Using Data Mining to Handle Missing Data in Multi-Hop Sensor Network Applications

Le Gruenwald Hanqing Yang Md. Shiblee Sadik Rahul Shukla
School of Computer Science,
University of Oklahoma
Norman, USA
{ggruenwald, hqyang3, shiblee.sadik, rahul.shukla-1}@ou.edu

ABSTRACT
A sensor's data loss or corruption, aka sensor data missing, is a common phenomenon in modern wireless sensor networks. It is more severe for multi-hop sensor network (MSN) applications where sensor data reach the base station via other sensors; hence a sensor's failure can cause multiple missing data. In this paper we present MASTER-M, a data estimation framework based on data clustering and association rule mining to estimate the values of missing sensor data for MSN. Estimating, instead of resending, the missing sensor data is becoming popular as it may reduce query response time and sensor energy consumption; however the current works cater to only single-hop sensor networks. To fill this gap, our novel technique addresses the issues related to MSN, such as simultaneous missing sensors and missing spatially correlated sensors. It consists of three steps: 1) clustering sensors online; 2) capturing association rules between sensors inside each cluster, and 3) estimating the values of the missing data using the obtained association rules. Experimental results on both real-life sensor data and synthetic sensor data demonstrate the efficacy of MASTER-M in terms of estimation accuracy compared to the existing techniques. Moreover, we also present experiments showing the supremacy of data estimation by MASTER-M in terms of energy savings over re-transmission of missing data.

Keywords
Sensor Databases, Missing Data, Multi-hop Sensor Networks.

1. INTRODUCTION
Wireless sensor networks are deployed and utilized widely in environment monitoring [18], scientific investigation [12], roads, bridges, and civil structure flaw detection, battle surveillance, medical applications [19] and many other fields. However, under the current wireless sensor networks technology, a wireless sensor node is prone to hardware failures such as power shortages and mal-functioning of sensor nodes. In addition, network issues, such as connection interruption and package collision, cause further problems in data assimilation. Due to those reasons, sensor data may fail to reach the base station; we call such data missing data. The sensors which generate these missing data are called missing sensors throughout this paper.

MobiDE'10, June 6, 2010, Indianapolis, Iindiana, USA.

To compensate for the missing data, researchers have proposed several solutions, such as ignoring the missing data, querying the network again, using backup sensors, and estimating missing data [5]. Ignoring missing data is not a very efficient solution for sensitive applications and querying the network again is neither time efficient nor realistic. Using backup sensors is another expensive solution which may bring up issues such as dealing with data duplication. Thus estimation of the values of the missing data may be one of the optimal solutions.

Some research has been done for estimating missing data in sensor networks [17], [2], [13], but most of the existing research is designed for single hop sensor networks where sensors send data directly to the base station through a single hop communication. As the demand of deploying sensor networks in a broader scope emerges, more and more wireless sensor networks are configured in a multi-hop communication fashion. In these networks, as shown in Figure 1, sensors are usually placed far away from the base station, and the distant sensors use other intermediate sensors to route the data to the base station. Most existing solutions for missing sensor data estimation are not suitable for multi-hop sensor networks as they do not consider the newly emerging issues associated with these networks, which we describe below.

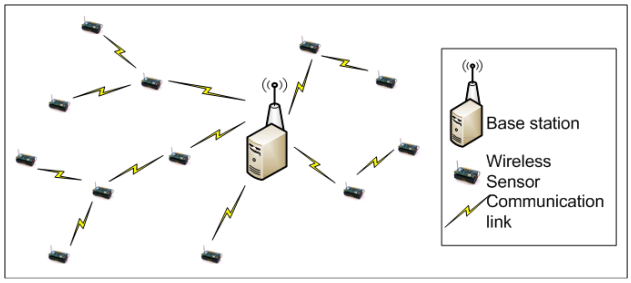

Figure 1. Multi-hop sensor networks architecture

Due to multiple hops in the routing path, in a multi-hop sensor network more missing data are generated compared to single-hop routing. Moreover, critical routing node failures may cause all messages routed by these nodes to miss simultaneously. Due to the large size and complexity of multi-hop networks, it is very difficult to predict how, when, and how many sensors will be missing.

Another issue related to sensor network is that the sensors close to the base station are more likely to die faster than the sensors far away from the base station; this situation is known as *black hole effect* [20]. Failures of the sensors close to the base station cause all traffic data which had been passing through the 'dead' sensors

previously to be missing. The problem is more severe in multi-hop networks; therefore, missing data estimation solutions in these networks need to be more robust with respect to the number of missing sensors.

Finally, single hop sensor networks are typically designed to monitor small-scale, stable phenomena, while multi-hop sensor networks are heterogeneous networks and designed for complex, large-scale, and changing phenomena. An example of the latter can be seen in the heat transferring application [15]. In this application, in a relatively big region, quite a few sensors are deployed to monitor and report temperature. There are multiple heat sources in a region, and complex isotherms can be formed. The heat resource might adjust the function to high or low level, and it might even move. All of these cause the temperature distribution in the region unstable and changeable, i.e., the phenomenon might keep changing in the environment that the multi-hop sensor networks are monitoring.

In this paper, we propose a robust algorithm, called MASTER-M, to estimate the values of the missing sensor data in multi-hop sensor networks. It takes the issues associated with these networks into consideration. The rest of the paper is organized as follows: Section 2 discusses the related works and issues to be addressed; Section 3 describes MASTER-M; Section 4 presents the experimental results comparing MASTER-M with existing techniques in terms of estimation accuracy and execution time for both real-life and synthetic test data; Section 5 discusses the energy savings of MASTER-M; and Section 6 provides conclusions and future research.

