Lightweight Sensing Uncertainty Metric – Incorporating Accuracy and Trust

Eskindir Asmare and Julie McCann

Abstract—The future will involve millions of networked sensors whose sole purpose is to gather data about various phenomena so that it can be used in making informed decisions. However, each measurement performed by a sensor has an associated uncertainty in its value, which if not accounted for properly, could potentially derail the decision process. Computing and embedding the associated uncertainties with data are therefore crucial to providing reliable information for sensor-based applications. In this paper, we present a novel unified framework for computing uncertainty based on accuracy and trust. We present algorithms for computing accuracy and trustworthiness and also propose an approach for propagating uncertainties. We evaluate our approach functionally by applying it to data sets collected from past deployments and demonstrate its benefits for in-network processing as well as fault detection.

Index Terms—Accuracy, sensing uncertainty, trust.

I. INTRODUCTION

SENSORS have become essential elements of data-driven real-life applications such as those of smart cities. For example, in a smart city [1], data is generated by many and different types of sources such as websites (including social networks), CCTV cameras, mobile phones, and specifically deployed sensors etc. These sources produce data with different levels of accuracy and trustworthiness. The data could further be processed in order to produce information that can be used for making real-time decisions (e.g., road traffic information based on traffic cameras, crowd sensed data and weather sensors). The sheer volume of data as well as the distributed nature of data sources make in-network processing such as on-the-fly summarisation, aggregation and fusion necessary to handle the data volume [2]. However, in-network processing performed without considering the uncertainty associated with the input data results in erroneous output thereby leading to incorrect decisions.

Although uncertainty in measurement caused by internal and external factors is not exclusive to sensor-based measurements, the effect is more pronounced for several reasons. Sensors are less precise and more fault prone than high precision and more durable but expensive measuring instruments. In addition, as they are embedded in the real world including underground, underwater and other harsh environments, the impact of external factors is more significant. Also, the increasing ubiquity of large-scale sensor networks, while enabling monitoring and event detection at an

unprecedented scale, has decoupled the data consumer and the data producer. This, its benefits notwithstanding, will introduce an information gap about the capabilities of sensors (e.g., precision, lifetime etc.) and loss of the implicit trust associated with small-scale deployments. In small-scale networks, the data consumer usually undertakes the deployments and also has information about the environment of deployment and the characteristics of every sensor. This can be used to check the data in real time or offline before a decision based on the data is made. However, this is neither feasible nor efficient in large-scale sensor networks such as those of smart cities. Hence, unless metrics pertaining to the precision, accuracy and trustworthiness of data are produced at the time of measurement and gathered with the data, they may never be available later on thereby rendering the data incomplete. Since communication is costly in sensor networks, integrating these metrics into one metric that indicates uncertainty is useful as it will give us the benefit without significant communication overhead.

Sensor and actuator networks may need to make real-time decisions autonomously based on the sensed data before the data reaches the base station and is checked for accuracy. A metric indicating the uncertainty of the data will facilitate these decisions. Consider a sensor-network assisted water distribution system. Monitoring drinking water quality to detect accidental or deliberate contamination is an important part of water distribution systems [3]. Advances in sensor devices have enabled monitoring water quality parameters such as pH, flow, turbidity, temperature and conductivity at the source or along the distribution system. Similarly, advances in actuator devices have enabled the use of isolation valves that could be activated by contamination. Based on the data from this monitoring, a water distribution network might detect contamination and decide in real time to isolate contaminated segments of the network. The correctness of this important decision depends on the correctness (quality) of the data which in turn depends on the accuracy and trustworthiness of the data obtained from the sensors. Data obtained from a trusted and properly functioning sensor comes with some uncertainty due to the internal behaviour (limitations) of the sensor as well as external factors. A measure of certainty taking these issues into consideration is therefore crucial for building applications that make use of the data generated by sensors.

The rest of the paper is organised as follows. In section II, we discuss the different types of errors in measurement and the metrics required to capture these errors. In section III, we present our proposed framework for computing a measure of uncertainty based on these metrics. We then present results of functional evaluation in section IV and conclude.

The authors are with the Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK. E-mail: {e.asmare, j.mccann}@imperial.ac.uk.

Copyright (c) 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

II. ERRORS IN MEASUREMENT

Errors in measurement can be broadly classified into three types – (1) Random Errors, (2) Systematic Errors, and (3) Mistakes [4], [5].

Fig. 1: Measurement errors.

