# Online Cleaning of Wireless Sensor Data Resulting in Improved Context Extraction

Sangeeta Mittal
Department of CS & IT
A-10 Sector-62
Noida, UP, India

Alok Aggarwal
Department of CS & IT
A-10 Sector-62
Noida, UP, India

S.L. Maskara
G-2W, Soura Niloy Housing
1-Kailash Ghosh Road
Kolkata, WB, India

## ABSTRACT
Wireless Sensors enable fine grain monitoring of activities of individual and social interest. Typically these sensors sense & send data continuously directly or through other sensor nodes to a base station. Wireless Sensor Data are inherently noisy and have frequent random spikes due to dynamic nature of the medium. Hence, the decision at the receiving node based on such data is likely to be erroneous. Erroneous data and decisions may affect its transformation to meaningful form like 'context'. It is therefore desirable to clean the data for improved context extraction. Bayesian Belief Networks are used here to quantitatively encode the dependencies among various sensors. These dependencies are then used to estimate missing data and also to detect and recover from errors. Cleaned data is then used for deriving Contextual Information and it results in improved context feature calculation. In this paper five algorithms for Bayesian Belief Network Construction have been evaluated and their performance of classification studied. Conjunctive rules are defined to map the sensors to already defined context. A secondary data obtained from weather sensor boards installed at Intel research lab at Berkeley have been used to demonstrate the approach.

## General Terms
Smart Sensor Systems, Ambient Intelligence, Sensor Information Processing

## Keywords
Wireless Sensor Networks, Bayesian Belief Networks, Sensor Data Recovery & Classification, Context Extraction.

## 1. INTRODUCTION
Wirelessly connected sensors are used for continuously emitting information about the environment in which they are placed. WSNs are enabling interesting monitoring applications like health care, security, habitat and environment monitoring. In a typical Wireless Sensor Network, tiny sensing devices, measure some physical parameters of their surrounding like body temperature, blood pressure, heart beat, ambient temperature, humidity wind speed and seismic activity using Richter scale and many more[1]. The sensed data is then transmitted to defined destination and presented to the end user. Over the past decade, sensor technologies have advanced in sensor hardware, routing mechanisms and data interpretation to gather this information from even remote unreachable places. Due to inherent uncertainties of wireless medium and low resources of sensors, errors are though frequently introduced by the time data is received at the receiver. Several mechanisms have been developed for detecting and minimizing errors [1]. Extraction of useful information from this ever increasing pile of raw data and that too in real time, is critical for taking remedial actions or enhancing the understanding of environment or undergoing activity [1]. Sensors are capable of sensing, calibrating and then transmitting the sensed values. This builds a heap of data at the receiving station very early; hence at the receiver methods are applied to extract the categorical knowledge from the physical parameters in real time. This categorical knowledge can be defined as pertaining to context of the place being monitored. Examples of such derived information can be location, activity, proximity, physical conditions etc. This abstract representation of physical parameters has to be derived from error free raw sensor data [2]. Suitable machine learning methods have to be applied for extraction of contextual knowledge from sensor data. In literature several machine based tools like Neural Networks, Decision Trees and Hidden Markov Models have been used for same [3]. Here Bayesian Belief Networks (BBNs) that are stochastic models that describe and quantify probabilistically the relationship between one or more set of data variables are used for contextual information extraction. The reason for choosing BBNs particularly is due to the fact that the sensors are densely deployed to capture the underlying phenomenon closely. The proximity among sensors results in high probability of correlations among some of them. In most of the situations these correlations, information and the associated knowledge are random in nature and as such require the use of probability and random theories for interpretation. The graphical representation of the independences between the modelled variables allows for ease of interpretation of the model and its parameters. Classification is done using BBNs by establishing posterior probabilities of the various classes for a given instance of the feature variables. A major advantage of classifiers based on BBNs lies in their ability to give reliable classifications even if evidence is available for only a subset of the feature variables [4]. Unlike [4] BBNs are used here as multipurpose tool, one of the purpose is domain knowledge modelling. The model constructed has been used to clean the data by detecting & removing errors from sensor data stream, provide energy efficient sampling and derive contextual information. It has been shown that this has resulted in improved feature extraction as compared to the previous results as described in [4]. A two-step mechanism of removal of errors & extraction of context from raw sensors data has been worked out. The first step is to do an offline modeling of relationships among attributes of dataset by Bayesian Belief Networks using various BBN construction algorithms as discussed in second section. The second step is to use the BBN as constructed for sensor data cleaning that is, detection of outliers and approximation of missing data. This is discussed in third section. In fourth section, the use of this model for deducing features of context i.e. class of temperature, humidity, ambient

light and time of the day from observable sensor data and extracting context using rule based system is discussed. The paper is concluded in the last section.

## 2. BAYESIAN BELIEF NETWORKS FOR MODELING SENSOR DATA

A Bayesian Belief Network is a Directed Acyclic Graph (DAG) consisting of a set of nodes and edges. Each node of the graph represents a random variable and each arc represents a direct probabilistic dependence between two variables. A BBN conveys a joint probability distribution of its variables, which is the product of the local distributions of each node and its parents. The DAG represents the structure of dependencies between nodes and gives the qualitative part of BBN. Quantification consists of prior probability distributions over those variables that have no predecessors in the network and conditional probability distributions over those variables that have predecessors [5].