2. RELATED WORK AND ISSUES

The problem of estimating missing data has received growing attention from researchers in various disciplines, such as mathematics, engineering and computer science. Most of the early proposed approaches are statistical in nature like maximum likelihood [9], imputation by regression [14], and expectation maximization [11]. These pure and/or simplex statistics methods, however, cannot be directly applied for wireless sensor network data as they do not address the stream data properties like unbounded data volume and one-pass data scan. In addition, they make certain assumptions on the data sets, such as Missing at Random, that may not exist in sensor data applications [5].

TinyDB [10] is a popular querying system for wireless sensor networks. However, TinyDB does the data estimation for a missing sensor by averaging the readings of all other sensors in the current round of sensor readings. This approach works well if all sensors are supposed to report similar values but it fails to incorporate non-linear relationships among sensors. Because of this fact, TinyDB gives erroneous estimation when complicated relations exist among sensors or multiple sensors fail simultaneously. It is very difficult to discover the underlying trends and patterns for sensor data properly with this straightforward technique.

Papadimitriou et al [13] proposed an auto regression model to extract correlations and hidden variables among multiple streams by the aid of principal component analysis (PCA) named SPIRIT. This technique can dynamically detect changes in streams through a single scan without storing data values. With the auto regression model, SPIRIT can predict the missing value based on the previous rounds using hidden variables which are a summary of the data correlation among all the sensors; but SPIRIT completely ignores the reported values in the current round from other sensors. The accuracy of SPIRIT is restricted because it ignores the relationships among the data in the current round of the sensor readings.

Vijayakumar and Plale proposed a Kalman filter based approach for estimating missing data [17]. It uses a dynamic linear model and provides a very accurate prediction of missing data. However, this approach is completely based on the history information. The data distribution in a data stream changes over time [8], and in extreme cases, the history data from the same sensor will no longer be useful to predict the current data value when the phenomena substantially changed; hence this approach has limitations still.

In [5], a missing sensor data estimation algorithm called FARM was presented where the association rules among the sensors are mined and used to estimate the missing data. Establishing the correlations among the sensors makes FARM achieve better accuracy than other popular statistical estimation methods. In addition, FARM employs a novel data freshness framework, which not only takes the temporal aspect of the data into consideration for mining association rules but also implements a data compaction scheme to store a large amount of stream data. The main bottleneck of FARM comes from its limitation on establishing association rules among the sensors. It establishes association rules between two sensors based on the equality between their readings, i.e., only equivalent relationships are mined.

Mining Autonomously Spatio-Temporal Environmental Rules (MASTER) proposed by Chok and Gruenwald [2] is a comprehensive spatio-temporal association rules mining framework which provides both a competitive data estimation method and an exploratory tool to investigate the evolution of patterns of the sensor data in a single-hop sensor network. MASTER is very well equipped to discover any kind of association rules among the sensors. This framework includes a novel data structure called MASTER-tree which stores the history data for each sensor and represents the association rules among the sensors. An example of an association rule in MASTER is $S_1[10,20], S_2[40,90] \rightarrow S_3[30,40]$ where S_1, S_2 and S_3 are three sensors, and S_1 and S_2 are called the antecedent sensors and S_3 is called the consequent sensor of the rule. This rule implies when the sensor reading of S_1 is within 10 to 20 and the sensor reading of S_2 is within 40 to 90, the sensor reading of S_3 would be within 30 to 40. Each node in the MASTER-tree represents a sensor except the root node which represents an empty node; and each path/sub-path starting from the root node represents an association rule. Hence a MASTER-tree is capable of representing any kind of relationship among the sensors which participate in the MASTER-tree.

MASTER limits the number of sensors in one MASTER-tree by clustering the sensors into small groups and producing an individual MASTER-tree for each cluster. The advantage of the clustering step is twofold: 1) the clustering step arranges spatially co-related sensors into one cluster and 2) it limits the number of sensors in a MASTER-tree which restricts the exponentially large number of association rules into a more manageable number. As each data round arrives, MASTER finds the appropriate MASTER-tree for each sensor and updates the MASTER-tree based on the arrived sensor readings. At any particular time if a sensor reading is missing, MASTER finds the appropriate

MASTER-tree for the missing sensor and evaluates the support and confidence of the association rules where the missing sensor appears as consequent. MASTER finds the best association rule comparing the obtained support and confidence with the user defined minimum support and minimum confidence. Finally it uses the best association rule and the current sensor readings of the antecedent sensors in the best association rule to estimate the consequent sensor's reading. Interested readers are referred to [2] for further details.

MASTER was designed for single-hop sensor networks. It suffers the following deficiencies. The cluster formation step is solely based on spatial attributes, which causes poor performance for multi-hop sensor network where closely located sensors are more likely to be missing together, although it performs excellently on single-hop sensor networks where closely located sensor are not likely to be missing together. Moreover, the multi-hop sensor networks are usually targeted for complex, large-scale and dynamically changing phenomena where the relationship among the sensors changes over time. The cluster formation step is used in MASTER to restrict the search space for association rules; However in a dynamically changing environment, the static cluster formation step may suffer from not to have the related sensors in the same cluster; hence the cluster formation step should be dynamic and events aware. Nevertheless, in a multi-hop network, a failure of an intermediate sensor can cause a loss of multiple sensors' data; therefore if the clusters are fixed based on spatial attributes, there is a chance that all the sensors of a cluster would be missing together, which will result in the unavailability of the antecedent sensors to estimate the consequent sensor.

