overall, the reliability of $\widehat{d^T}$ is dominated by the high-weight data that has the high possibility to be more approaching to the true value.

V. Evaluation

In this section, we evaluate the efficacy of our proposed methodology for generating reliable data that underpins cloud monitoring. Our evaluation is conducted in two steps: 1.) We assess the effectiveness of *Whetstone* to generate the primitive result by cleansing defective monitoring data, 2.) We investigate *Whetstone*'s performance towards generating the reliable value by tuning up the value of the optimization coefficient. We first describe the settings of our evaluation. Then, we provide a discussion and interpretations of the experimental results.

A. Experimental Setting

To evaluate the performance of our proposed methodology, we perform the evaluation in the scenario where a set of collected data values is aggregated to generate a reliable monitoring value. The collected data follows the unknown Gaussian distribution whose mean is the true value with

respect to a monitored target. The collected data set may partially contain defective data caused by varied reasons.

To simulate the above scenario, we adopt the following experimental settings for evaluating our proposed methodology. Specifically, we randomly generate a set positive values for simulating the values of system overhead collected by security monitors from a target virtual machine which is used for running cloud services. The generated data set follows the Gaussian distribution where the standard deviation is set to $\sigma = 1$ and the mean value is set to $\mu = 80$ which is used as the unknown true value for benchmarking *Whetstone*'s performance. To simulate the defective data, we manipulate a percentage of data in the data set by adding random offsets while keeping the rest data unchanged. Thus, we create four defective data profiles which respectively contain 5%, 10%, 15%, and 20% manipulated data.

B. Evaluation on Primitive Results

We first examine the primitive result generated by *Whetstone* in data cleanse process. In order to check *Whetstone*'s capability of generating more reliable values than the existing majority voting method, we carry out four rounds of experiments on the test data set with different numbers of simulated defective data. In each experiment, we first record the mean value of the data set where the simulated data is distant from the mean value of the Gaussian distribution μ. Such mean value is value generated by applying the majority voting method. Next, we execute our proposed methodology to cleanse the defective data and collect the primitive result which is the mean value of the cleansed data set.

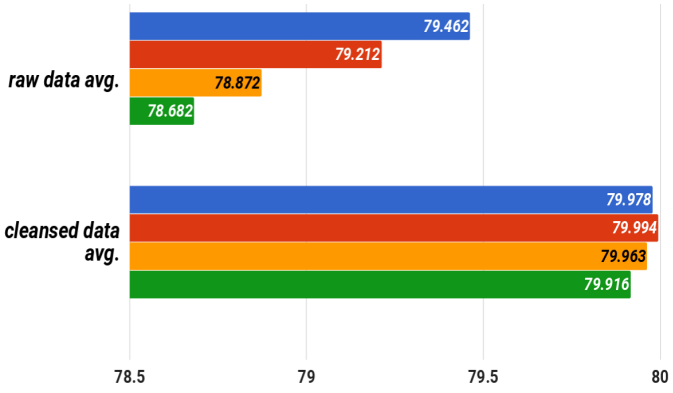

Fig. 3. The primitive results by cleansing defective data

Figure 3 depicts the primitive results that are obtained from the experiments. In this figure, we present the results of different test sets with different colors. Namely, the test set with 5% defective data is in blue, the test set with 10% defective data is in red, the test set with 15% defective data is in orange, and the test set with 20% defective data is in green. From the figure, we observe that *Whetstone* can successfully filter most defective data. Compared to the mean value of the raw data set, the mean value of the cleansed set becomes closer to the benchmarking line at $\mu = 80$ for every defective data profile. As an example of the test set with 20% defective data (shown in green bar), its mean value gets improved from 78.682 to 79.916 by carrying out the cleansed process. It is worth noticing that *Whetstone* successfully filters 15% defective data which is distant from the benchmarking line in the experiment. The other 5% defective data is overlooked by *Whetstone* as it is not greatly deviated from the benchmarking line even with adding the manipulated offsets. In the rest three experiments, we also observe the similar situation that the defective data is overlooked only when the value deviation of the defective data is quite tiny. It is worth mentioning that the primitive result obtained by *Whetstone* outperforms the result derived by the existing work that simply aggregate all data for generating the mean of the raw data set.

C. Evaluation on Final Result

After cleansing defective data, *Whetstone* still needs to deal with the remaining data to generate a final reliable value. The remaining data (including the overlooked defective data) is deviated from the true value to different extents. To investigate the performance of our proposed methodology for generating the reliable value, we perform a series of experiments on the four cleansed test sets by tuning up the coefficient C with different values.