1) *Random Error*: A random error is caused by uncontrollable and small changes inside the measuring device or the environment in which the measuring device is placed. It is depicted by an irregular and unavoidable dispersion in the value of the measured quantity, as shown in Fig. 1, about the true value [5], [6]. Given a measuring instrument and a constant physical quantity (e.g., a temperature sensor and the temperature of a room that is known to have a constant temperature for a certain period of time), a repeated measurement of this physical quantity with the same measuring instrument might result in different values scattered over a range. This variation of values is referred to as random error. The range of variation is narrow for a precise measurement¹, as shown in Fig 2. Random errors pertaining to a measurement of a physical quantity such as temperature can be quantified statistically by treating every measurement as a random sample from a probability distribution. The distribution of measurement results for most physical quantities as well as the distribution of the mean of repeated or multiple simultaneous measurements of a single physical quantity are well described by the normal probability distribution [4].

2) *Systematic Error*: A systematic error refers to a non-random shift in the value of the measured quantity from the true value, as shown in Fig. 2. This type of error can be caused by miscalibration and other various reasons unknown to the measurer. A smaller systematic error results in a more accurate measurement². Systematic errors³ from results of a single measuring device cannot be quantified statistically. However, comparison of measurement results of the same quantity from different measuring devices can be used in detecting and quantifying systematic errors.

3) *Mistakes*: Mistakes (faults) are errors that are caused by malfunctioning of the measuring device. Mistakes are very difficult to detect as well as quantify. Detecting faults in event detection applications, where novel data is expected, is especially difficult as both mistakes and interesting events appear to be anomalous [8]. In a similar manner to systematic errors, comparison of measurements can be used to detect faults.

¹A precise measurement is where the spread of results is small, either relative to the average result or in absolute magnitude [4].

²An accurate measurement is one in which the results of the measurement are in agreement with the accepted or true value [4].

³A generalised Gaussian error calculus takes unknown systematic errors as components of random errors [7].

In our context, random errors affect the precision of sensor data. Precision refers to the variability of the value of a measured quantity when measured repeatedly [6]. Values exhibiting high variability are said to be less precise than those exhibiting low variability. Systematic errors and mistakes affect both accuracy and trustworthiness of data. Thus, a metric involving precision, accuracy and trust can capture the three types of errors. We refer to this metric as uncertainty since it characterises the dispersion of measured values due to various types of errors in measurement. Accuracy refers to the closeness of the value of a measured quantity to the true value [6]. When the value is close to the true value, it is said to be accurate. Trust refers to various notions depending on the field of applications [9]. Mui et al. [10] integrated various concepts and relationships pertaining to trust and reputation, and proposed a beta distribution based computation model. They define reputation as a *perception that an agent creates through past actions about its intentions and norms*, and trust as a *subjective expectation an agent has about another's future behaviour based on the history of their encounters*. In this paper, we use this notion of trust. It is also worth noting that the trust notion in this paper is pertaining to faulty sensors (nodes) only and not malicious ones. Fig. 2 depicts the relationship between precision, accuracy, the measured value (M_v) and the true value (T_v) using normally distributed measurement results.

Fig. 2: Precision and accuracy.

III. PROPOSED FRAMEWORK

Determining uncertainties associated with measurement results as well as propagating these uncertainties, i.e., computing the combined uncertainties associated with quantities that are computed from measurement results, is a well established practice in physical sciences involving measurements. We argue that this is essential but lacking in wireless sensor networks and propose a framework, shown in Fig. 3, that employs precision, accuracy and trust in order to compute and propagate the uncertainty in sensor data. Each sensor node will be equipped with this framework thereby distributing the process of computing uncertainty.

Fig. 3: Sensor data uncertainty computation framework.

The precision and accuracy of a sensor are periodically computed by the sensor itself. However, the trust and uncertainty of a sensor are computed by a one-hop neighbouring sensor that transacts with it (e.g., aggregating data obtained from the sensor). Trust is computed periodically by the neighbouring sensor. Uncertainty is computed on demand whenever the neighbouring sensor includes the data obtained from the sensor in further transactions.

We illustrate our approach using the example network shown in Fig. 4. In this figure, p , a , u and d represent precision, accuracy, uncertainty and data respectively. Sensors 1 and 2 with precisions p_1 and p_2 compute the accuracy of their measurements using these precision values and other quantities (discussed later on). They, then, send their measurement results with associated accuracy values to sensor 3. Sensor 3, which regularly computes trust values (not shown in the figure) associated with these neighbouring sensors, uses d_1, d_2, a_1, a_2 and the trust values (possibly including its own data and metrics) to produce data d_3 with uncertainty u_3 . This result is then sent to sensor 6 which also transacts with the other sensors. Sensor 6 uses the data and uncertainty values from sensors 3 and 9 (and possibly other sensors) to produce data d and the combined (propagated) uncertainty u using the appropriate uncertainty combination rule (discussed later on).