Motivated by the drawbacks of MASTER, in this paper we develop a new version of MASTER, called MASTER-M, for multi-hop sensor network applications. MASTER-M makes use of a dynamic clustering method that tackles the problems of simultaneously missing spatially correlated sensors and static location based cluster formation of spatially correlated sensors. Our new clustering approach dynamically adjusts the clusters with the change of the relationships between the sensors. Moreover MASTER-M is more robust with respect to the number of simultaneously missing sensors. The details of MASTER-M are presented in the next section.

3. THE PROPOSED MASTER-M

In this section, we first give an overview of MASTER-M, and then describe the details of its individual modules.

3.1 The MASTER-M Overview

MASTER-M consists of three major modules: online clustering, MASTER-tree projection, and Data estimation. It takes the following input: the stream sensor data including missing values, user-defined maximum number of sensors in each cluster, and user-defined number of rounds at which a phenomenon change (event) occurs.

In the clustering module, MASTER-M groups the sensors into some clusters based on our proposed distance function described in Section 3.2 to compute the distance between the sensors. Here the distance measurement is derived in a bootstrapping fashion, i.e., the initial distance value is computed using the first few rounds of data, and the consequent distance value is updated incrementally., A re-clustering procedure is invoked once the distance bound in a cluster does not hold any more. Another trigger for the re-clustering procedure is the user- defined number of rounds when a phenomenon change occurs. The MASTER-tree projection module computes the MASTER-tree for each cluster. The MASTER-tree is capable of holding the relationships among all the sensors in the cluster. Through the MASTER-tree, association rules along with their support and confidence are implicitly maintained on the fly when a sensor reading arrives. With the first two modules we keep an effective cluster structure for sensors, and within a cluster, we maintain an up-to-date MASTER-tree. Then in the final data estimation module, the valid association rules are derived from the MASTER-tree to produce accurate estimation results for the missing sensors' readings. This estimation procedure is iterative and adjusted progressively: the nearest known node is used to infer the missing sensor reading first; if the error margin from the produced results meets the minimum consequent spanning space (MCSS), which is supplied by the user, then output the estimation results; otherwise more known nodes according to the path order from the MASTER-tree are selected to refine the rules for estimation, i.e. adding the antecedent sensors to the rules and cutting down the search space for the consequent sensor, which is the missing sensor node.

We describe the steps in MASTER-M in details in the following sections. In Section 3.1, we present our novel online clustering technique for MASTER-M. Then in Sections 3.2 and 3.3, we describe the MASTER-tree projection module and data estimation module, respectively. These two modules are mainly inherited from the original MASTER approach [2]. We briefly explain the two modules for the completeness of our approach, MASTER-M.

3.2 The Clustering Module

In this section, we describe how the clustering module works.

3.2.1 Definitions and Preliminaries

At the beginning, we arrange all the sensors according to their data missing rates in a descending order. The missing rate is defined as $\text{missing rate} = \frac{\text{number of missing rounds}}{\text{total number of rounds}}$. Let $(S_1, S_2, S_3, \ldots S_n)$ be the sorted list of sensors after sorting them in descending order of their missing rates, i.e. Sensor S_n misses least often and sensor S_1 misses most often. Sensors with the highest missing rates will be the "seeds" of the clusters. The significance of a seed is twofold. For a clustering technique, careful seeding is usually important and helpful [1]. For data estimation, seeds are the most demanding nodes as they are most likely to miss. For each pair of sensors, S_i and S_j, we compute the distance between them. There are two types of distance between these two nodes: the standard deviation of the differences of the data readings $d_{SDOD}(S_i, S_j)$, and the simultaneously missing rate $d_{SMR}(S_i, S_j)$. $d_{SDOD}(S_i, S_j)$ shows the degree that S_i and S_j are related to each other. A relatively small $d_{SDOD}(S_i, S_j)$ implies a better correlation between S_i and S_j. $d_{SMR}(S_i, S_j)$ shows whether S_i and S_j tend to be missing simultaneously; a small $d_{SMR}(S_i, S_j)$ implies a small chance that the S_i and S_j are missing together. So $d_{SDOD}(S_i, S_j)$ and $d_{SMR}(S_i, S_j)$ both are very important for deriving association rules between S_i and S_j and estimating missing sensor data. Note that both distances between a sensor and itself is always zero i.e., $d_{SDOD}(S_i, S_i) = 0$ and $d_{SMR}(S_i, S_i) = 0$.

We further normalize $d_{SDOD}(S_i, S_j)$ and $d_{SMR}(S_i, S_j)$ to be the values between 0 and 1 and we name them $n_{SDOD}(S_i, S_j)$ and $n_{SMR}(S_i, S_j)$ respectively. These two distances form a two