### 2.1 Construction of Bayesian Belief Networks

Let U = {$x_1$…… $x_k$} for k >= 1 be a set of feature variables that describe a problem domain. A Bayesian Belief Network, B over these set of variables U is a network structure BBNs, which is a Directed Acyclic Graph (DAG) over U and a set of probability tables BBNp , where

$$BBN_p = p(u_i \mid pa(u_i)) \quad \text{for all i= 1 to k} \qquad (1)$$

where pa($u_i$) is the set of parents of $u_i$ in BBNs[5]. A Bayesian Belief Network also represents joint probability distribution on whole set of variables, U as

$$P(U) = \prod_{i=1}^{k} p(u_i \mid pa(u_i)) \qquad (2)$$

The main issue in learning a BBN is determining the dependence probabilities. One of the methods is to approach the experienced persons of a domain and get the probabilities as well as arcs from them. But many times, due to non-availability of sufficient number of experts or even. if available, to assist them , machine learning algorithms making use of graph algorithms and information theory are used to autonomously find both the graph and probabilities of dependence among the various features. The second method is a supervised one and requires labelled training dataset, D of that domain. Even for a small feature set, the exhaustive possible probability distribution calculation with all possible graphs is an enormous task. Therefore most of the algorithms proposed to find BBNs, first find the graph and then the probability distribution. Coarsely, the methods are divided as using either of the two approaches. First of these assume, total dependencies and then after going through data removes all the arcs where dependency is not detected in data. The second type of method initiates with empty arc sets and calculate its information score with respect to data. It keeps on searching for other possible graphs until the one with highest score is found [5]. As the number of possible graphs over a set of variables can be large, heuristic searching methods like hill climbing and greedy search are used instead of linear searching. Due to exhaustive calculation of independence among variables, algorithms in first category face the problem of 'Memory Crash' with graphs of order of more than ten nodes. Due to this problem of conditional independence based algorithms and size of our dataset, representative algorithms from second type of methods are chosen [6]. These are: A Naïve Bayesian Structure (A1) employs a static graph structure of problem domain where class variable is root and all other variables are only dependent on it and independent of each other [7]. Algorithm A2 creates a Maximum Weighted Spanning Tree Structure. Weights are based on either the mutual information between the two variables or the score variation (when one node becomes a parent of the other) and assigned to each edge in an initial random DAG. Given a desired root, the minimum weighted spanning tree can be obtained, using Kruskal's or Prim's methods. Another metric for reducing search space has been devised by Cooper et al in [8] as K2 Algorithm (A3). The method requires an initial DAG to be provided as node topological order. The initial DAG may be a totally random graph or an informed one like a MWST DAG, reversed MWST and a randomly ordered DAG. With respect to each of these orders, the search space is explored to maximize probability of structure given data i.e. maximize the *'Bayesian Score'* of DAGs as calculated in equation 3 [9]. A detailed description of these algorithms can be found in [10].

$$\max_{B_s}[P(B_s, D)] = \prod_{i=1}^{n}[P(\pi_i \rightarrow x_i)\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}N_{ij}\prod_{k=1}^{r_i}N_{ijk}! \quad (3)$$

Heuristic based searching algorithms like Greedy Search (A4) and Hill Climbing Search (A5) were used to study the effect of search methods on computation time of classifier and its performance. Bayesian Score as defined in (3) was used in greedy search to choose the next DAG with maximum score and in hill climbing to jump away from local maxima. All the algorithms are implemented using BNT Structure learning package available at [11].

### 2.2 Experiments – Datasets and Models

For validating our methods, a real wireless sensor board data set is used. The data has been collected from 54 Mica2Dot weather sensor boards deployed in planned but non uniform manner in the Intel Berkeley Research lab between February 28th and April 5th, 2004. The area of the lab is approximately 41* 31 sq. m. At [12], time stamped data from all the boards with schema as shown in figure 1 is provided.

| Date | Time | Epoch | Moteid | Temperature | Humidity | Light | Voltage |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

**Fig 1: Schema of the Sensor Database**

Tuples with node id, current humidity, temperature, light and voltage measurements reach once every 31 seconds to the gateway node. The observation data of 28 continuous days is available in the dataset which is large enough for training as well as testing.

*Data Preprocessing*

The data received in the database at the receiving server is continuous. All the physical parameters are real values depicting the measurements in Celsius for temperature, percentage for relative humidity, lux for light and volts for voltage. The sensors data was highly noisy and had lot of missing and erroneous values. For structure learning purpose tuples with missing were discarded at the time of training.

Records with erroneous values were identified based upon non - feasibility of physical parameters like relative humidity can never be negative, temperature can't reach 122 degree Celsius.

*Quantization and Clustering*

Though the lab is an indoor, probably controlled, environment but still in a typical day temp variations of up to 9 degrees at same point of time is observed. The reasons for this kind of variation could be real world conditions like incidence of sunlight at some boards while location of air vents near others. At night time, areas near windows get colder, while at sun rise the east side of the lab report increased temperature due to sun. Uniform temperatures are sensed by all boards towards end of the day. Thus it is clear that the climatic data has spatial and temporal properties even within indoors. These correlations are modelled by belief networks where spatiality is modelled in clusters and temporality in the time stamped data itself. Given small area of the lab and physical proximity of some boards with each other, the motes are clustered into thirteen groups, named from G1 to G13. Figure 2 shows the arrangement of sensors in the lab and our clustering scheme in bold rectangles.

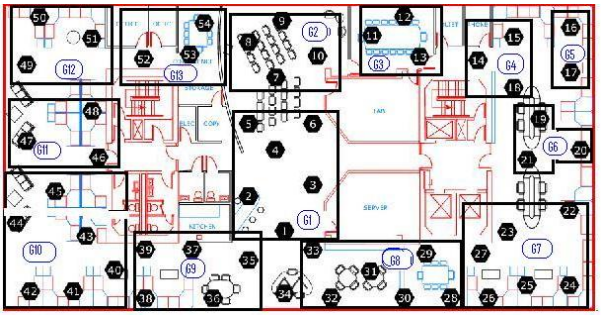


**Fig 2. Clustering of Sensors placed in a 41*31 sq.m Indoor Area**

The temporal dimension of the data set is prepared first by converting dates into day numbers starting from 28th February as day-1. Then parameter wise average of a set of 120 tuples each is taken to obtain 24 hour wise instances of data in one day. All the physical parameters being measured are inherently continuous. These are quantized to discrete categories. There are two reasons for doing this. One of the reasons is Bayesian Belief Networks are known to work better on discretized values. The other reason is that categorized symbolic values as shown in table 1 are more understandable instead of actual real values [13]. Taking inputs from various weather experts and information on Internet, the categorization of temperature is done in 5 classes, humidity in 4 classes and light in 7 classes. The detail of range of values within each class is given in table 1. The quantized values of the features replaced the actual values in the data sets accordingly.