Fig. 4: Computing and propagating uncertainty in sensor networks.

In the following, we describe the functions of each unit of the uncertainty computation framework.

A. Updating the Precision of a Node

The precision of a measuring device can be found from a device's data sheet which could be electronic as the IEEE 1451 series of transducer (sensors and actuators) standards include the Transducer Electronic Data Sheet (TEDS) as part of the data format specifications. Each measurement can be considered as taking a random sample from a parent normal distribution with mean μ , which is the true value of the

measured quantity, and standard deviation σ . In the absence of systematic errors and faults, the mean value estimates the true value of the measured quantity, and the associated standard deviation for N measurements, σ_{N-1} , determines the uncertainty in the value of the measured quantity.

For a new (newly deployed) sensor, the precision value is supposed to be the same as (or smaller than) this standard deviation. However, as the sensor gets older (or as time since deployment increases) the precision of the sensor degrades [8], [11] and consequently the sensor becomes less precise than its data sheet would suggest. To account for this degradation, the Precision Unit performs multiple ($Measurement_{number}$) short-period ($Measurement_{period}$) successive measurements, computes (Algorithm 1) the relative standard deviations, and updates the precision with the minimum of the set of relative standard deviations if the minimum relative standard deviation is greater than the current precision. The rationale behind performing multiple computations of standard deviations is to decrease the probability of misinterpreting actual rapid variations caused by the measured quantity as degradation of precision.

Algorithm 1 Update Precision

Require: $Factory_{precision}$, $Measurement_t$, $Measurement_{period}$, $Measurement_{number}$
Ensure: $Precision$
 $Precision = Factory_{precision}$
for $n = 1$ **to** $Measurement_{number}$ **do**
 $MeasurementLength = 0$
 $SumValues = 0$
 $SumSquareValues = 0$
 for $t = 0$ **to** $Measurement_{period}$ **do**
 $SumSquareValues += Measurement_t^2$
 $SumValues += Measurement_t$
 $MeasurementLength ++$
end for
 $\sigma_n^2 = \frac{SumSquareValues - \frac{SumValues^2}{MeasurementLength}}{MeasurementLength - 1}$
 $\mu_n = \frac{SumValues}{MeasurementLength}$
 $wait(Measurement_{period})$
end for
if $(\sigma^2 = Min(\sigma_n^2)) > Factory_{precision}^2$ **then**
 $Precision = \frac{\sigma}{\mu}$
end if

B. Computing the Accuracy of a Node

While precision can be independently determined for each sensor, accuracy is more difficult to do so as it needs a ground truth in order to evaluate the measured value against the true value. Since there is no practical way of knowing the true

value of a measured quantity in sensor networks, two broad types of approximations are used to estimate the ground truth – modeling the measured quantity or consensus among nearby sensors [11]. We chose the latter approach as it is more general with respect to the type of measured quantity and can be used for both accuracy and trust computations. This choice limits the applicability of our accuracy and trust computation methods to classes of sensor network applications where there is some spatial and temporal redundancy. However, the uncertainty computation framework can still be used by replacing the accuracy and trust computation algorithms.

In our approach, the mean of the sensor's neighbours measurement results is considered as the true value of the measured quantity and the sensor's accuracy is estimated based on its deviation from this mean. The error in estimating the true value with the mean is computed as $\frac{\sigma_{N-1}}{\sqrt{N}}$. This error which is also referred to as the standard error or the error in the mean can be used when the measurements of one quantity are performed independently by more than one measuring device (e.g., measurement of one quantity by neighbouring sensors). The sensor observes a sequence (N) of its neighbours' measurement results and computes the following statistical parameters for each neighbour j : the mean ($\bar{x}_j$), the variance (S_j^2) and the standard error of the mean (ϵ_j). From these parameters, it produces an estimate of the true value ($TrueValue$) of the measured quantity by computing a weighted mean as follows:

$$TrueValue = \frac{\sum_j w_j \bar{x}_j}{\sum_j w_j} \text{ where } w_j = \frac{1}{\epsilon_j^2}$$