dimensional geometric space for a sensor node (S_i) where $n_{SDOD}(S_i, S_j)$ is placed along the x-axis and $n_{SMR}(S_i, S_j)$ is placed along the y-axis. Each data point in the two dimensional space formed for S_i represents a sensor node (S_j) where the abscissa is $n_{SDOD}(S_i, S_j)$ and the ordinate is $n_{SMR}(S_i, S_j)$. The origin is composed of the sensor itself, i.e., the point $(0, 0)$ represents the sensor (S_i). The Euclidean distance ($d(S_i, S_j) = \sqrt{n_{SDOD}(S_i, S_j)^2 + n_{SMR}(S_i, S_j)^2}$) is measured from the origin to S_j. The distance is then characterized as a measurement of the priority/benefit of putting S_i and S_j into the same cluster. Now we establish a matrix of distances from each node to all other nodes. Note that the distance relationship is symmetric i.e., $d(S_i, S_j)$ and $d(S_j, S_i)$ are the same ($d(S_i, S_j) = d(S_j, S_i)$). Due to the symmetry of the distance function, we do not need the full matrix. The half matrix is defined as M,

$$M = \begin{pmatrix} 0 & d(S_1,S_2) & d(S_1,S_3) & \dots & d(S_1,S_n) \\ & 0 & d(S_2,S_3) & \dots & d(S_2,S_n) \\ & & 0 & \dots & d(S_3,S_n) \\ & & & \ddots & \vdots \\ & & & & 0 \end{pmatrix}$$

```
procedure initialClusterSetup
1    construct a sorted list of the sensors according to their missing rates: DS = {S1, S2, S3, …, Sn};
2    form a set of clusters C1, C2, C3, …, Cn where Ci = {Si} for i=1 to n;
3    loop until no change takes place
4      find the two closest sensors (Si, Si) (without losing any generality we can assume i < j);
5      find the cluster Ci  where sensor Si belongs to;
6      find the cluster Cj where sensor Sj belongs to;
7      if |Ci| + |Cj| < resource constraint (c)
8        merge (Ci, Cj);
9      end if;
10   end loop;
end procedure
```

Figure 2. The Initial Clustering Algorithm

3.2.2 The Initial Cluster Structure and Clustering Algorithm

Figure 2 shows the detailed algorithms for the initial cluster setup. The initial clustering algorithm starts with sorting the sensors according to their missing rates (line 1). In the next step we setup a set of clusters where each cluster contains only one sensor (line 2). In the third step the two nearest sensors that do not belong to the same cluster is identified (line 4) and their respective clusters are also obtained (lines 5 & 6). Merge the two clusters unless the sum of their size is greater than the resource constraint (*c*) [2] (lines 7 and 8). Step 3 is repeated until no merge operation can take place. Finally the algorithm outputs a set of clusters where each cluster contains no more than *c* number of sensors and two sensors in the same cluster are less likely to be missing together and more likely to be correlated.

3.2.3 Online Cluster Adjustment

In Figure 3 we describe the online cluster adjustment procedure. As each round of sensor readings (or each round for short) comes we compute the distances between the reported values of each pair of sensors and compute the number of simultaneously missing sensors if there is any sensor missing. We compute $n_{SDOD}(S_i, S_j)$, $n_{SMR}(S_i, S_j)$ and $d(S_i, S_j)$ (lines 2, 3 and 4) for each pair of sensors S_i and S_j from the rounds arrived since the cluster has formed. In the next step, for each cluster we evaluate the distance between every two sensors inside a cluster. If the distance between any pair is greater than 1, we identify the current cluster as an obsolete cluster where the standard deviation of difference and/or simultaneous missing rate changed substantially; hence we need re-clustering. The value 1 signifies either the correlation or the simultaneously missing rate among sensors in a cluster reaches the maximum limit. Concurrently we check if the number of rounds reaches a user-defined ceiling as the user who has domain knowledge may anticipate phenomenon changes occurring and the need of re-clustering. The re-clustering is done by invoking the initial cluster setup algorithm (line 12). By online adjustment we maintain the most correlated sensors in a separate cluster and the sensors that are more likely to be missing together in other clusters.

```
procedure onlineClusterAdjustment (each data round)
1    for each pair of sensors Si and Sj
2      compute n_SDOD(Si, Sj)
3      compute n_SMR(Si, Sj)
4      compute d(Si, Sj)
5    end loop
6    for each cluster
7      if the distance between any two sensors d(Si, Si) is greater than 1 or the number of rounds reaches the user defined number of rounds at which a phenomenon change occurs
8        needReCluster = true;
9      end if;
10   end loop;
11   if needReCluster
12     invoke initialClusterSetup();
13   end if;
end procedure
```

Figure 3. The Online Cluster Adjustment Algorithm

3.3 The MASTER-tree Projection Module

Providing a feasible data structure for storing and mining stream data is challenging as the volume of stream data is usually unbounded, and thus they cannot be completely stored due to practical storage restrictions. For stream data association rules mining purposes, a compact data structure which can hold and represent various kinds of relationships among objects effectively and efficiently is greatly desired. MASTER-tree is a state-of-the-art representation of such data structure.

The MASTER-tree is illumined by pattern tree, which was proposed to present arbitrary relationships among all Boolean itemsets [4]. A pattern tree can be equivalent to a spanning tree of

a binary hypercube structure which also catches all possible Boolean item relationships. Pattern tree has a computational exponential complexity thus is very expensive in terms of computation. So grouping items to a set of clusters and pruning the pattern tree or its equivalent hypercube lowers the computational complexity substantially. As the pattern tree favors one node (the right most leaf node) and extracts all relationships to other nodes from that node, it cannot handle all nodes fairly. The MASTER-tree is proposed to solve this issue: it combines all various pattern trees regarding each node and prunes the common paths in the resulting tree, then forms a new tree called MASTER-tree [2].