Both the steps discussed above are domain specific and domain specific knowledge and/or inputs from experts are required to undertake these. For various instances of similar applications, the preprocessing is required to be changed only when quantification criteria are modified or new features are added [14].

## 2.3 Domain Modeling Using Bayesian Belief Network

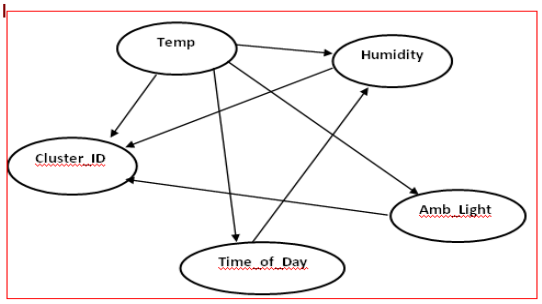
The BBN construction algorithms discussed in section 2.1 model relationships between feature variables Cluster Ids and Time of the day, the virtual sensors and temp, humidity and ambient light of the described data set. The final sensor data obtained per cluster per hour over a period of 25 days is used for offline learning BBN models. Models were extracted using all the algorithms described in 2.1; some of these are also shown in Figure 3.

**Table 1. Quantization of WSN Data for BBN Modeling**

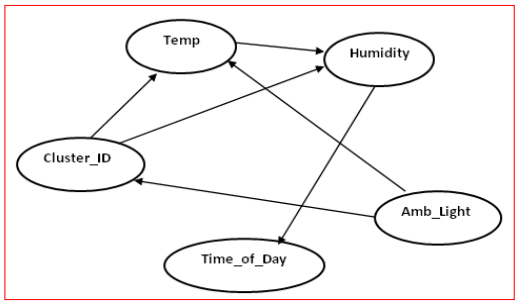
| Feature of Context | Class No. | Range | Symbolic Name |
|---|---|---|---|
| Temperature (Range in Degree Celsius) | 1 | <10 | Very Cold |
| | 2 | 10 – 18 | Cold |
| | 3 | 19-25 | Normal |
| | 4 | 26-35 | Mild |
| | 5 | >35 | Hot |
| Humidity ( Range in % of Relative Humidity) | 1 | <=20 | Dry |
| | 2 | 21-28 | Comfortably Humid |
| | 3 | 29-45 | Quite Humid |
| | 4 | >45 | Highly Humid |
| Ambient Light ( in Lux) | 1 | <=10 | Pitch Dark |
| | 2 | 11-50 | Very Dark |
| | 3 | 51-200 | Dark Indoors |
| | 4 | 201-400 | Dim Indoors |
| | 5 | 401-1000 | Normal Indoors |
| | 6 | >1000 | Bright Indoors |

The algorithms as discussed work on different principles, to work out the output structure in a non-deterministic manner. This explains the differences in dependency graphs obtained by various algorithms. Still, some obvious dependencies like that of temperature with time of the day e.g. cold at night time and location i.e. clusters e.g. clusters near windows show higher temperature during day is correctly represented in all three models. For evaluation of all structures in classifying

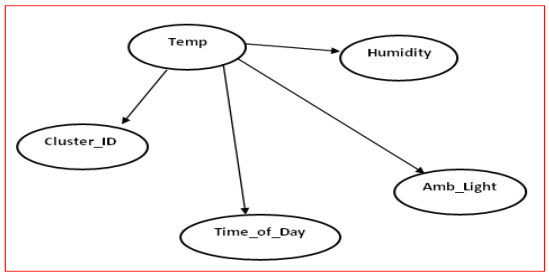
unseen data, the conditional probability tables for each of the variables were calculated using maximum likelihood estimation (using frequency counts). This provided the complete quantitative models of all the features. In next section use of BBN as a tool for online cleaning of data transmitted by sensors is shown.




(a)




(b)




(c)

**Fig 3. Bayesian Belief Network obtained using (a) K2+MWST (b) GS (c) Naïve Bayesian Network**

## 3. ONLINE DATA CLEANING FOR CONTEXT EXTRACTION

A layered architecture was defined by us in [4] for context extraction at base station from raw sensor data transmitted to it. A modified mechanism with an additional step of BBN based data cleaning & optimization is discussed here for recovering correct data in presence of errors and facilitating energy efficient sampling.

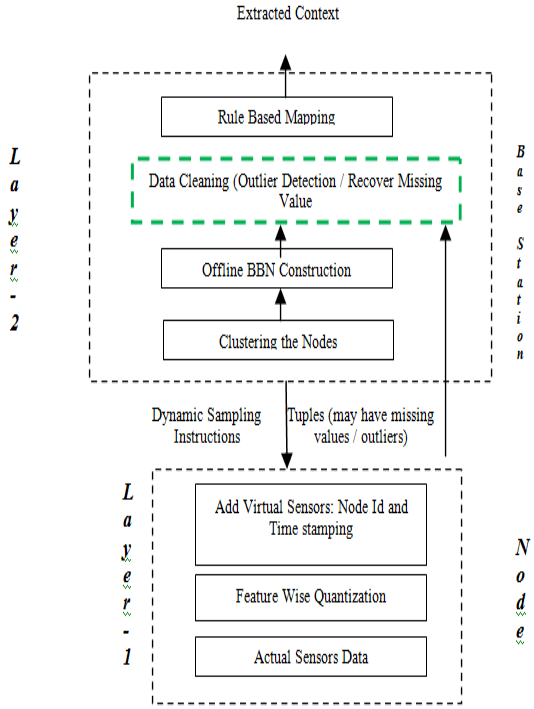



**Fig 4. Mechanism of Context Extraction from Cleaned Raw Sensor Data**

This results in refined input data for better context extraction. A hierarchical WSN, where the sensing nodes are at one layer and the collector node which may be a more resourceful computer acting as base station for all nodes is another layer forms the part of context extraction architecture. The processing is distributed at these two layers as given in Figure 4.

