Application of Bayesian Belief Networks for Context Extraction from Wireless Sensors Data

Sangeeta Mittal*, Alok Aggarwal*, S.L. Maskara**

* Department of CS&IT, Jaypee Institute of Information Technology, A-10, Sector-62, Noida, UP, India

**G-2W, Soura Niloy Housing Complex, 1, Kailash Ghosh Road ,Kolkata, WB, India

snj_mittal@rediffmail.com, alok.aggarwal@jiit.ac.in, maskara.shankar@gmail.com

Abstract— **Networks of Wirelessly connected low power sensors have ability to closely sense activity of individual and social interest. The usefulness of Wireless Sensor Networks is increased further by deriving contextual information from it. From sensors data, context like activity, location, weather and surroundings (nearby persons, devices) can be deduced. Techniques to represent & extract the context include ontology, Markov Models, decision trees, clustering and Bayesian approaches. Given incomplete and erroneous nature of sensor data, Bayesian Belief Networks (BBN) are used here to obtain features defining context. Five algorithms of BBN construction have been evaluated for comparing feature classification performance. Simple rule based matching is then applied to map the features to already defined context. The mechanism is applied here on sensors data obtained from Intel research lab at Berkeley to extract the "weather" context. Similar mechanism can be applied to other application and contexts also.**

Keywords—**Wireless Sensor Networks, Bayesian Belief Networks, Sensor data classification, Context Extraction.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are networks of wirelessly connected sensors that continuously emit data about the environment in which they are placed. Applications of such networks are in the area of military and civil defense, health care and environment monitoring etc. The information sensed is mostly about the physical parameters like Body Temperature, Blood pressure, Heart beat, Ambient Temperature, Humidity, Wind Speed, Seismic activity using Richter scale etc. The sensed data is then transmitted to defined destinations for taking remedial actions or enhancing the understanding of environment or activity. Related knowledge is then have to be extracted from this data and presented to the user. Over the past decade, sensor technologies have advanced in sensor hardware, routing mechanisms and data interpretation to gather meaningful information from even remote places. Contextual knowledge such as location, activity, environment, physical conditions etc have to be extracted from the physical parameters available as raw sensor data [12]. Though it may be possible to sense, measure and transmit the necessary raw contextual information, but it may be economical and more efficient to extract the contextual knowledge from the physical parameters. Further in view of the large amount of information, extraction of contextual knowledge has to be done by machines [10]. In literature several machine based tools like Neural Networks, Decision Trees and Hidden Markov Models [1]. In data from wireless sensors are random in nature and as such require the use of probability and random theories for interpretation. In most of the situations the information and the associated knowledge are random in nature and as such require the use of probability and random theories.

In this work the context we seek to extract is "weather" of Intel's Berkeley Lab at a particular time in a particular area. The context has been defined in terms of various weather related attributes i.e. temp, humidity and ambient light values. The values for these are obtained from wireless sensors deployed in the lab. A two step mechanism of extraction of context from raw sensors data has been worked out. The first step is to model relationships among attributes of dataset by Bayesian Belief Networks using various BBN construction algorithms. The performance of these algorithms is evaluated in third section to judge their suitability of use with respect to accuracy and computational time, which are primary requirements in context extraction from real time WSNs. The models are used for deducing features of context i.e. class of temperature, humidity, ambient light and time of the day from observable sensor data. This makes the feature extraction immune to WSN noise. In the second step, which is discussed in fourth section, the obtained features are matched against a rule based system to obtain the sought context. These steps are distributed in two layers of a wireless sensor network model namely individual sensor nodes and the WSN gateway node. The related details are given in next section. The paper is concluded in the last section.

II. MODELING FEATURES OF CONTEXT WITH BAYESIAN BELIEF NETWORKS

Bayesian Belief Networks (BBNs) are stochastic models that describe and quantify probabilistically the relationship between one or more feature variables and a class variable. BBNs provide a graphical representation of the independences between the modelled variables. 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 [2]. Standard measures used for performance evaluation of classifiers are *Classification Accuracy, Error Rate and Rejection Rate*. A comparative study of Bayesian Belief Network structure learning algorithms is done here. It will help establish suitability of algorithms for use in classification of wireless sensors' data. A similar study will hold for all problems that are prognostic in nature. BBNs are used in this work, as per figure 1 to facilitate context extraction. The working is distributed at two layers of