Specifically, in the MASTER-tree data structure, each tree node represents a sensor. The data distribution in one sensor node over a particular vector space is stored in each node. The complete vector space where the sensor readings may fall into is discretized into a finite number of cells. For each cell, an arbitrarily accurate data distribution function or probability distribution function can be represented by an infinite number of moments in statistical theory. In computational practice, only a finite number of moments plus elements counter are stored in the MASTER-tree nodes (a typical configuration is the first four moments). Elements counter is the number of sensor readings, the value of which belong to the cell associated with the corresponding MASTER-tree node. Now for each cell, a few moments are stored, and cells across nodes are linked following the MASTER-tree paths. These cells and links form a grid structure (GS). GS satisfies the compactness requirements as it does not grow along with data rounds because it only depends on the finite number of cells and the fixed number of nodes in a cluster. As the data distribution information and elements counter are stored in the nodes of the MASTER-tree, from that information and following the path (representing the relationships) among the nodes of the tree, antecedent nodes with a value over specified cells can infer consequent nodes value distribution, so we can claim that association rules are implicitly stored along the MASTER-tree paths between nodes.

The MASTER-tree projection module is to establish a MASTER-tree for each cluster when the initial clustering procedure or the re-clustering process happens, then to incrementally update GS as a new round of data comes in. By doing this, the up-to-date association rules between the sensors in a cluster are implicitly held to serve data analysis purposes.

3.4 The Data Estimation Module

This data estimation module produces the estimation result for the missing sensor (MS). It accomplishes the task in an iterative way. First the module obtains the prior distribution of MS from the MASTER-tree, i.e., the rule ø → MS (here ø means empty). If the rule satisfies the user-defined error margin (the MCSS) and the support and confidence thresholds, the rule holds and the estimation result is produced by taking the average of the prior distribution of MS; . If it is not the case, i.e. the error margin requirement is not satisfied, the related information from other nodes needs to be considered to refine the estimation. The data estimation module chooses one more new antecedent node to infer the MS's reading. The initial relevant subspace for the antecedent node is simply the cell picked up based on its current reading. If the actual support does not satisfy the minimum support threshold, the relevant subspace is augmented iteratively until the actual support is not less than the minimum support. If it is not the case again, i.e. the support requirement cannot be satisfied even thought the relevant space reaches its limit, which is the complete subspace, the module takes this node away and switch to try a new prior node. The process of adding a new antecedent node repeats until the estimation procedure meets one of these two cases: (1) a rule satisfying all requirements as we showed above, or (2) no more node within the cluster is to be added to the antecedent nodes set. The procedure then returns the estimated value using the last expected value (the average) over the obtained consequent subspace.

4. EXPERIMENTAL DESIGN AND ANALYSIS

In this section, we compare MASTER-M with two existing algorithms: SPIRIT [13] and TinyDB [10].

4.1 Experimental Dataset

We perform our experiments based on one real-life dataset and one synthetic dataset which we describe in the next two sections.

4.1.1 Intel Berkeley Lab Data

This real life application dataset is from the Intel Berkeley Lab. It contains environmental readings collected between February and April in 2004 in an indoor setting [7]. The dataset was collected using a multi-hop sensor network consisting of 54 sensors (Mica2Dot). Each sensor detects the temperature of the floor. The number of hops and the network topology for the dataset change dynamically based on TinyDB [10]. The total number of rounds collected for all the sensors are approximately 65,000 ([7]). Some random sensors' readings are missing in every round. Although the original dataset contains missing data we cannot use the inherent missing data to evaluate the performance of the algorithms. This is because we do not know the correct value of the missing sensor readings; hence it is impossible to determine the accuracy of the algorithms. Therefore we cleaned the data in the first step and implanted the missing values in random for a number of consecutive rounds into the cleaned dataset. Our cleaning process is iterative. Each round consists of sensor readings from all the sensors. If any of the sensors' readings is missing in a round, we removed the entire round. This is necessary because we process the data round by round. But we found that very few rounds can be obtained if we cleaned round by round; therefore in the second step we cleaned sensor by sensor. If a sensor is missing in more than fifty percent of the rounds we removed that sensor. Removing such a sensor will stop us removing the rounds where only that sensor was missing. By repeating the entire process we ended up with nine sensors (sensor ids 41 to 49). We obtained three thousands rounds of data for those nine sensors.

4.1.2 Factory Floor Temperature Data

Besides the above real-life application dataset, we also synthesized a factory floor temperature dataset [16] which exhibits dynamically changing phenomena. In this experiment machines are placed on a grid floor. In the beginning all machines are off and the initial temperature for all grid points is set to zero. The boundary grid point temperature is held at zero throughout the experiment. Some machines will be turned on for a number of rounds; the temperatures on those machines will reach a high constant temperature and heat will disperse on the floor. For each time step, at any non-boundary grid point (i, j), the temperature $T(i, j)$ is updated using the following formula [3]:

$T(i,j) \leftarrow T(i,j) + alpha * [T(i+1,j) + T(i-1,j) - 2 * T(i,j) + beta * [T(i,j+1) + T(i,j-1) - 2 * T(i,j)]$ where alpha and beta are ≤ 0.25 and are the dispersion factors in the *x* and *y* directions, respectively. In this simulation, we simulated the scenario in which we sampled the sensor readings once per hour. In total we gathered 4500 rounds of readings from 24 sensors. It is equal to a six month period, much longer than the duration in which the Intel Berkeley Lab dataset was collected. For this dataset, the machines' on and off status reflects the thermal phenomena changes. Machines were placed at different locations and they were turned on randomly. As a set of machines turned on, the heat transfer started from the turned-on machines to the boundary and the transfer process took place in a different direction. So the relationship among the different locations changed overtime; hence this dataset reflects the phenomena change, a property of many applications in multi-hop sensor networks.