### 3.1 Processing within Nodes

*Processing at Sensor Node*

The first layer, which is present in individual sensor nodes does threshold based quantization of raw sensor values. This is required as the data is to be utilized Bayesian Belief Networks which may not perform well on continuous real valued data. This layer after adding location as Node Id/Cluster Id & time information transmits the quantized tuples which are used as primary context feature. The tuples may introduce error either at the source due to faulty or compromised sensor or during transmission.

*Processing at the Base Station*

This data is utilized by the second layer at the base station. The classification component based on BBNs stores probabilistic relations of various features quantitatively in compact form. Dynamic sampling decisions are taken here according to the need of application in current scenario. For example, sampling one hourly temperature data performs satisfactorily in indoor office environments and significant reductions can be obtained in number of transmissions. While in a freezer with perishable goods, transmission has to be more frequent. Sampling can also be adjusted as to selectively sample only few types of sensors while maintaining the quality of query answer. It is explained in more detail in next section. In case a value is missing or an outlier is introduced during transmission BBN based inference detects & recovers that value. In this layer the semantics of context are defined and an automatic procedure to classify context features into

abstract context situations is laid [15]. Features obtained in such way provide a reliable and energy efficient context after rules are applied on them. The underlying WSN can be large such that nodes don't always directly transmit to base station. In such cases we make use of existing routing and topology maintenance mechanisms provided in several research works and sensor applications. In next subsections, evaluations of the use & performance of BBNs in handling data errors in form of outliers and missing values is done. The values assigned by the classifiers are compared against the actual classes. Accuracy is defined as the percentage of instances that were labelled with correct class values. On the other hand the percentage of misclassified instances is termed as Error rate. The BBN based classifier labels instances with class value having maximum a posterior probability. Given evidence attributes, in some instances more than one class have maximum a posterior probability, such instances are not classified by the classifier here and are counted as percentage of 'Rejection'. Any test instance contributes to exactly one of these percentages. Any classifier having higher Accuracy is better. Algorithms were also compared on Error Rate vs Rejection Rate. An algorithm that rejects instances instead of misclassifying them is preferable to reduce false alarms in case of actuation [16].

## 3.2 Outliers & Detection of Faulty Sensors

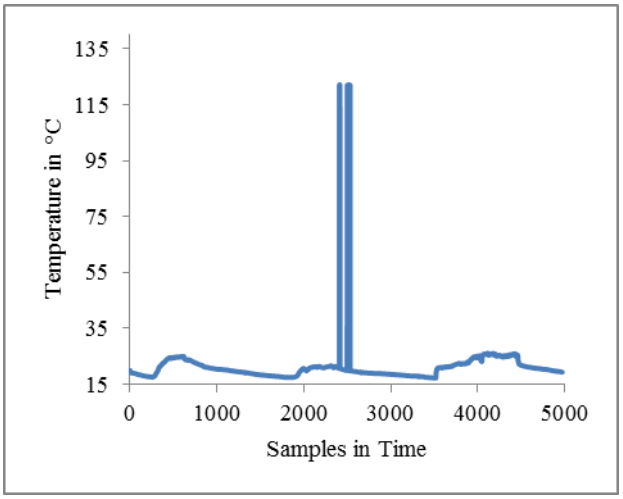
Owing to various factors such as the surroundings, the quality of the sensors, sensors running out of power, compromised sensors etc outliers are introduced in data. Right now, we are assuming that all outliers are false and introduced due to errors. Figure 5(a) & (b) showing the values obtained from 'Temperature' Sensors & Humidity Sensors over a period of 2 days show unexpected or infeasible values randomly scattered in the data. For Outlier Detection BBN is used as a classifier. The classification task in general consists of classifying a variable, C called the class variable given a set of variables A = $a_1$.....$a_n$, called attribute variables. A classifier Z: A → C is a function that maps an instance of A to a value of C. Given dataset D over U, BBNs have been constructed in previous section. To use a Bayesian network as a classifier, $\text{argmax}_y P(C|A)$ using the distribution P(U) represented by the BBN is calculated. That is,

$$P(C \mid A) = \frac{P(U)}{P(A)} \qquad (4)$$

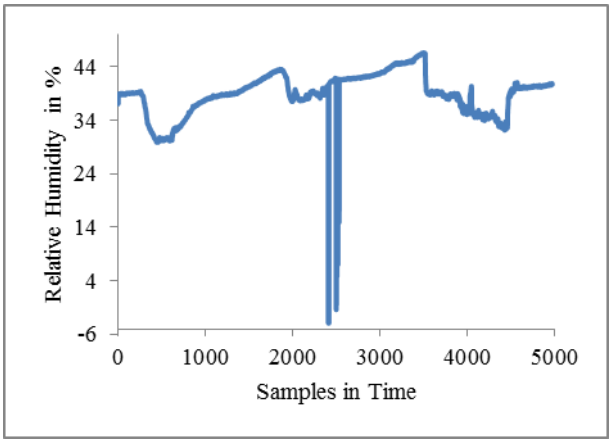
Maximizing LHS, $\alpha$ maximizing $P(U)$

$$= \prod_{i=1}^{n} p(a_i \mid pa(a_i)) \qquad (5)$$

If all variables in A are known, eq. (5) can be used to find posterior probability distribution of class values in C. This is known as Probabilistic Inference and is used to estimate probability of a set of query nodes, given values for some evidence. This is also called belief propagation in BBNs.