The weights (standard errors of the mean values) are computed from the variances as follows:

$$\epsilon_j^2 = \frac{S_j^2}{N}$$

The standard error of the weighted mean (ϵ_{tv}) is computed from the weights using the following relation:

$$\frac{1}{\epsilon_{tv}^2} = \sum_j \frac{1}{\epsilon_j^2}$$

Once it establishes the true value, the sensor compares it to the mean ($\bar{x}_i$) of its measurements. The result of this comparison, i.e., the absolute error (δ) is then used to compute the accuracy as follows:

$$Accuracy(x_i) = \begin{cases} 1 - MAX(Precision_i, \frac{\epsilon_{tv}}{TrueValue}) & \frac{\delta}{\epsilon_{tv}} \leq 3 \\ MAX(0, 1 - |\frac{\delta}{TrueValue}|) & \frac{\delta}{\epsilon_{tv}} > 3 \end{cases}$$

Assuming normally distributed measurement results⁴ with the true value as the mean, 99.7% of the measurement results should lie within three standard errors of the true value estimate. If the sensor's measurement result is outside three standard errors, we assume the presence of systematic errors significant enough to shift the mean. Hence, the measurement is considered as inaccurate (not in agreement with the true value) and the relative error is used to quantify the degree of (in)accuracy. For measurement results that lie within three standard errors, the measurements could be considered accurate. However, we took a more conservative approach and assign a degree of accuracy that takes the precision and relative standard error of the true value estimate into consideration. The rationale behind this is that the measurement result can

only be as accurate as the sensor's capability (with respect to precision) and the error in estimating the true value. The Accuracy Unit computes the accuracy as shown in Algorithm 2. The computation of the neighbours' statistical parameters is done in a similar way to Algorithm 1 and hence not shown here. The choice of confidence interval is application dependent and can be changed accordingly. Also, note that the precision value is obtained from the Precision Unit and it can be either the updated precision value or the manufacturer specified precision value depending on whether the sensor's precision has degraded or not.

Algorithm 2 Compute Accuracy

Require: $Precision_i, Measurement_i, Measurement_j$
Observation_{period}
Ensure: $Accuracy_i$
Observe $Measurement_j$ for
a period of $Observation_{period}$
Compute statistical parameters
 $\delta = |Measurement_i - TrueValue|$
 $RelativeError = |\frac{\delta}{TrueValue}|$
 $RelativeStandardError = |\frac{\epsilon_{tv}}{TrueValue}|$
 $Deviation = \frac{\delta}{\epsilon_{tv}}$
if $Deviation \leq 3$ **then**
 $Accuracy_i = 1 - \max(Precision_i, \epsilon_{tv})$
else
 $Accuracy_i = \max(0, 1 - RelativeError)$
end if

C. Computing the Trust Value of a Node

We use the beta probability distribution [12] for computing trust. The beta distribution is a continuous distribution with a probability density function f defined in a finite interval $[A, B]$ and parameters α and β as follows:

$$f(x) = \begin{cases} \frac{1}{B-A} \frac{1}{Beta(\alpha, \beta)} \left(\frac{x-A}{B-A}\right)^{\alpha-1} \left(\frac{B-x}{B-A}\right)^{\beta-1} & A \leq x \leq B \\ 0 & \text{otherwise} \end{cases}$$

where $Beta(x, y)$ known as the beta function is defined in terms of the gamma distribution [12] function Γ as $\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$. The expected value and the variance of a beta random variable are $A + (B-A)\frac{\alpha}{\alpha+\beta}$ and $\frac{(B-A)^2\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$ respectively. When $A = 0$ and $B = 1$, we get the standard beta distribution. Due to its finite interval, its flexibility and ability to capture proportions, the beta distribution is convenient for modeling reputation and trust and has been used to do so in a number of research work [11], [13].

We model a sensor node i 's trust for a neighbouring sensor node j , i.e., T_{ij} using the standard beta distribution, i.e., T_{ij} is a beta random variable in the interval $[0, 1]$. A higher trust value indicates more trust in a neighboring sensor node's measurement. A value of 1 indicates full trust and 0 indicates full distrust. For each measurement (sequence of measurements) by a sensor node i , the Comparator Unit observes the measurement (mean of measurements) of a neighbouring node j and classifies the neighbour's measurement as acceptable or not based on the similarity of the measurement value within the limits of a specified threshold ($Acceptance_{threshold}$). Using this binary classification, and the conjugate prior property of the beta distribution, a Bayesian approach is used to compute a new (posterior) beta distribution from the prior beta

⁴The mean of measurements is also normally distributed due to the Central Limit Theorem.

distribution. The prior values for α and β are set to 1 initially to simulate uniform distribution of trust for sensor nodes that do not have recent observation experiences. The expectation of the posterior distribution ($E[T_{ij}]$) is used as the trust value.