4.2 Performance Comparison Studies

In this section we compare the performances of MASTER-M, SPIRIT [13], and TinyDB [10] in terms of mean absolute error (MAE). MAE is calculated using the following formula: $MAE = \frac{\sum_{i=1}^{n}|e_i - v_i|}{n}$ where e_i is the estimated value and v_i is the original value for the i-th data point). We specifically study the impacts of the number of rounds of sensor readings on the estimation accuracy.

4.2.1 Results for the Intel Berkeley Lab Dataset

The results (Figure 4) show that when the number of rounds of sensor readings is large, i.e. the amount of data used in the estimation process is large; MASTER-M performs much better than the other two algorithms although it is not the best one when the number of rounds is small. MASTER-M shows a very stable performance over time, while the other two methods perform very well at the beginning but deteriorates over time. The data distribution changes and different sensor readings vary differently over time; hence the estimation accuracy for TinyDB and SPIRIT drops. The stable performance of MASTER-M over time implies that MASTER-M is not vulnerable to concept drift. As an approach applied on data streams, the long term trend is more important than the results obtained in the beginning stage, and MASTER-M shows its advantages.

Table 1. Relative average error compared to MASTER-M for the Intel Berkeley dataset

Approach	Average MAE	Error percentage	Relative average error
MASTER-M	1.11	1.71%	Best Approach
TinyDB	2.70	4.17%	58.89%
SPIRIT	2.20	3.39%	49.55%

Figure 5 is a further illustration of how MASTER-M improves the estimation accuracy. Figure 5 demonstrates the relative error for TinyDB and SPIRIT compared to MASTER-M. The relative error for TinyDB is computed as 100 x ($MAE_{TinyDB} - MAE_{MASTER-M}$)/$MAE_{TinyDB}$, where MAE_{TinyDB} and $MAE_{MASTER-M}$ are MAE computed for TinyDB and MASTER-M; similarly we compute the relative error for SPIRIT. At the beginning SPIRIT shows a negative relative error which means at the beginning SPIRIT performs better than MASTER-M; but over time, the error for SPIRIT increases; while MASTER-M shows an almost constant error rate. Table 1 shows the average MAE for all the three approaches, average percentage of error and the relative average error for TinyDB and SPIRIT compared to MASTER-M. According to Table 1 MASTER-M has 58.89% less error than TinyDB and 49.55% less error than SPIRIT.

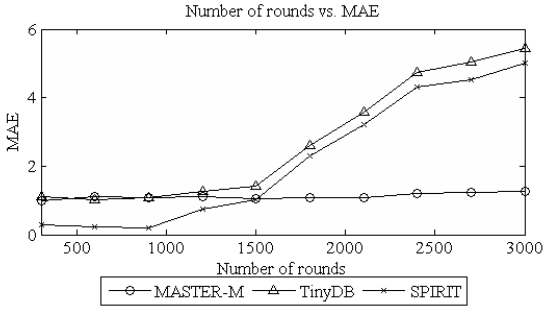

Figure 4. MAE vs. multiple number of rounds for Intel Berkeley Lab dataset

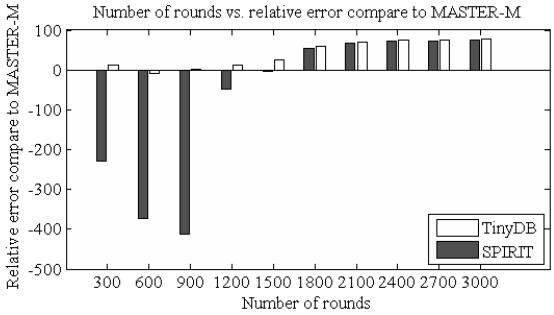

Figure 5. Relative error of TinyDB and SPIRIT compared to MASTER-M for the Intel Berkeley dataset

4.2.2 Results for the Factory Floor Temperature Dataset

Figure 6 shows the MAE with respect to the number of rounds using the synthetic dataset. MASTER-M shows a constant performance over time even though the dataset includes phenomena changes. During the 4,500 rounds time period, the phenomena change many times, and each time MASTER-M correctly puts the related set of sensors into the same cluster; therefore, MASTER-M produces more meaningful association rules and hence better estimation accuracy. TinyDB and SPIRIT show a poor performance because they are not capable of estimating missing sensor readings when the sensor readings change randomly and there exist different relationships among the sensors at different points of time.

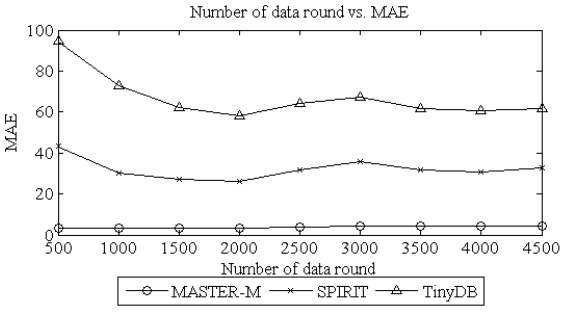

Figure 6. MAE vs. multiple number of rounds for the synthetic dataset

Figure 7 shows the relative error for TinyDB and SPIRIT compared to MASTER-M for the synthetic dataset. MASTER-M performs much better than the other two methods; hence the relative error is very large for both TinyDB and SPIRIT. MASTER-M has approximately 90% less average relative error than TinyDB and SPIRIT. Table 2 shows the average MAE, average percentage of error and average relative error for all three approaches. On average MASTER-M outperforms other two methods significantly.