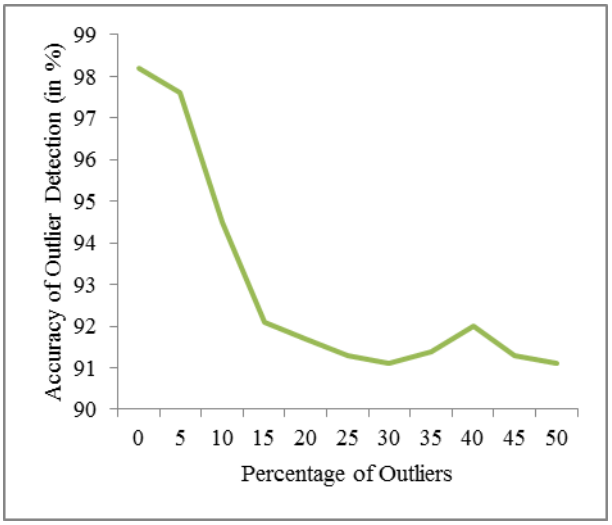

(a)



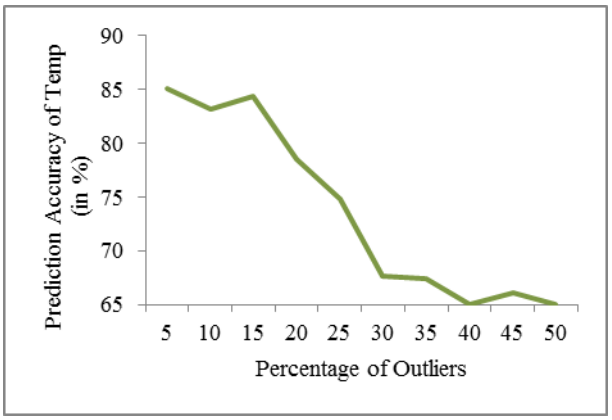

(b)

Fig 5 (a) Temperature Data of 2 days (b) Relative Humidity Data of 2 Days

Using belief propagation the base station compares the probability of its most likely data, with the probability of its actual sensed reading. If the two differ largely then the data is designated as an outlier [17]. To simulate a realistic behaviour, random outliers are added to each frame of the generated data. The percentage of outliers is varied in each frame of the learning data to test the robustness of our classifier. Figure 6(a) shows the effect of performance of outlier detection on different percentages of outliers in Test data. Figure 6(b) shows the error percentage in prediction using test data after the learning phase. As the number of outliers increases the accuracy decreases and then remains constant at 66%. BBNs obtained above provide a tool for real-time detection of outliers. BBN provide information about the




(a)




(b)

**Fig 6 a) Effect of Outliers in Training Data on Outlier Detection using K2+MWST (b) Effect of Outliers on the prediction accuracy of Temperature Feature using K2+ MWST Algorithm**

distribution of the actual data at sensors and hence predicts the temperature values of any sensor at any time. It then compares this prediction with its sensed value. If the two differ significantly, and if the sensed value is not probable, then its' decided that reading is indeed an outlier. After this the base station may replace the outlier with probable value, ask for new value or stop the node entirely from sending further data if it repeatedly sends outliers.

## 3.3 Approximation of Missing Values

In sensor data sampled at a specific time instance value of one type of sensor may be missing in some instances. The sample of original lab data obtained at the base station is shown in figure 7. The highlighted data tuples have missing values of Ambient Light. Light has important role to play to derive context like presence of persons. It is not always true that people are present only in the daytime. There are data instances that represent working in labs till late in the night. Automation related to weather control hence critically requires light information all the time. Such errors frequently occur in sensor networks due to node failures and lost packets. Congestion, Collisions, fading and other interferences due to moving objects or bad weather are major reasons of partial or completely missed data [18]. Missing values impact the monitoring & tracking applications. Generally missing values

```
2004-03-21 03:07:09.948539 63617 54 19.3808 46.4593 0.46 2.34751
2004-03-21 03:08:24.311233 63620 54 19.3808 46.4265 0.46 2.35683
2004-03-21 03:16:59.913939 63637 54 19.371 46.4265 0.46 2.35683
2004-03-21 03:18:01.633626 63639 54 19.3612 46.4265    2.35683
2004-03-21 03:19:36.329432 63642 54 19.3318 46.4921    2.35683
2004-03-21 03:21:54.478856 63647 54 19.3318 46.4921    2.35683
2004-03-21 03:26:00.659967 63655 54 19.2926 46.5577    2.35683
2004-03-21 03:26:23.73739 63656 54 19.3024 46.5577     2.35683
2004-03-21 03:27:10.199589 63657 54 19.2926 46.5577    2.35683
2004-03-21 03:27:54.532845 63659 54 19.2926 46.5577    2.35683
2004-03-21 03:28:26.424869 63660 54 19.3024 46.5577    2.35683
2004-03-21 03:29:30.797571 63662 54 19.2926 46.6232    2.35683
2004-03-21 03:32:39.232764 63668 54 19.224 46.8197     2.34751
2004-03-21 03:35:31.229537 63674 54 19.1946 46.8851    2.35683
2004-03-21 03:36:23.668814 63676 54 19.1946 46.8851 0.46 2.35683
2004-03-21 03:40:53.577677 63685 54 19.2338 46.7542    2.35683
2004-03-21 03:41:58.37438 63687 54 19.2436 46.7215     2.34751
2004-03-21 03:44:30.255195 63692 54 19.2044 46.7869    2.35683
2004-03-21 03:48:25.048064 63700 54 19.1946 46.8197    2.34751
2004-03-21 03:50:54.461202 63705 54 19.1652 46.8851    2.35683
2004-03-21 03:51:47.410009 63706 54 19.1652 46.8851    2.35683
2004-03-21 03:51:54.524274 63707 54 19.1848 46.8851    2.35683
2004-03-21 03:54:09.062771 63711 54 19.1652 46.9178    2.34751
2004-03-21 03:54:58.72854 63713 54 19.1652 46.9178     2.35683
2004-03-21 03:56:54.048886 63717 54 19.126 47.0486 0.46 2.35683
2004-03-21 03:58:07.048782 63719 54 19.126 47.0813     2.34751
2004-03-21 03:58:32.283928 63720 54 19.1064 47.114 0.46 2.34751
2004-03-21 03:58:53.578161 63721 54 19.1064 47.114     2.35683
2004-03-21 04:02:02.176014 63727 54 19.1358 47.0159    2.35683
2004-03-21 04:04:22.494538 63731 54 19.126 47.0159     2.35683
2004-03-21 04:05:35.740395 63734 54 19.1064 47.114 0.46 2.35683
2004-03-21 04:07:14.334692 63737 54 19.0672 47.2773 0.46 2.35683
2004-03-21 04:07:24.257353 63738 54 19.0672 47.3426 0.46 2.34751
2004-03-21 04:10:03.182624 63743 54 18.979 47.6687 0.46 2.34751
2004-03-21 04:11:32.943437 63746 54 18.93 47.8966 0.46 2.34751
2004-03-21 04:11:53.481454 63747 54 18.9104 47.9942 0.46 2.34751
```