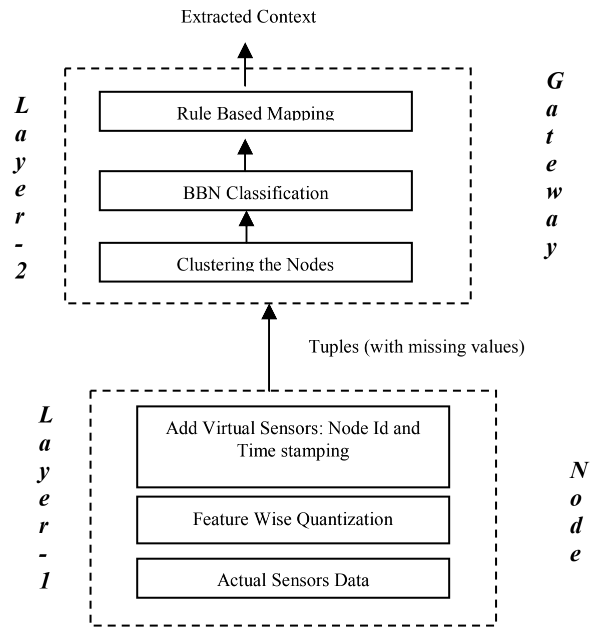


Figure 1. Mechanism of Context Extraction from Raw Sensor Data

WSN. 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 by machine learning methods which may not perform well on continuous real valued data. This layer after adding location & time information transmits the quantized tuples, which are used as primary context feature. This data is utilized by the second layer at the gateway node. In this layer the semantics of context are defined and an automatic procedure to classify context features into abstract context situations is laid. The classification component based on BBNs stores probabilistic relations in various features quantitatively in compact form. If the model using one hourly data performs satisfactorily then significant reductions can be obtained in number of transmissions as nodes can transmit after half or one hour instead of 30 secs and still similar performance on maintaining weather related information can be achieved. In case a value goes missing during transmission BBN based inference can provide that value. Features obtained in such way provide a reliable and energy efficient context after rules are applied on them

A. Context Dataset and Models :

The context dataset here is a real wireless sensor data set .The data has been collected from 54 Mica2Dot sensors with weather boards deployed in planned non uniform manner in the Intel Berkeley Research lab between February 28th and April 5th, 2004. The data is provided by the owners at [5] for research use. The area of the lab is approximately 41* 31 sq. m. Sensors send time stamped node id information, along with humidity, temperature, light and voltage values once every 31 seconds to the gateway node. The schema of database for storing this data is as given in figure 2.

date: yyyy-mm-dd	time: hh:mm:ss	Epoch :int	Moteid :int	Temperature real	Humidity :real	Light :real	Voltage :real

Figure 2. Schema of the Sensor Database

The attributes of this dataset are used as features for defining context in this case.

Data Preprocessing:

The sensors data was highly noisy and had lot of missing and erroneous values. The tuples with missing values were discarded. 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. Frequently such data was encountered due to spikes in sensor signals.

Quantization:

All the fields are quantized to obtain discrete categories. The dates are converted into day numbers starting from 28^{th} feb as day-1. The time is quantized into 24 1hr time steps for one day, starting from 00:00:00 hrs to 01:59:59 as time step 1. The epoch id is not used for classification. The weather context is not represented by readings of a single sensor hence the motes are also clustered into thirteen groups, named from G1 to G13, based upon their physical proximity. Figure 3 shows the arrangement of sensors in the lab and our clustering scheme in bold rectangles.

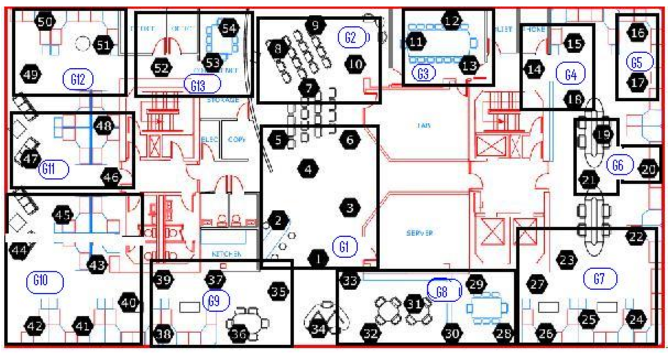


Figure 3. Clustering of Sensors placed in a 41*31 sq.m Indoor Area

The data though is of an indoor, probably controlled, environment but has e.g. temp variations of up to 9 degrees at a particular time in the lab. The reasons for this kind of variation may be due to varied sunlight during the day in different parts of lab. Lower temperature may owe to location of the sensor near windows and air vents. At night time, areas near windows are colder, while at sun rise the east side of the lab report increased temperature due to sun. The lab is seen to be uniformly warm 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. All the three sensor data from motes are continuous. Owing to the use of machine learning methods that perform not so well on continuous values, features are divided into several categories as thought suitable in general for these physical quantities. Simple thresholding has been applied to divide the temperature into 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.