Algorithm 3 Compute Trust

Require: $Acceptance_{threshold}, Measurement_i, Measurement_j, Observation_{length}$
Ensure: $Trust_{ij}$
 Observe $Measurement_j$ for $Observation_{length}$ number of measurements
 $\alpha_{prior} = 1$
 $\beta_{prior} = 1$
 $Acceptable_m = 0$
 $Unacceptable_m = 0$
 Compute $Trust_{ij}$ for each observation
if $|Measurement_i - Measurement_j| > Acceptance_{threshold}$ **then**
 $Unacceptable_m = Unacceptable_m + 1$
else
 $Acceptable_m = Acceptable_m + 1$
end if
 $\alpha_{posterior} = \alpha_{prior} + Acceptable_m$
 $\beta_{posterior} = \beta_{prior} + Unacceptable_m$
 $Trust_{ij} = \frac{\alpha_{posterior}}{\alpha_{posterior} + \beta_{posterior}}$

D. Computing Uncertainty

Given a sensed data, a data consumer might be uncertain about whether the data is accurate or not and whether the source of the data is trustworthy or not. Suppose the systematic error in a particular measurement is negligible and the measuring device is trusted. In this case, the accuracy of the measurement is determined by the precision of the measuring device, i.e., the data consumer can be certain that the sensed data is as accurate as the sensor's capability permits. Consequently, the uncertainty is estimated by the accuracy of the measuring device [5]. On the other hand, for cases of degraded precision, significant systematic error and less than full trust (trust value 1), the certainty is discounted by the accuracy as well as trustworthiness of the measuring device. To capture this notion, we model uncertainty as follows:

$$Uncertainty(x_j) = 1 - T_{ij} * Accuracy(x_j)$$

The Uncertainty Unit uses this model and the values obtained from the Accuracy and Trust units to compute the uncertainty of a sensor node. It should be noted that while the precision and accuracy values of a sensor are computed by itself, trust and uncertainty are computed by a neighbouring sensor that is dealing with the sensor (e.g., aggregating data obtained from the sensor). Also, the precision, accuracy and trust computations involve parameters – $Measurement_{period}, Observation_{period}, Acceptance_{threshold}$ respectively – whose values should be chosen in application dependent manner. The measurement period ($Measurement_{period}$) for computing precision should be long enough to allow making enough measurements for computing the statistical parameters but not too long so as to avoid capturing actual variations of the measured quantity. The same applies to the observation period ($Observation_{period}$) for

TABLE I: Rules for combining uncertainties.

f	ϵ_f
$f(x, y) = kx + my$	$\epsilon_f = \sqrt{k^2(\epsilon_x)^2 + m^2(\epsilon_y)^2 + 2mk(\epsilon_{xy})}$
$f(x, y) = \frac{x}{y}$	$\epsilon_f = f\sqrt{(\frac{\epsilon_x}{x})^2 + (\frac{\epsilon_y}{y})^2 - 2(\frac{\epsilon_{xy}}{xy})}$

computing accuracy. However, given a $Measurement_{period}$, while the Precision Unit can get as many measurements as the maximum sampling frequency supported by the sensor would allow, the Accuracy Unit may not be able to do so within the observation period ($Measurement_{period}$) since its measurements come from passive observation of neighbouring sensor nodes.

E. Propagating Uncertainty

The measurement of a physical quantity to determine its numerical value and estimating the associated uncertainty with that value is an important part of experiments in physical sciences. Consequently, the propagation of uncertainty is a well studied concept in these fields. Since sensors are (or can be viewed as) measuring devices, the uncertainty of data generated by sensors can be propagated by applying the methods used in measurement uncertainty analysis. In this work, we adopt these methods for propagating uncertainties in sensor networks.