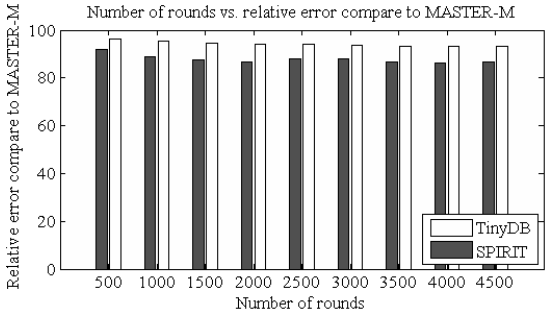

Figure 7. Relative error for TinyDB and SPIRIT compared to MASTER-M for the synthetic dataset

The next section (Section 5) demonstrates the feasibility and supremacy of data estimation using MASTER-M over re-transmission of missing sensor data in terms of energy consumption to emphasize the significance of data estimation for multi-hop sensor networks.

Table 2. Relative average error compared to MASTER-M for the synthetic dataset

Approach	Average MAE	Error percentage	Relative average error
MASTER-M	3.90	0.78%	Best Approach
SPIRIT	32.2	6.44%	87.88%
TinyDB	67.1	13.42%	94.18%

5. ENERGY CONSUMPTION EVALUATION

In this section, we study how much energy savings MASTER-M would produce in comparison to simple re-transmission of missing readings. Heinzelman [6] proposed a power calculation equation (PCE) where the amount of energy used in transmitting a sensor reading is directly proportional to the number of bits and the distance over which they are transmitted. It considers a network of n sensors arranged linearly and gives the power consumed by the network in transmitting k-bit data from the nth sensor to the base station. It also incorporates the energy used by the intermediate hops (sensors between the data originating sensor and the base station) in receiving and forwarding the data to the base station. To have more accurate results, we use the real distances among the sensors instead of the average distances used in [6] to calculate energy consumption. Our modified energy calculation formula, given by Equation (1), calculates the energy consumed (E_n) using the actual distance (r_i) between the sensors. $E_n = n(E_{transmit} \times k) + E_{amplifier} \times (r_1^2 + \ldots + r_n^2) \times k + (n-1) \times (E_{receive} \times k) \ldots \ldots \ldots$ (1) where n is the number of hops, including the sensor where the data originates, through which the data pass before reaching the base station,; k is the number of bits transmitted; r_i is the distance of the ith hop; $E_{transmit}$ and $E_{receive}$ are the amount of energy consumed in running each transmit and receive circuitry, respectively; and $E_{amplifier}$ is the energy dissipated in amplification circuitry for achieving acceptable transmission capability [6]. The values used for each of the above mentioned parameters are given in Table 3 [6].

Table 3. List of constants for the energy equation

$E_{transmit}$	50 nJ/bit
$E_{receive}$	50 nJ/bit
$E_{amplifier}$	100 pJ/bit/m^2
k	2000 bit

We divided our network into sub-networks based on the routing path. The sub-networks consist of linearly arranged sensors so that each sub-network can be considered a linear network as given by Heinzelman. The PCE is now readily applicable to each of the sub-networks. Using Equation (1), we calculate the energy consumed in transmitting a k bit data originating from each of the n sensors in an individual sub-network. Then, the total energy consumed by that individual sub-network with n sensors, in one data round of transmission, is given by Equation (2). Finally, the summation of the total energy consumption by each sub-network gives us the total transmission energy cost of a multi-hop sensor network.

$$\text{Total } E_{power} = \sum_{i=1}^{n} E_i \ \ldots \ldots \ldots \ (2)$$

However, using a simple re-transmission of the missing sensor data instead of MASTER-M, all the sensors, after one transmission, are in the receiving mode for a possible re-transmission request from the base station. For our real-life dataset [7], as each data round is for 31secs, this implies that the possible re-transmission requests take place during this time. Hence in such a scenario, all the sensors are using energy in staying 'awake'. Here, we assume that the re-transmission requests involve a single re-transmission of the missing data. Then, the total energy (E_{nM}) consumed in this case is given by Equation (3) where t is the duration for which a sensor must be in 'awake' mode for possible re-transmission requests. $E_{nM} = n(E_{transmit} \times k) + E_{amplifier} \times (r_1^2 + \ldots + r_n^2) \times k + (n-1) \times (E_{receive} \times k) + t \times (E_{receive} \times k) \ldots \ldots \ldots$ (3) Thus, Equation (3) gives us the total energy consumed in transmitting k bit data originating from each of the n sensors in an individual sub-network using a simple re-transmission process. Next, the total energy consumption by each of the sub-networks and the entire network as a whole is calculated using Equation (2).

The difference in total energy consumption in transmission when using the data estimation algorithms like MASTER-M (Equation 1) and when using a simple re-transmission (Equation 3) gives us the amount of energy saved using MASTER-M. From our experiments, the energy savings amount to 20% which is significant considering that we fixed the missing data rate at 20% and limiting to single re-transmission of the missing data. Figure 8 shows the percentage of energy saved for various missing data rates at 1- 20%. Thus, greater the percentage of missing data in a network, greater the energy consumed by the network in re-transmissions, and greater the energy savings produced by MASTER-M. This justifies our stated argument for developing data estimation techniques like MASTER-M rather than using simple re-transmissions.

