Wireless Sensor Network Project Report on

Classification of ADL data collected from WSN

Anusha Prakash (12IT10)

Bhuvan M S (12IT16)

Siddharth Jain (12IT78)

Under the Guidance of,

MS. Deepthi

Department of Information Technology, NITK Surathkal

Date of Submission: 21 April 2016

in partial fulfillment for the award of the degree

of

Bachelor of Technology In **Information Technology** At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

April 2016

Department of Information Technology, NITK Surathkal
Wireless Sensor Network Project
End Semester Evaluation Report (April 2016)

Course Code : IT410

Course Title: Wireless Sensor Network Project

Project Title: *Classification of ADL data collected from WSN*

Project Group:

Name of the Student	Register No.	Signature with Date
Anusha Prakash	12IT10	
Bhuvan M S	12IT16	
Siddharth Jain	12IT78	

Place:NITK, Surathkal

Date:18 April 2016 *(Name and Signature of Wireless Sensor Network Project Guide)*

Abstract

Activities of Daily Life (ADLs) basically describe a person's day to day doings. ADLs include sleeping, leisure activities, eating and anything that a user does on a daily basis. Wireless sensors can be used to capture these ADLs. After recording the activities, proper studies and analysis can be done to predict user's behavior and build smart houses based on the analysis. In this research, multiple machine learning algorithms like Naive Bayes, Multilayer Perceptron, Bayes Network etc. have been considered to determine which classifier gives the best results in a cost effective manner. A simulation of HEED protocol is also done to show how data can be collected using such protocols.

Keywords: *Wireless Sensor Network; Machine Learning CLassifier; Naive Bayes; Multilayer Perceptron; Bayes Network; Random Forest; Adaboost; Logistic Regression*

Contents

List of Figures

List of Tables

1 Introduction

Population aging is currently having a significant impact on health care systems. Improvements in medical care are resulting in increased survival into old age, thus cognitive impairments and problems associated with aging will increase. It has been estimated that one billion people will be over the age of 60 by the year 2025. As the burden of healthcare on society increases, the need for finding more effective ways of providing care and support to the disabled and elderly at home becomes more predominant. Monitoring human activities of daily living (ADL), in order to assess the cognitive and physical well being of elderly, is considered a main aspect in building intelligent and pervasive environments. Systems that recognize ADL from sensor data are now an active topic of research; indeed diverse approaches have been proposed to deal with the activity recognition problem, ranging from video cameras, RFID readers and wearable sensors. However, Wireless Sensor Networks (WSN) are considered one of the most promising technologies for enabling health monitoring at home due to their suitability to supply constant supervision, flexibility, low cost and rapid deployment. Besides, the inherent non-intrusive characteristics of these networks have been proved to suit perfectly with environments where privacy and user acceptance is required. Previous approaches have shown how simple binary sensors have solid potential for solving the ADL recognition problem in the home, and can be applied in human-centric problems such as health and elder care. Binary sensors measuring the opening or closing of doors and cupboards, the use of electric appliances, as well as motion sensors were used to recognize ADLs of elderly people living on their own. Indeed, this kind of sensors is considered one of the most promising technologies to solve key problems in the ubiquitous computing domain, due to their suitability to supply constant supervision and their inherent non-intrusive characteristics. In this research, we evaluate and compare the activity recognition performance of these models on multiple fully annotated real world datasets: three well known datasets generated by Kasteren et al., and two new datasets (OrdonezA and OrdonezB). This kind of approach has been previously applied for recognizing human activities using wearable devices as well.

2 Literature Survey

2.1 Problem Statement

The aim of our project is to classify the ADL of the user using the data sensed by sensor nodes in the wireless sensor networks by modeling separate user dependent machine learning model and analyse the performance of various classifiers.

2.2 Objectives

1. To develop ADL prediction model on the data collected from sesor nodes in a Wireless Sensor Network.

2. Evaluate each prediction model using Receiver Operating Characteristics (ROC), Precision-Recall Curves, F1 Measure.

3. To implement simple version of HEED protocol to illustrate how the ADL data could be collected from the Wireless Sensor Network.

Table 1: Features details

Feature	**Type**
Start Timestamp	Numeric
End Timestamp	Numeric
Duration	Numeric
Location	Nominal
Type	Nominal
Place	Nominal
Class: Activity	10 Nominal Classes

3 Methodology

3.1 ADL Classification Model

3.1.1 Data description

This dataset comprises information regarding the ADLs performed by two users on a daily basis in their own homes. This dataset is composed by two instances of data, each one corresponding to a different user and summing up to 35 days of fully labelled data. Each instance of the dataset is described by three text files, namely: description, sensors events (features), activities of the daily living (labels). Sensor events were recorded using a wireless sensor network and data were labelled manually.