Table I presents the evaluation results collected from these experiments. In this table, the first column represents the amount of defective data contained in the test set that is used for evaluating our proposed methodology. The rest of the columns record the results by adopting different values of the coefficient C. For each value of C, the generated reliable value (denoted by $Opt.Value$ in the table) is recorded. Apart from that, the symbol of the quantified weight (denoted by $Symbl.W$ in the table) is also recorded in order to check the correctness of the determined weight.

From the experimental results in Table I, we observe that the reliable value generated by *Whetstone* keeps approaching towards the true value before crossing the critical point that denotes the alteration of the weight's symbol from *positive* to *negative*. After crossing the critical point, the generated reliable value keeps increasing with assigning greater values to C. For example, we tune up the value of coefficient C to obtain the reliable value for the test set with 20% defective data by using five respective values as $C = \{2, 5, 10, 50, 100\}$. Specifically, *Whetstone* generates a reliable value $Opt.Value = 79.931$ when $C = 2$. The generated value 79.931 is closer to the true value 80 than the mean value directly obtained by the cleansed data set which still contains 5% defective data as mentioned in Section V-B. Tuning up C with greater values, The reliable value generated by *Whetstone* at the 95% confidence level also keeps increasing (e.g., $Opt.Value = 79.935$ when $C = 5$ and $Opt.Value = 79.940$ when $C = 10$). Apart from that, it is worth noticing the alteration of the weight's symbols. We can observe that the symbol of weight is positive when $C = \{2, 5, 10\}$ while the symbol changes to negative when $C = \{50, 100\}$. As a result, we can regard the

TABLE I
THE FINAL RESULTS GENERATED BY TUNING UP THE COEFFICIENT WITH DIFFERENT VALUES

Percentage of Defective Data in Cleansed Set	Coefficient Setting (C)									
	C=2		C=5		C=10		C=50		C=100	
	Opt.Value	Symbl.W	Opt.Value	Symbl.W	Opt.Value	Symbl.W	Opt.Value	Symbl.W	Opt.Value	Symbl.W
5%	79.987	positive	79.989	positive	79.992	positive	80.014	negative	80.080	negative
10%	80.002	positive	80.005	positive	80.008	positive	80.030	negative	80.106	negative
15%	79.975	positive	79.978	positive	79.982	positive	80.015	negative	80.167	negative
20%	79.931	positive	79.935	positive	79.940	positive	79.983	negative	80.213	negative

critical point located in the range $(10, 50)$. The alteration of weight's symbol indicates the validity of the generated value. Considering the weight is used to represent the reliability degree of collected data, the weight has to be a positive value as for its practical significance. If the symbol of weight becomes negative, it indicates that the generated reliable value is invalid.

D. Discussions

From the collected experimental results, *Whetstone* successfully demonstrates its capability of generating reliable values for supporting cloud monitoring. For instance, *Whetstone* is able to derive a more reliable value than the mean value of the data set where the reliability of collected the data is difficult to known. In addition, it can manage to generate reliable values, even if the data set might contain defective data. Besides, *Whetstone* also possesses several advantages which are worth mentioning as follows.

Whetstone can generate reliable monitored data values without assuming all collected data with equal reliability which is challenging to ascertain without conducting a careful investigation. From a reliability perspective, it is questionable for adopting the mean value as reliable value of the collected data. *Whetstone* addresses this challenge by quantifying the reliability of collected data based on its value deviation.

Whetstone introduces a data cleanse process for further improving the reliability of generated results. By cleansing the defective data from the raw monitored data set, the remaining data is less-deviated from the true value and thus provides a better foundation for quantifying the reliability of collected data (w) with higher precision.

Whetstone supports generating the reliable value from a set of collected monitored data in an automatic way. By keeping tuning up the value of coefficient C, the generated result keeps approaching towards the true value. Once the tuning up process yields any negative weight, the automatic process can be terminated and take the latest result before the alteration of the weight symbol as the most likely reliable value.

Whetstone is proposed for generating the reliable monitoring value based on a static data set in our paper, while it can be adapted to manage the dynamic data flow generated by deployed security monitors in cloud systems. As a dynamic data set can be considered as the compilation of many static data set snapshots sequentially ordered by a temporal order.