In addition to computing the uncertainty of a neighbouring node, the Uncertainty Unit also deals with combining (propagating) uncertainties of sensed data if it processes data with uncertainty values received from other sensors (e.g., in-network processing). Consider a multi-variable function $f(x, y, z, \dots)$ with independent variables. The error in f , denoted by ϵ_f , as a result of the errors in $x, y, z, \dots$ denoted by $\epsilon_x, \epsilon_y, \epsilon_z, \dots$ respectively can be approximated as follows [4]:

$$(\epsilon_f)^2 = \left(\frac{\partial f}{\partial x}\right)^2 (\epsilon_x)^2 + \left(\frac{\partial f}{\partial y}\right)^2 (\epsilon_y)^2 + \left(\frac{\partial f}{\partial z}\right)^2 (\epsilon_z)^2 + \dots$$

Hence, once the uncertainty in each independently measured quantity is approximated (by the Uncertainty Units of each sensor node), the uncertainty in a combined quantity (e.g., the difference of two measurements) can be computed using this relation and the uncertainties of the component quantities. As an example, Table I shows rules for combining uncertainties in simple multi-variable functions where ϵ_{xy} is the covariance between x and y . These rules could be specified in the Uncertainty Unit.

IV. EVALUATION

We test our scheme using two data sets from the Grand Saint Bernard [14] and Victoria & Albert Museum (of London) deployments.

1) *Grand Saint Bernard*: The Grand Saint Bernard (GSB) deployment consisted of 17 Sensorscope stations [15] along a 900 metre path at the Grand Saint Bernard Pass (a mountain pass located between Switzerland and Italy). The deployment measured a number of environmental quantities every two

minutes for a month and a half. We selected four sensor nodes (nodes 6, 11, 13 and 14) that were placed relatively near to each other (the maximum Euclidean distance in space between any one of the nodes was less than 350 meters).

Fig. 5: Temperature measurement by four Sensors – GSB.

Fig. 5 shows ambient temperature measured by the four sensors for one month. In [14], the authors reported that node 11 had suffered a short circuit and had to be repaired on site. It can also be observed from this plot that node 6 gives faulty data in the first few days of the month. To test whether our framework can detect and account for these faults, we ran the trust computation algorithm for this data set and show the trust values for each node as computed from node 13 in Fig. 6. We can observe that, in line to what was reported from the deployment (for node 11) and observed from the data (for node 6), the trust values of nodes 6 and 11 decrease significantly immediately after the deployment and then slowly rise afterwards thereby capturing both the fault and recovery. Once a node loses its higher trust level, it takes time for it to reach this level again after recovering from a fault (or being replaced with a new node that has the same ID). This is so because reputation-based trust cannot be gained instantly.

Fig. 6: Trust values computed by node 13 for nodes 6, 11 and 14 with $Acceptance_{threshold} = 1.5$.

To test the applicability of our uncertainty computation framework for in-network processing, we chose node 13 as an aggregator node and computed the average temperature using the measurement results of the four sensors. First, we show the aggregation in the absence of fault in order to establish a comparison point. Fig. 7 shows (for one day) the average temperature as computed by node 13. As can be observed

from the figure, the measurement results from the four sensors follow a similar trend and that is also reflected by the average.

Fig. 7: Average temperature computed by node 13.

We then introduced a Gaussian noise of standard deviation 0.3 and mean 5 to the data from sensor 14. The choice of the standard deviation for the noise was based on the precision of the temperature sensors [15] used by the deployment. The result of the aggregation as computed by node 13, without using the uncertainty framework, is shown in Fig. 8. As can be observed from the figure, the average is distorted by the noisy sensor since node 13 did not take the accuracy and trustworthiness of data from sensor 14 into consideration. It implicitly considered the data as accurate and also trusted it with the same level as the non-faulty sensors thereby producing a distorted output.

Fig. 8: Average temperature computed by node 13 with noisy sensor 14 – without using the uncertainty framework.

Keeping the same level of noise, a weighted (by uncertainty) average temperature was recomputed by node 13 using the uncertainty computation framework. The result is shown in Fig. 9. This time the average temperature is not distorted by the noisy sensor as the accuracy and trustworthiness of the data from sensor 14 were computed and taken into consideration by the uncertainty computation framework. The difference is more visible in Fig. 10. The root mean square deviation of the average computed by the uncertainty framework is 87% less than the one computed without the framework. These results demonstrate how the uncertainty framework can be used in in-network processing.

2) *Victoria and Albert Museum*: Victoria and Albert museum has deployed a wireless sensor network for monitoring

Fig. 9: Average temperature with noisy sensor 14 – using the uncertainty framework.