3.1.2 Work Flow

The workflow of the methodology has been described in the Figure 3.1

3.1.3 Data Preprocessing

The cleaned, preprocessed and normalized data contains 408 number of instances for User-A and 2334 number of instances for User-B. The features are tabulated in the Table 1. The classes (ADL) are: Spare$_{T}ime/TV, Grooming, Toileting, Sleeping, Breakfast, Showering, Snack,$

3.1.4 Machine learning algorithms

Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Such algorithms operate by building a model from example inputs and using that to make predictions or decisions, rather than following strictly static program instructions.

We have chosen following machine learning algorithms for the project.

3.1.4.1 Naive Bayes In machine learning, naive Bayes classifiers [5] are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method; Russell and Norvig note that "[naive Bayes] is sometimes called a Bayesian classifier, a somewhat careless usage that has prompted some Bayesians to call it the idiot Bayes model."

3.1.4.2 Multilayer Perceptron - 2 nodes in one hidden layer Multi-Layer Perceptron Classifier-2 (MLPC 2):A multi-layer perceptron (MLP) is a feed-forward artificial neural network model [6] that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable. It is trained with one hidden layer by minimizing the squared error plus a quadratic penalty with the BFGS method. The hidden layer was designed to have 2 nodes.

3.1.4.3 Bayesian Network A Bayesian network [7] is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph. Formally, Bayesian networks are DAGs whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not connected (there is no path from one of the variables to the other in the bayesian network) represent variables that are conditionally independent of each other. Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node. For example, if m parent nodes represent m Boolean variables then the probability function could be represented by a table of $2\hat{m}$ entries, one entry for each of the $2\hat{m}$ possible combinations of its parents being true or false. Similar ideas may be applied to undirected, and possibly cyclic, graphs; such are called Markov networks.

Efficient algorithms exist that perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g. speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

3.1.4.4 Random Forest Random Forest is composed of a set of decision trees. Each decision tree acts as a weak classifier and pooling the responses from multiple decision trees leads to a strong classifier [9]. Each decision tree is trained independently and determines the class of an input by evaluating a series of greedily learned binary questions. Random forests with 10 trees with no limit on maximum depth was used for experimentation.

3.1.4.5 Adaboost Adaboost constructs a strong classifier by sequentially combining a set of weak classifiers [10]. At first iteration, a single classifier is learnt to minimize the classification error. At each consequent iteration, a new classifier is learnt which seeks to minimize the error of the classifier composed of the set of classifiers learnt until the previous iteration. In all our experiments, decision trees were used as weak classifiers.

3.1.4.6 Logistic Regression Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Thus, it treats the same set of problems as probit regression using similar techniques, with the latter using a cumulative normal distribution curve instead. Equivalently, in the latent variable interpretations of these two methods, the first assumes a standard logistic distribution of errors and the second a standard normal distribution of errors [11].

3.1.5 Evaluation metrics

The following evaluation metrics are calculated for each prediction model.

3.1.5.1 Accuracy Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{1}$$

3.1.5.2 F1-Measure The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$$F1score = (2 * Precision * Recall)/(Precision + Recall) \tag{2}$$

3.1.5.3 Receiver Operating Characteristic Curve: ROC In statistics, a receiver operating characteristic (ROC) [2], or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. ROC analysis is related in a direct and natural way to cost/benefit analysis of diagnostic decision making.

The Area Under the Curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. This measure is used to evaluate different binary classifiers. Higher the ROC AUC the better is the classifier.

3.1.5.4 Precision and Recall Curve In pattern recognition and information retrieval with binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance.

1. Precision (P): is defined as the fraction of ground truth positives among all the examples that are predicted to be positive.

   $$Precision = TP/(TP + FP) \tag{3}$$

   Where, TP, FP stand for True Positives and False Positives respectively

2. Recall (R): is defined as the fraction of ground truth positives that are predicted to be positives.

   $$Recall = TP/(TP + FN) \tag{4}$$

   Where, FN denotes False Negatives.

3. Precision-Recall Curve [3]: The tradeoff between the precision and recall can be studied by looking at the plot of how precision changes as a function of recall. This plot is known as the Precision-Recall (PR) curve. The precision is plotted on the y-axis and recall on x-axis. The area under precision recall curve is a cost-effective metric for evaluation of classifiers. High value of area under the PR curve indicates better performance. Area under the PR curve is preferred way of studying the performance of classifiers in skewed (i.e. imbalance between positives and negatives) datasets such as the one considered in this work [3].

3.2 HEED Protocol

3.2.1 Need for HEED Protocol

Nodes in LEACH independently decide to become cluster heads. While this approach requires no communication overhead, it has the drawback of not guaranteeing that the cluster head nodes are well distributed throughout the network. While the LEACH-C protocol solves this problem, it is a centralized approach that cannot scale to very large numbers of sensors. One approach that uses a distributed algorithm that can converge quickly and has been shown to have low overhead is called HEED.