TABLE I. 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 quantized values of the features replaced the actual values in the data sets accordingly. The readings from nodes are to be seen cluster wise instead of individually for reasons stated above. Hence per hour average of each cluster was computed to obtain attribute values and hence classes. Design of preprocessing is a onetime activity for any application and only required to be changed when quantification criteria are modified or new features are added. In the next subsection, the datasets of feature values are used to learn the Bayesian Belief Network among them.

B. Construction of Bayesian Belief Networks - Algorithms

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 BBN_S, which is a Directed Acyclic Graph (DAG) over U and a set of probability tables BBN_p , where

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

where $pa(u_i)$ is the set of parents of u_i in BBN_S[4]. A Bayesian Belief Network also represents joint probability distribution on whole set of variables, U as

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

For a given problem, BBNs can be obtained by experts of that domain. In absence of an expert, alternatively algorithms making use of statistics, graph algorithms and information theory are used to autonomously find most probable BBN given experiential dataset, D of that domain.

Many algorithms combining principles from all three fields, for learning Bayesian Belief Networks automatically from data have been proposed in literature. These can be subdivided into at least two categories: methods based on conditional independence tests, and methods based on a scoring function and a search procedure [2]. Algorithms in first category face the problem of '*Memory Crash'* with graphs of order of more than ten nodes. This happens due to exhaustive calculation of independence among variables. The algorithms mentioned below are chosen as representative samples from second category [3].

Naïve Bayesian Structure (A1): A Naïve Bayesian Structure is a static structure representing problem domain where class variable is root and all other variables are only dependent on it and independent of each other.

Maximum Weighted Spanning Tree Structure (A2):

This method associates a weight to each edge in an initial DAG. This weight can be either the mutual information between the two variables or the score variation when one node becomes a parent of the other. When the weight matrix is created, a usual Kruskal's or Prim's MWST algorithm gives an undirected tree that can be oriented given a root.

K2 Algorithm(A3):

Another metric for reducing search space has been devised by Cooper et al in [4]. The method requires an initial node order to be provided. Here three types of orders have been tried, as follows: using the MWST DAG *(A3)*, reversed MWST *(A4)* and a randomly ordered DAG *(A5)*. 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 [4].

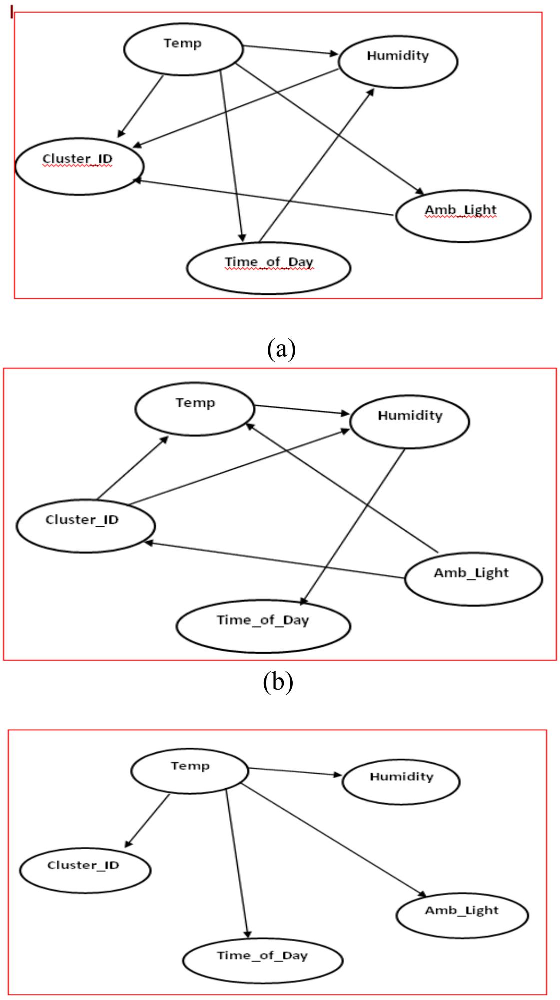
$$\max_{B_s}[P(B_s, D)] = \prod_{i=1}^{n}[P(\pi_i \to x_i)\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}N_{ij}\prod_{k=1}^{r_i}N_{ijk}! \qquad (3)$$

This algorithm requires feature variables to be discrete valued and complete. A detailed description of these algorithms can be found in [3].