VI. RELATED WORK

Our survey work first reveals that the validity of existing monitoring mechanisms commonly suffer from the issue of collected data [4]-[9][17]-[20]. For example, Seshaderi et al., [17] proposed a hash-based security technique to monitor remote device state by utilizing the memory information that is assumed to be collected from a noise-free scenario. Practically, it is challenging for security monitors to collect data under the influence of the random noise. Ma et al., [18] proposed *ProTracer* as a monitoring approach to address advanced persistent threat (APT) attacks by making use of the logs of system calls and relate events. While the logs are likely to be tampered by malicious attackers in many cases, the tampered log information leads *ProTracer* to generate incorrect monitoring results. Furthermore, [19] et al., proposed a flow-analyzing security method for monitoring malwares. The proposed method is developed to function with the default prerequisite that the information captured from data flows is fully reliable. In reality, it is hard to meet such a requirement without an effective mechanism for checking the reliability of captured data. Moreover, Gulmezoglu et al., [20] proposed a machine learning based methodology to monitor cloud applications in terms of cache-accessing patterns. Without reliable cache information, it is hard to extract the feature vector that dominates the correctness of monitoring results generated by this methodology.

To the best of our knowledge, few work particularly targets ascertaining reliable monitoring value for achieving reliable cloud monitoring. We observe the work that is relevant to the topic of discovering reliable data for different purposes. For instance, Mahdisoltani et al., [21] proposed a technique to predict reliable data for improving the reliability of storage systems by utilizing training data which requires a lot of effort to prepare. Our proposed approach does not require any extra preparation and can directly work with collected data. Additionally, Zheng et al., [22] proposed a data reliability improvement technique for assigning task packages in an optimized manner. To elevate data reliability, this technique carries out an iterative updating process that causes significant computational overhead. Likewise, Li et al, [10] proposed a method to improve data source reliability by adopting a continuous updating process. Compared to both work, our approach does not need to execute iterations and thus can reduce the computational overhead. Furthermore, Fan et al., [23] proposed a statistical technique to improve data reliability by leveraging the similarity between different topics.

Unfortunately, the value of similarity is hard to obtain due to the reason that the constraints of the statistical model are sometimes too difficult to meet. By contrast, our approach does not need to meet strict constraints for applying it. Moreover, Li et al., [24] proposed an optimization method to improve data reliability while it is hard to derive the suitable values of the optimization parameters without expert knowledge. However, our approach supports generating reliable value by using the suitable optimization parameter.

VII. Conclusion

Acquiring a reliable value of monitoring data is the key factor that affects the correctness of monitoring results generated by cloud monitoring mechanisms. For performing rigorous cloud monitoring, the methodology to obtain the reliable value of the collected data is still a challenge. Therefore, we propose *Whetstone* as a novel multi-step approach to address this gap.

To obtain the reliable monitoring value of monitored data, *Whetstone* starts with cleansing the defective data in collected data set by taking advantage of the significant deviation between the value of defective data and the true value. Afterwards, *Whetstone* quantifies the reliability of the cleansed data by leveraging the relationship of the monitored data's reliability inversely proportional to its value deviation from the true value. Finally, *Whetstone* successfully generates the reliable value of the collected data by aggregating the cleansed data with its reliability in a weighted manner.

The merits of our proposed approach are the capability of generating reliable value of the collected data and supporting to ascertain the reliable value by tuning up the value of the constraint coefficient. In the future, we plan to adapt *Whetstone* to manage categorical type cloud monitoring data. Overall, our paper offers a novel angle for obtaining reliable values of monitored data to support cloud monitoring.

References

[1] M. Kovatsch, M. Lanter, and Z. Shelby, "Californium: Scalable cloud services for the internet of things with coap," in *Proceedings of 2014 International Conference on the Internet of Things (IOT '14),* IEEE, 2014, pp. 1–6.

[2] D. Zhang, T. He, F. Zhang, M. Lu, Y. Liu, H. Lee, and S. H. Son, "Carpooling service for large-scale taxicab networks," *ACM Transactions on Sensor Networks (TOSN)*, 2016, vol. 12, no. 3, p. 18.

[3] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE transactions on cloud computing*, 2015, vol. 3, no. 2, pp. 233–244.

[4] H. Deng, Q. Wu, B. Qin, J. Mao, X. Liu, L. Zhang, and W. Shi, "Who is touching my cloud," in *European Symposium on Research in Computer Security (ESORICS '14)*. Springer, 2014, pp. 362–379.