**Fig 7. Excerpts of Actual Sensors Data Received at Base Station ( Highlighted Tuples have Msiing Values)**

are handled by error correction codes and extra protection bits which are not known to very effective when missing values are frequent. Information obtained as BBN can be used to recover missing values. In the learning phase the data with missing values were discarded as there was enough complete data to learn the belief network structure and parameters. Though for online extraction of context from incoming sensor data, any missing value can't be simple discarded as it will affect the quality of context deduced. Here the offline modeled BBN is used as a classifier to predict the missed value. This is performed by inferring its class using inference algorithm of message passing. For example in above instances, given values of all other sensor variables, ambient light can be deduced with high confidence. BBNs can also predict missing values even if values from more than one sensor are missing.

# 4. Energy Efficient Context Extraction Using Cleaned Data

Prediction of values from the Bayesian Belief Network structure and parameters is an inference problem. BBN based classifier when applied on available incoming stream provides the values of missing data and also predict with probabilistic confidence whether it is erroneous or not. Here use of BBN has been described to enable energy efficient data sampling and context information extraction. Further, the energy efficiency of data collection can be improved owing to knowledge encoded in the BBN maintained at the base station typically for stationary phenomenon like indoor weather, where BBN parameters don't change frequently.

## 4.1 Dynamic Sensor Sampling

The readings of different sensors are correlated with each other across time and space at times. Bayesian Belief Networks quantify this correlation [19]. If one sensor is dependent on another, using BBN the value of either can be probabilistically calculated given the other. The accuracy of this calculation depends upon the amount of correlation and correctness of received data. For evaluating the effect of

dynamic sampling on accuracy, 3 days of sensor data was used for testing. Evaluation of all algorithms in deducing values of each sensor types for test data set is shown in figure 8. The accuracy rate of K2 based algorithms is overall better. It is noted that humidity's prediction given rest other information yields maximum accuracy in all algorithms. It shows that, humidity need not be sent every time in sensor transmission. This dynamic sampling provides significant energy savings while not compromising information.

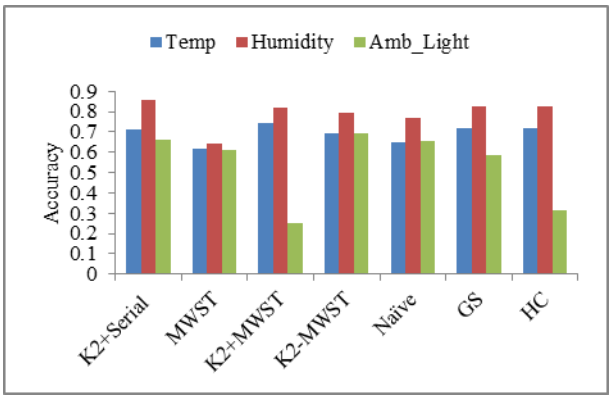



**Fig. 8 Accuracy of Various Learning Algorithms in Classifying Context Features**

The accuracy rate of any algorithm is not near perfection i.e. close to 100 %. It was found by further analysis that all algorithms performance improved if the time window size is increased to 2 hrs. This however decreases the quality of monitoring to a coarser level which may be undesirable in sensitive monitoring like store of perishable goods or livestock.

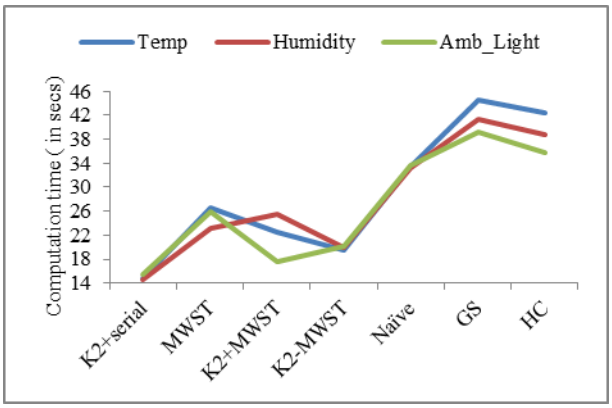



**Fig. 9 Computational Time Taken by Different Algorithms on a 2.4Ghz Clock Speed Machine**

Another important factor for successful dynamic sampling is the amount of time taken to calculate the pseudo values of the sampled parameters is Computational Time (CT). The time taken should be negligible, such that the sample looks to be real. All the algorithms were run on same data using a core 2 duo processor of 2.4GHz clock speed with 2 GB RAM. As shown in figure 9 there are slight variations in computation time of each algorithm. Greedy Search and Hill Climbing take more time to classify due to their search space being larger than those of other algorithms used here.