Greedy Search (A4)[8] and Hill Climbing Search (A5)[9]:

These two algorithms come under score & search and differ only in searching methods namely first being greedy that chooses the next DAG with maximum score and second hill climbing to jump away from local maxima.

The algorithms discussed here are used here are implemented using BNT Structure learning package [6] The relations between various features of context dataset described earlier are then modelled using it. The sensor data as per discussion is obtained for approx 28 days per cluster per time period is used for learning BBN models. The nodes of such models are Cluster Ids and Time of the day, the virtual sensors and temp, humidity and ambient light, the actual sensors. The relationships between these sensors or feature variables were obtained using the algorithms discussed earlier. Models extracted by few of the algorithms are also represented in Figure 4.



(a) (b) (c)

Figure 4. Bayesian Belief Network obtained using (a) K2+MWST (b) GS (c) Naïve Bayesian Network

It can be seen that the obvious relationship 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. Once the Network Structures were determined, the conditional probability tables for each of the variables were estimated using maximum likelihood estimation (using frequency counts) in all cases [10]. This provided the complete quantitative models of all the features.

III. BAYESIAN BELIEF NETWORKS BASED FEATURE CLASSIFICATION

Wireless Sensor Data is inherently noisy and has frequent spikes due to dynamic nature of wireless medium. Hence data reaching the decision making node is error prone. BBN based classifier when applied on available data can provide the values of missing data and also predict with probabilistic confidence whether it is erroneous or not. In general, the classification task consist of classifying a variable, C called the class variable given a set of variables A = $a_1 \ldots a_n$, called feature variables. The classifier Z: A → C is a function that maps an instance of A to a value of C. Given dataset D over U, set of BBNs like ones constructed in previous section a probabilistic classifier can be constructed. To use a BBN as a classifier, $\text{argmax}_y P(C|A)$ using the distribution P(U) represented by the BBN is calculated. i.e.

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

Maximizing LHS, α 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 updating* in BBNs.**

There are exact inference algorithms for same, mainly performing inference by enumeration [9]. These algorithms though work only for BBNs having *polytree* structure. Up to an order of 35 nodes, exact inference work well, but becomes computationally intractable for larger networks hence several approximate algorithms making use of randomized sampling, also called Monte Carlo algorithms are to be used. Refer [9] for details of various inference algorithms.

In real world BBNs, attributes are generally multiply connected. Hence a well known algorithm for such trees namely Junction Tree (JTree) Algorithm (Clustering + Variable Elimination) is used here [9]. JTree is an exact clustering algorithm that performs inference in two stages:

1. Transform the multiply connected network into a polytree
2. Perform belief updating on the obtained polytree using message passing mechanism.

The datasets available is only for 28 days that is too small to be divided into separate parts for satisfactory training and testing. Hence 5-Fold 10 Times Cross-Validation has been used to generate both sets. The Class variable in the dataset is the feature variable whose value needs to be determined. Multiple categories of classes in each feature have been described in Table 1 already.

The Confusion matrix is used to represent the number of misclassified test instances. Confusion Matrix is created to assess the ability of classifier to distinguish all class values appropriately with high accuracy. One such matrix for humidity classification is shown table II & other for light in table III. Due to shortage of space same for other algorithms and features is not shown here. 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. So there is a tradeoff between number of clusters and accuracy. As increasing the number will increase the processing & storage requirements but will also provide more granular data. From table III it is noted that the accuracy in light is lesser than that of humidity. Moreover the spills of misclassification are not merely to adjacent classes but spill over further also. This may be due to the fact that though light exhibits pattern behavior during 24hrs time but is easily hindered by a presence of some object near the sensor.

TABLE II. CONFUSION MATRIX FOR CROSS VALIDATION TESTING OF HUMIDITYCLASSIFICATION WITH K2+MWST

Actual Class Values	Predicted Class Values			
	1	2	3	4
1	23	14	0	0
2	4	1289	468	44
3	0	356	2635	265
4	0	60	418	1669

TABLE III. CONFUSION MATRIX FOR CROSS VALIDATION TESTING OF AMBIENT LIGHT CLASSIFICATION WITH K2+MWST

Actual Class Values	Predicted Class Values					
	1	2	3	4	5	6
1	107	64	15	27	20	4
2	57	1688	225	141	40	6
3	3	442	1324	442	20	11
4	27	235	348	1706	65	26
5	18	51	12	107	432	54
6	0	0	32	55	67	398

The accuracy was also tested using only the time of the day and cluster id as available data, reasoning done on BBNs using merely these two values also gave up to 55-60% correct results for temperature as well as humidity. While in classifying temperature given all other features mainly adjacent classes were confused with.