In summary, our evaluation of energy consumption shows that using MASTER-M saves energy by avoiding re-transmission. There is a linear correlation between the percentage of missing data and the percentage of energy savings by MASTER-M (Figure 8). MASTER-M saves more energy with increasing percentages of missing data. In our energy calculation we did not consider the subsequent missing sensor readings after a single re-transmission which will require even more energy than the one we show for a single re-transmission. In that case the actual energy savings by MASTER-M in real life scenarios is even greater than what we showed in Figure 8.

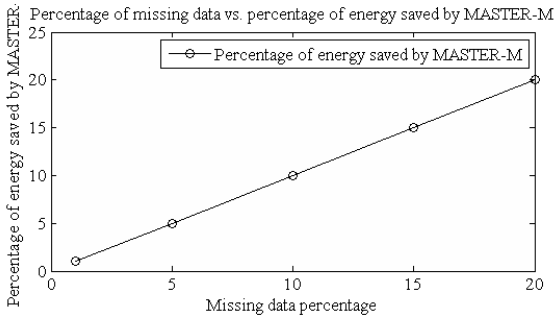

Figure 8: Energy Saved using MASTER-M compared to single re-transmission for variable missing data rate per round from 1-20%

6. CONCLUSION AND ONGOING WORK

In this paper, we have presented an algorithm that makes use of data clustering and association rule mining to estimate the values of missing sensor data in multi-hop sensor networks. We propose a dynamic clustering algorithm based on the distance between two sensors. Our novel distance function definition addresses the simultaneous missing problem and phenomenon change in the environment. The novel distance represents the relationships of the sensor pairs in multi-hop networks. We performed extensive experiments on both real-life and synthetic datasets, which show that our algorithm provides better estimation accuracy compared with existing algorithms. The experiments also show that our algorithm is able to save energy. We are currently considering how to extend MASTER-M to accommodate mobile sensor networks as mobility introduces new challenges which complicate data estimation.

7. ACKNOWLEDGMENTS

This work has been supported in part by the NASA under the grants No. NNG05GA30G.

8. REFERENCES

[1] D. Arthur , S. Vassilvitskii: k-means++: The Advantages of Careful Seeding, ACM-SIAM symposium on Discrete algorithms, Jan 2007.

[2] H. Chok: Spatio-Temporal Association Rule Mining Framework for Estimating Missing Data in Sensor Networks and Analyzing Trend Evolution of Co-evolving Multidimensional Data Streams, Master's thesis, University of Oklahoma, May 2009.

[3] C. Conner: Modeling Heat Transfer in Parallel, http://www.cas.usf.edu/~cconnor/parallel/2dheat/ 2dheat. html.

[4] C. Giannella, J. Han, J. Pei, X. Yan, and P. Yu.in H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.): Mining Frequent Patterns in Data Streams at Multiple Time Granularities. Next Generation Data Mining, AAAI/MIT, 2003.

[5] L. Gruenwald, H. Chok, and M. Aboukhamis: Using Data Mining to Estimate Missing Sensor Data, IEEE International Conference on Data Mining Workshops Oct. 2007

[6] W. R. Heinzelman, A. Chandrakasan,H. Balakrishnan: Energy-Efficient Communication Protocol for Wireless Microsensor Networks, 33rd Hawaii International Conference on System Sciences, 2000

[7] Intel Berkeley Research Lab. http://db.csail.mit.edu/labdata/labdata.html, accessed 2009

[8] N. Jiang and L. Gruenwald: Research Issues in Data Stream Association Rule Mining. ACM SIGMOD RECORD, 2006.

[9] S. Kay: Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall, 1993.

[10] S. Madden, M. Franklin, J. Hellerstein and W. Hong: TinyDB: An Acquisitional Query Processing System for Sensor Networks, Transactions on Database Systems 2005.

[11] G. McLachlan and T. Krishnan: The EM Algorithm and Extensions, Wiley series in probability and statistics, 1997

[12] Metar, http://metar.noaa.gov/, Jan 2010

[13] S. Papadimitriou, J. Sun, and C. Faloutsos: Streaming Pattern Discovery in Multiple Time-Series, 31st VLDB, 2005.

[14] J. Shafer: Model-Based Imputations of Census Short-Form Items, The Annual Research Conference, 1995.

[15] A. Silberstein , R. Braynard , J. Yang: Constraint chaining: On Energy-efficient Continuous Monitoring in Sensor Networks, ACM SIGMOD ICMD, June 2006

[16] A. Silberstein, K. Munagala, and J. Yang: Energy-Efficient Monitoring of Extreme Values in Sensor Networks, ACM SIGMOD ICMD, June 2006.

[17] N. Vijayakumar and B. Plale: Missing Event Prediction in Sensor Data Streams Using Kalman Filters, book chapter in Knowledge Discovery from Sensor Data, Taylor and Francis/CRC Press, 2009

[18] A. Mainwaring , D. Culler , J. Polastre , R. Szewczyk , J. Anderson: Wireless Sensor Networks for Habitat Monitoring, 1st ACM international workshop on Wireless sensor networks and applications, Sept 2002.

[19] L. Schwiebert, S. Gupta, J. Weinmann: Research Challenges in Wireless Networks of Biomedical Sensors, MOBICOMZOOJ, 2000

[20] J. Ibriq and I. Mahgoub: Cluster-based Routing in Wireless Sensor Networks: Issues and Challenges, SPECTS, July 2004.