## 4.2 Extracting Features of Context from Cleaned Data

Information of interest from sensors like activity, location and surroundings (nearby persons, devices) are called 'contexts'. As compared to our previous work in [4], here the same test data is used, but here it is first cleaned on arrival at the base station (outlier detection & missing value replacement) and then used to classify the feature of interest. Previously, BBN was directly applied to test data for feature classification. Like earlier, 5-Fold 10 Times Cross-Validation has been used to generate training & testing datasets from available data. Given the flexibility of our model any of the feature variables whose value needs to be determined can act as class variable. Multiple categories of classes in each feature have been described in table 1 already. Confusion matrix is used to assess the ability of the learnt model in distinguishing all class values appropriately with high accuracy. In tables 2 & 3 confusion matrix of classification of humidity and ambient light are shown. For every class two columns of values are given, first column named 'P' i.e. Previous contains the classified instances as reported in [4]. The column labelled 'I' ,that is, Improved is the number of classified instances using approach described here. Similar matrices for other features are not shown here due to shortage of space. It is interesting to note that mostly the classifier gets confused between adjacent classes this is due to the fact that humidity doesn't change abruptly after every one hour. The intra cluster variations are also not captured.

**TABLE 2. Confusion matrix for cross validation testing of humidity classification with K2+MWST**

| Actual Class Values | Predicted Class Values | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | |
| | P | I | P | I | P | I | P | I |
| 1 | 2 | 2 | 14 | 10 | 0 | 0 | 0 | 0 |
| 2 | 4 | 4 | 128 | 131 | 468 | 434 | 44 | 44 |
| 3 | 0 | 0 | 356 | 406 | 263 | 258 | 265 | 265 |
| 4 | 0 | 0 | 60 | 60 | 418 | 376 | 166 | 171 |

**TABLE 3. Confusion matrix for cross validation testing of Ambient Light Classification with K2+MWST**

| Actual Class | Predicted Class Value | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | |
| | P | I | P | I | P | I | P | I | P | I | P | I |
| 1 | 107 | 129 | 64 | 49 | 15 | 14 | 27 | 24 | 20 | 14 | 4 | 4 |
| 2 | 57 | 45 | 1688 | 1742 | 225 | 204 | 141 | 129 | 40 | 32 | 6 | 5 |
| 3 | 3 | 3 | 442 | 388 | 1324 | 1451 | 442 | 383 | 20 | 10 | 11 | 7 |
| 4 | 27 | 27 | 235 | 160 | 348 | 287 | 1706 | 1850 | 65 | 60 | 26 | 23 |
| 5 | 18 | 11 | 51 | 40 | 12 | 10 | 107 | 76 | 432 | 498 | 54 | 39 |
| 6 | 0 | 0 | 0 | 0 | 32 | 32 | 55 | 55 | 67 | 62 | 398 | 403 |

The confusion matrix in table 3 shows a high misclassification rate not only to adjacent classes but further also. The effect is easily explainable as this may be due to the fact that though light exhibits regular pattern during 24hrs window but is easily hindered by the presence of some object near the sensor, thus giving abrupt values. Besides above matrices, the accuracy was also tested using only the time of the day and cluster id as available data. The model inferred 55-60% correct results for temperature as well as humidity in this case. Very good accuracy with confusions mainly in adjacent classes was obtained while classifying temperature given other features. Similar evaluations were done in [4], but results of confusion matrix are better here in terms of number of misclassifications among classes. Specifically the spilling to adjacent classes has reduced in most of the cases. The confusion matrices of only two features with only two algorithms have been shown here due to limitations of space. Matrices of other combinations were created and similar results were obtained. The correctly classified features would prove more reliable for identifying associated context. It would act as a breakthrough in sensor based actuator networks. The mechanism to do so is discussed in next section.

## 4.3 Rule Based Context Extraction

After obtaining features of context with sufficient confidence, the context "weather" can be extracted from the data using simple rule based substitution. The type of weather context that is of interest to us is pleasant, comfortable, suitable for work, not suitable for work and uncomfortable. The qualitative description of context in table 4 specifies mapping of features of weather to type of weather. . The rules are defined based upon personal opinion and description is qualitative and intuitive. Background domain knowledge is leveraged to specify domain specific rules to be satisfied by patterns of data from available sensors. Simple Rule Based matching will be useful due to its ease in interpretation, generation and instant classification of new instances [20]. Preliminary context features are used in rule preconditions to derive weather context sought here

**Table 4. Definition of Context and its Classes**

| | Values of Context | Features of Context |
|---|---|---|
| **Context (Weather)** | Pleasant | Normal Temperature, Comfortable Humidity, Normal Light |
| | Comfortable | Normal or Mild Temp, quite humid, Normal Light |
| | Suitable_to_work | Hot or Cold Temperature, quite humid and Normal Light |
| | Not_Suitable_to_work | Hot Temp, Humid and Dim Light or dark |
| | Uncomfortable | Very Cold or Very Hot Temp, Very Humid or Dry, Dim Light |

Few instances of rules that will be used by classifier are:

$r_1$: (Temp=3) ^ (Humidity = 2) ^ (Ambient_Light = 5) → *Weather = Pleasant*

$r_2$: (Temp=3 or 4) ^ (Humidity = 3) ^ (Ambient_Light = 5) → *Weather = Comfortable*

$r_3$: (Temp =2 or 4) ^ (Humidity = 3) ^ (Ambient_Light = 4 or 5) → *Weather = Suitable_to_work*

$r_d$: ( ) → *Weather =Suitable_to_work*

In case of conflict in rule matching, the criteria for matching will be majority voting. The rule with the greatest number of antecedents matching is applied. The default rule triggers if the input sensor pattern doesn't match any rule. The default rule, if triggered, classifies the instance in majority class. In the rule set above last rule is the default rule as the case should always be. It has been found to be most frequent prevalent context from available data. As the testing data is not labelled with actual contexts, heuristic validation was applied to find that results are significantly correct. For example, if the arriving instance is Temp=2, Humidity =4 and Ambient_Light = 4; none of the rule matches exactly, so $r_3$ that has maximum antecedents matching is fired and accordingly context is mapped as *"Suitable_to_work"*.