Fig. 10: Average temperatures.

temperature and humidity in different sections of the museum. The data collected from this long-term deployment consists of measurement results from 78 temperature and humidity sensors. The measurement was performed every 15 minutes over two years. In [16], the authors have identified a pair of neighbouring sensor nodes – nodes 127 and 128 – that should have exhibited very similar readings but did not do so over a long period of time due to node 127 drifting consistently up to a difference of 2°C – 3°C . Fig. 11 shows a plot of the daily average measurements of temperature for 110 days (days 378-488). Node 128 was replaced by a new node with the same ID a couple of months ago before day 378 and node 127 was replaced on day 448.

We ran our trust computation algorithm, as computed from node 128, for the two year trace data. Fig. 11b shows a plot of the trust values for days 378-488. It can be observed that the trust value of node 127 as computed by node 128 decreases consistently until the day of replacement and it starts increasing afterwards. This result demonstrates how the framework can be used to detect when sensors start to fail.

V. RELATED WORK

Klein and Lehner [17], [18] propose data quality measures divided into two types namely intrinsic and contextual data quality. Intrinsic measures are those that characterise a single data item (e.g., accuracy) and contextual measures are those that characterise datasets (e.g., data volume). A sensor data stream is partitioned into consecutive non-overlapping windows and the data quality measures are computed for

(a) Temperature measurements.

(b) Trust values.

Fig. 11: Temperature measurement by nodes 127 & 128, and trust values computed by node 128 for node 127 with $Acceptance_{threshold} = 1$.

each window. The window accuracy in their framework is equivalent to our accuracy metric. However, their computation of the accuracy of a single data item, from which the window accuracy is computed, is based only on the precision class (manufacturer specified precision) of the sensors. For example, a sensor node that is stuck at some value due to a fault would be considered as reporting an accurate measurement since the accuracy metric is based only on precision.

Ganerwal et al. [11] developed a reputation-based system for computing trust in wireless sensor networks. In their system, a node observes the actions of its neighbours (broadcast neighbours) and classifies the observations to a value in the range $[0,1]$ referred to as cooperation metrics. This is used as a measure of the observer's node agreement to the observed node. The system is comprised of two components namely Watchdog and Reputation System, which rate the input data and compute the reputation based on the rating respectively. The reputation-based system can deal with the uncertainty in sensor data pertaining to trustworthiness. However, it does not take accuracy and its impact on uncertainty into consideration.

In [19] a statistical trust computation approach for wireless sensor networks is presented. In this system, a node observes the behaviour of another node with respect to a certain context (e.g., ability to sense accurately, routing etc.) and records this observation. The record consists of an identifier for the observed node (e.g., unique ID, location etc.) and the observing node, the context type of the observation, a timestamp, a trust value associated to the trustworthiness of the observed node and a weight associated to the length or

amount of observation. The observations are further weighted by their age and a weighted mean and variance are calculated. A confidence interval around the mean is created and if the confidence interval is narrow enough the mean is used as the trust value, otherwise more records are collected and the process is repeated. This approach deals with trust computation and propagation and can be used to partially capture the uncertainty in sensor data. It, however, is not suitable for real-time applications as the trust computation can take a long time while waiting for enough experience records to attain a specified confidence interval.

Work in the area of truth discovery and fact finding [20], [21] is also related to sensor data uncertainty. Wang et al. [22] proposed an optimal approach for determining the correctness of measurements and the reliability of the measurement sources using maximum likelihood estimation. This approach is targeted for crowd-sourced binary measurements (e.g., determining whether something measured/reported is true or not). Wen et al. [23] developed a learning technique for estimating the accuracy of co-located indoor positioning systems. In [24], the authors present a thorough comparison of learning and inference techniques with respect to their application for accuracy estimation. The Accuracy Unit of our framework can be enhanced by incorporating these approaches.

VI. CONCLUSION

In this paper, we have presented a framework for computing uncertainties in sensor data. Our approach is novel in that the uncertainty computation considers precision, accuracy and trust in a unified manner. It computes a single metric for uncertainty thereby adding an essential metric to sensor data with minimal communication overhead. Since both accuracy and trust are computed through overhearing of broadcast neighbours, there is also no additional communication overhead pertaining to gathering the necessary information for uncertainty computation. The accuracy and trust computations depend on the number of neighbours. However, since only single-hop neighbours are considered the cost will not be significant. Also, the computations involving trust, including expectations for the probability distribution, involve only simple arithmetic. Consequently, the power overhead is less significant. In the future, we plan to test our framework on ICRI's (Intel Collaborative Research Institute) deployments of environment and air pollution monitoring sensor networks at Hyde Park and Enfield (both in London) respectively.