[5] D. Moldovan, G. Copil, H.-L. Truong, and S. Dustdar, "Mela: Monitoring and analyzing elasticity of cloud services," in *Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom '13)*, vol. 1. IEEE, 2013, pp. 80–87.

[6] K. D. Bowers, M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "How to tell if your cloud files are vulnerable to drive crashes," in *Proceedings of the 18th ACM conference on Computer and communications security (CCS '11)*. ACM, 2011, pp. 501–514.

[7] J. Du, X. Gu, and T. Yu, "On verifying stateful dataflow processing services in large-scale cloud systems," in *Proceedings of the 17th ACM conference on Computer and communications security (CCS '10)*. ACM, 2010, pp. 672–674.

[8] Y. Wu, P. Sun, C. Huang, S. Lu, S. Lai, and Y. Chen, "Eagleeye: Towards mandatory security monitoring in virtualized datacenter environment," in *Proceedings of the 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '13)*. IEEE, 2013, pp. 1–12.

[9] H. Zhang, R. Trapero, J. Luna, and N. Suri, "deQAM: A dependency based indirect monitoring approach for cloud services," in *Proceedings of the 14th IEEE International Conference on Services Computing (SCC '17)*. IEEE, 2017, pp. 27–34.

[10] 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 *Proceedings of the 2014 ACM SIGMOD international conference on Management of data (SIGMOD '14)*. ACM, 2014, pp. 1187–1198.

[11] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, 2012, vol. 13, no. Feb, pp. 491–518.

[12] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proceedings of the 24th Annual Conference on Advances in neural information processing systems (NIPS '11)*, 2011, pp. 1953–1961.

[13] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han, "A survey on truth discovery," *ACM SIGKDD Explorations Newsletter*, 2016, vol. 17, no. 2, pp. 1–16.

[14] D. Yu, H. Huang, T. Cassidy, H. Ji, C. Wang, S. Zhi, J. Han, C. Voss, and M. Magdon-Ismail, "The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding," in *Proceedings of the 25th International Conference on Computational Linguistics (COLING '14)*, 2014, pp. 1567–1578.

[15] Z. Guo, G. Jiang, H. Chen, and K. Yoshihira, "Tracking probabilistic correlation of monitoring data for fault detection in complex systems," in *Proceedings of the 36rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '06)*. IEEE, 2006, pp. 259–268.

[16] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, 1969, vol. 11, no. 1, pp. 1–21.

[17] A. Seshadri, A. Perrig, L. Van Doorn, and P. Khosla, "Swatt: Software-based attestation for embedded devices," in *Proceedings of the 25th IEEE Symposium on Security and Privacy Security and privacy (S&P '04)*. IEEE, 2004, pp. 272–282.

[18] S. Ma, X. Zhang, and D. Xu, "Protracer: Towards practical provenance tracing by alternating between logging and tainting." in *The Network and Distributed System Security Symposium (NDSS '16)*, 2016.

[19] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: capturing system-wide information flow for malware detection and analysis," in *Proceedings of the 14th ACM conference on Computer and communications security (CCS '07)*. ACM, 2007, pp. 116–127.

[20] B. Gulmezoglu, T. Eisenbarth, and B. Sunar, "Cache-based application detection in the cloud using machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIACCS '17)*. ACM, 2017, pp. 288–300.

[21] F. Mahdisoltani, I. Stefanovici, and B. Schroeder, "Proactive error prediction to improve storage system reliability," in *2017 USENIX Annual Technical Conference (USENIX ATC '17)*. USENIX Association, 2017, pp. 391–402.

[22] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng, "QASCA: A quality-aware task assignment system for crowdsourcing applications," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15)*. ACM, 2015, pp. 1031–1046.

[23] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng, "icrowd: An adaptive crowdsourcing framework," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15)*. ACM, 2015, pp. 1015–1030.

[24] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, "On the discovery of evolving truth," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '15)*. ACM, 2015, pp. 675–684.

[25] A. Gervais, H. Ritzdorf, G. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proceedings of the 22nd ACM conference on Computer and communications security (CCS '15)*. ACM, 2015, pp. 692–705.

[26] M. Hiller, A. Jhumka, and N. Suri, "An approach for analysing the propagation of data errors in software," in *Proceedings of the 31rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '01)*. IEEE, 2001, pp. 161–170.