All the steps towards context extraction are scalable to increase in number of input modalities for defining features of context [21]. The increase in type of such modalities will improve the semantics of context. For example, sensors to define presence of no. of persons in the lab can improve the extraction of current weather. The increase in number will make context extraction more accurate [22]. It would take an expert's opinion or testing on already tagged data to quantify the amount of improvement achieved in context extraction. But given the mechanism of extraction, it can be concluded that it will be at least proportional to the improvement in feature extraction.

# 5. CONCLUSIONS

Wireless Sensors generate lot of data about the phenomenon they are sensing. Thus making sense out of this data in real time is challenging. Being sent wirelessly, the data is also prone to errors. In this paper use of Bayesian Belief Networks has been demonstrated to improve quality of sensor data by recovering missing values and detecting outliers. The cleaned data is then used for context extraction. Mechanism to implement energy efficient sampling has also been included. Five of the BBN learning algorithms are evaluated for doing these tasks. Good results are obtained in terms of accuracy and computation time taken to predict the feature values. The actual and inferred features thus can be used in a simple rule based system to abstract the desired context which is current weather of the indoor environment under study. The methods described here are applicable to any new application with its own set of contexts and corresponding features.

# 6. REFERENCES

[1] IST Advisory Group, Scenarios for Ambient Intelligence in 2010, European Commission, 2001.

[2] A. Subramanya,A. Raj,J. Bilmes, and D. Fox, " Recognizing activities and spatial context using wearable sensors," In proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI), 2006.

[3] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni, " A survey of context

modelling and reasoning techniques", *Pervasive Mobile Computing* 6,( 2), April 2010, 161-180.

[4] Mittal, S.; Aggarwal, A.; Maskara, S.L., Application of Bayesian Belief Networks for context extraction from wireless sensors data , In Proceedings of 14th International Conference on Advanced Communication Technology (ICACT),2012 , Page(s): 410 - 415 ,2012.

[5] J. Cheng and R. Greiner, Learning Bayesian Belief Network Classifiers: Algorithms and System, *Lecture Notes in Computer Science*, (2056) pages 141.151, Springer Verlag, 2001.

[6] Friedman, Geiger and Goldszmidt (1997)] Friedman, N., Geiger, D., Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, pp. 131 – 163.

[7] GF Cooper, E Herskovits, "A Bayesian method for the induction of probabilistic networks from data", Mach Learning 9(4):309–347(1992)

[8] O.C.H. François and P. Leray. Learning the tree augmented naive bayes classifier from incomplete datasets. In Proceedings of the Third European Workshop on Probabilistic Graphical Models (PGM'06), pages 91–98, Prague, Czech Republic, Sep 2006.

[9] J. Heckerman, D., Meek, C. & Cooper, G. (1999). A Bayesian Approach to Causal Discovery. In Glymour, C. and G. Cooper, (ed.), Computation, Causation, and Discovery, 141-165. MIT Press.

[10] S. B. Kotsiantis. : Supervised Machine Learning: A Review of Classification Techniques. In Proceeding of Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, 2007.

[11] Leray P, Francois O: BNT structure learning package: documentation and experiments. Technical Report 2004.

[12] Intel Berkley Research lab [Online]. http://db.csail.mit.edu/labdata/labdata.html

[13] A. C and Y. PS, "A framework for clustering uncertain data streams," in Proceedings of IEEE 24rd International Conference on Data Engineering, 2008, pp. 150–159.

[14] J. Han and M. Kamber, Data mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann, 2009.

[15] Francesco Chiti, Romano Fantacci, Francesco Archetti, Enza Messina, and Daniele Toscani, "An integrated communications framework for context aware continuous monitoring with body sensor networks", *IEEE Journal of Selected Areas in Communication*, 27(4) (May 2009), 379-386.

[16] M.Raymer, T. Doom, L. Kuhn, and W. Punch, "Knowledge discovery in medical and biological datasets using a hybrid bayes classifier/evolutionary algorithm raymer", IEEE Trans Syst., Man, Cybern. B, Cybern., vol. 33, no. 5, pp.802 - 813 , 2003.

[17] Pham D, Ruz G (2009) Unsupervised training of Bayesian networks for data clustering. Proc Royal Soci A 465(2109):2927–2948.

[18] Chandola, A. Banerjee and V. Kumar (2007) Outlier detection: a survey, Technical Report. Univeristy of Minnesota, USA.

[19] Bruno M. Nogueira, Tadeu R. A. Santos, and Luis E. ZÃ¡rate. Comparison of classifiers efficiency on missing values recovering: Application in a marketing database with massive missing data. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007), 2007.

[20] W. W. Cohen, "Fast effective rule induction," in Proc. of the 12th Intl. Conf. on Machine Learning, 1995, pp. 115–123.

[21] Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: An Ontology-based Context Model in Intelligent Environments. In: Proceedings of communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, USA, pp. 270–275 , 2004.

[22] Sangeeta Mittal,Alok Aggarwal, S.L. Maskara, "Contemporary Developments in Wireless Sensor Networks", International Journal of Modern Education and Computer Science, vol.4, no.3, pp.1-13, 2012.