REFERENCES

- [1] L. Sanchez, J. Galache, V. Gutierrez, J. Hernandez, J. Bernat, A. Gluhak, and T. Garcia, "SmartSantander: The meeting point between future internet research and experimentation and the smart cities," in *Future Network Mobile Summit (FutureNetw)*, 2011, June 2011, pp. 1–8.
- [2] M. Balazinska, A. Deshpande, M. J. Franklin, P. B. Gibbons, J. Gray, S. Nath, M. Hansen, M. Liebholt, A. Szalay, and V. Tao, "Data management in the worldwide sensor web," pp. 30–40, 2007.
- [3] M. V. Storey, B. van der Gaag, and B. P. Burns, "Advances in on-line drinking water quality monitoring and early warning systems," *Water Research*, vol. 45, no. 2, pp. 741 – 747, 2011.
- [4] I. Hughes and T. Hase, *Measurements and their uncertainties: a practical guide to modern error analysis*. Oxford University Press, 2010.
- [5] H. J. Berendsen, *A student's guide to data and error analysis*. Cambridge University Press Cambridge, 2011.
- [6] L. Kirkup and R. Frenkel, *An introduction to uncertainty in measurement using the GUM (Guide to the expression of uncertainty in measurement)*. UK: Cambridge University Press, 2006.
- [7] M. Grabe, *Generalized gaussian error calculus*. Springer, 2010.
- [8] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sen. Netw.*, vol. 5, no. 3, pp. 25:1–25:29, Jun. 2009.
- [9] D. H. McKnight and N. L. Chervany, "Trust and distrust definitions: One bite at a time," in *Proceedings of the Workshop on Deception, Fraud, and Trust in Agent Societies Held During the Autonomous Agents Conference: Trust in Cyber-societies, Integrating the Human and Artificial Perspectives*. London, UK, UK: Springer-Verlag, 2001, pp. 27–54.
- [10] L. Mui, M. Mohtashemi, and A. Halberstadt, "A computational model of trust and reputation for e-businesses," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7 - Volume 7*, ser. HICSS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 188–.
- [11] S. Ganerwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 3, pp. 15:1–15:37, Jun. 2008.
- [12] J. Devore, *Probability & Statistics for Engineering and the Sciences*. Cengage Learning, 2012.
- [13] M. C. Huebscher and J. A. McCann, "A learning model for trustworthiness of context-awareness services," in *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*. IEEE, 2005, pp. 120–124.
- [14] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "Wireless sensor networks for environmental monitoring: The sensorscope experience," in *Communications, 2008 IEEE International Zurich Seminar on*, March 2008, pp. 98–101.
- [15] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 17:1–17:32, Mar. 2010.
- [16] T. Bourdenas, M. Sloman, and E. Lupu, "Self-healing for pervasive computing systems," in *Architecting Dependable Systems VII*, ser. Lecture Notes in Computer Science, A. Casimiro, R. Lemos, and C. Gacek, Eds. Springer Berlin Heidelberg, 2010, vol. 6420, pp. 1–25.
- [17] A. Klein, "Incorporating quality aspects in sensor data streams," in *Proceedings of the ACM First Ph.D. Workshop in CIKM*, ser. PIKM '07. New York, NY, USA: ACM, 2007, pp. 77–84.
- [18] A. Klein and W. Lehner, "Representing data quality in sensor data streaming environments," *Journal of Data and Information Quality*, vol. 1, no. 2, pp. 1–28, 2009.
- [19] M. Probst and S. Kasera, "Statistical trust establishment in wireless sensor networks," in *Parallel and Distributed Systems, 2007 International Conference on*, vol. 2, 2007, pp. 1–8.
- [20] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *COLING*, Beijing, China, 8 2010.
- [21] D. Wang, T. Abdelzaher, H. Ahmadi, J. Pasternack, D. Roth, M. Gupta, J. Han, O. Fatemeh, H. Le, and C. Aggarwal, "On bayesian interpretation of fact-finding in information networks," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, July 2011, pp. 1–8.
- [22] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, ser. IPSN '12. New York, NY, USA: ACM, 2012, pp. 233–244.
- [23] H. Wen, Z. Xiao, N. Trigoni, and P. Blunsom, "On assessing the accuracy of positioning systems in indoor environments," in *Proceedings of the 10th European Conference on Wireless Sensor Networks*, ser. EWSN'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 1–17.
- [24] H. Wen, Z. Xiao, A. Symington, A. Markham, and N. Trigoni, "Comparison of accuracy estimation approaches for sensor networks," in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, May 2013, pp. 28–35.