IV. EVALUATION OF CLASSIFICATION

Classifiers obtained are used for inferring test instances with classes. 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. In Figure 4, summary of accuracies obtained (between 0-1) after classifying multiclass sensor data is shown. Algorithms are also to be 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. The accuracy rate of K2 based algorithms is overall better considering total classification instances. 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 will provide significant energy savings while not compromising information.

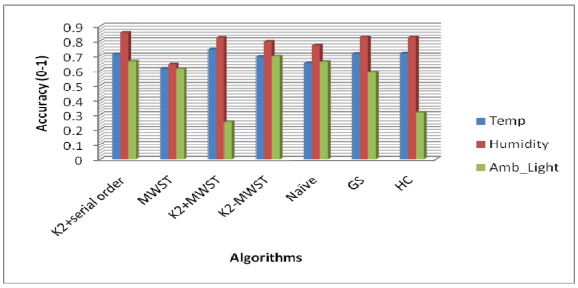


Figure 5. Accuracy of various learning algorithms in classifying various features

Computational time (CT) taken for classification is important for any real time classification. All the algorithms were run here using a core 2 duo processor of 2.4GHz clack Speed. The RAM Size was 2 GB. It is observed that on given data there is not much variations in computation time taken by all the algorithms. Greedy Search and Hill Climbing take more time to classify due to their search space being larger than those of other algorithms used here. Based upon the parameters used here for evaluation no single class performs overall best.

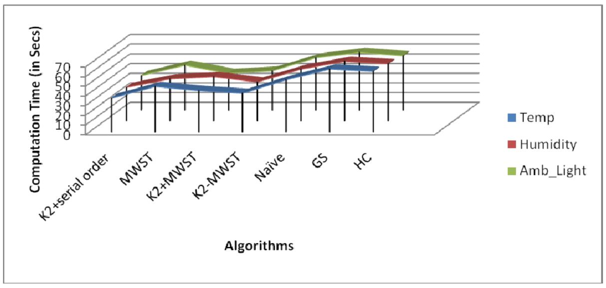


Figure 6. Computational Time Taken by different algorithms on a 2.4Ghz Clock Speed machine

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.

V. RULE BASED CONTEXT EXTRACTION

The "context" to be extracted from the data is the weather inside the lab. The classes defined are pleasant, comfortable, suitable for work, not suitable for work and uncomfortable.

The qualitative description of context to be extracted here from low level sensor based context features are described in table 4. Simple Rule Based matching will be useful due to its ease in interpretation, generation and instant classification of new instances [15]. Preliminary context features are used in rule preconditions to derive *weather* context sought here. The rules are defined based upon personal opinion. The description of context to define rules is given in table 4. As said before, this description is qualitative and intuitive.

TABLE IV. 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*

Twenty such rules are defined using the data available. The system is used in real scenario so the mapping needs to be done on new data. In case of conflict in rule matching, identifying the rule with the greatest number of antecedents or picking the first rule matched is applied. The default rule that is triggered if the current data doesn't match any rule, 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 probable prevalent context from available data. As the testing data is not labeled 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"*.

As a future work it is proposed to develop methods that do fuzzy matching to avoid 'somewhat' similar instances mapping to default rule. The context definition itself can be enhanced to include more features like time of the day. These additions will make context extraction more adaptive and dynamic. Semantics of more such contexts like "location" can be defined and found from same set of data.

VI. CONCLUSION

Bayesian Belief Networks are useful for representing in compact manner knowledge about an inherently uncertain domain like sensor data. An important use of BBN is for classification in decision support systems. In this paper use of BBN is made to represent missing values from quantized features. Five of the BBN learning algorithms were evaluated with respect to their classification performance for predicting same. The learnt features are used in a rule based system to abstract the desired context which is current weather of the indoor environment under study. Realistic contexts that characterize a situation may be composed of more features, the simple rule based classification may not work then and some fuzzy method may have to be developed. The computation time and processing requirements will be main constraints in any such method.

REFERENCES

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

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

[3] 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.

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

[5] Intelberkely research lab [Online]. http://db.csail.mit.edu/labdata/labdata.html

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

[7] 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.

[8] 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.

[9] K Korb, Bayesian Artificial Intelligence, CRC Press,2004.

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

[11] 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.

[12] 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.

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