3.2.2 HEED Protocol Design and Working

Hybrid, Energy-Efficient Distributed Clustering (HEED) extends the basic scheme of LEACH by using residual energy and node degree or density as a metric for cluster selection to achieve power balancing. It operates in multi-hop networks, using an adaptive transmission power in the inter-clustering communication. HEED was proposed with four primary goals namely :

1. prolonging network lifetime by distributing energy consumption
2. Terminating the clustering process within a constant number of iterations
3. Minimizing control overhead
4. Producing well-distributed CHs and compact clusters

HEED uses an iterative cluster formation algorithm, where sensors assign themselves a cluster head probability that is a function of their residual energy and a communication cost that is a function of neighbor proximity. Using the cluster head probability, sensors decide whether or not to advertise that they are a candidate cluster head for this iteration. Based on these advertisement messages, each sensor selects the candidate cluster head with the lowest communication cost (which could be the sensor itself) as its tentative cluster head. This procedure iterates, with each sensor increasing its cluster head probability at each iteration until the cluster head probability is one and the sensor declares itself a final cluster head for this round. The advantages of HEED are that nodes only require local (neighborhood) information to form the clusters, the algorithm terminates in O(1) iterations, the algorithm guarantees that every sensors is part of just one cluster, and the cluster heads are well-distributed.

The HEED clustering improves network lifetime over LEACH clustering because LEACH randomly selects CHs (and hence cluster size), which may result in faster death of some nodes. The final CHs selected in HEED are well distributed across the network and the communication cost is minimized.

Table 2: Performance comparison of different classifiers for User A.

Algorithm	**F1_Score**	**Accuracy**
MLP_CLASSIFIER_2	0.328498863313	0.441176470588
LOGISTIC_REGRESSION	0.752633628582	0.764705882353
BAYES_NETWORK	0.720978373888	0.735294117647
NAIVE_BAYES	0.764874797931	0.764705882353
RANDOM_FOREST	0.64279135996	0.656862745098
ADABOOST	0.110876854162	0.254901960784

4 Results and Analysis

The classifier algorithm learns on the training data and saves a trained model, which is hence used to predict the probabilities of the occurrence of each class for each test instance. These scores are used to calculate the weighted area under the ROC curve and the weighted area under the PR curve, F1 score and plot the ROC and PR curves.

4.1 ADL Classification Model

THe results of the ADL prediction models for User-A and User-B have been discussed in following subsections.

4.1.1 User A

4.1.1.1 ROC and PR plots The ROC curves and PR curves for all algorithms for User-A have been presented in Figure 4.1 and Figure 4.3 respectively.

4.1.1.2 Comparison of Results Summary of the results containing the scores from various evaluation metrics as explained in the Methodology section is shown in Table 2.

4.1.2 User B

4.1.2.1 ROC and PR plots The ROC curves and PR curves for all algorithms for User-B have been presented in Figure 4.3 and Figure 4.3 respectively.

Table 3: Performance comparison of different classifiers for User B.

Algorithm	**F1_Score**	**Accuracy**
MLP_CLASSIFIER_2	0.278021577872	0.349914236707
LOGISTIC_REGRESSION	0.287792488475	0.360205831904
BAYES_NETWORK	0.317570449807	0.391080617496
NAIVE_BAYES	0.289539715817	0.36192109777
RANDOM_FOREST	0.309216828748	0.368782161235
ADABOOST	0.0966266437965	0.245283018868

4.1.2.2 Comparison of Results Summary of the results containing the scores from various evaluation metrics as explained in the Methodology section is shown in Table 2.

4.2 HEED Protocol Results

Figure 4.5 depicts the initial network architecture. It contains the node ID, node position, number of neighbour nodes and the IDs of neighbour nodes. The simulation consists of 100 nodes where each node initialized 4J of energy.

The distributed clustering algorithm is run for one iteration, i.e one TDMA schedule. During this, each node in the network gets assigned a schedule or a timeslot to transmit the data once. Figure 4.6 depicts the CH and energy level of the nodes before the TDMA schedule.

Figure 4.7 depicts the CH and energy level of the nodes after one TDMA schedule or after data transmission by nodes.

References

[1] Ordnez, Fco Javier, Paula de Toledo, and Araceli Sanchis. ”Activity recognition using hybrid generative/discriminative models on home environments using binary sensors.” Sensors 13.5 (2013): 5460-5477. APA

[2] Receiver Operating Characteristics. Wikipedia, The Free Encyclopedia.Web.2016.

[3] Precision Recall Curves. Wikipedia, The Free Encyclopedia.Web.2016.

[4] F1 Score. Wikipedia, The Free Encyclopedia.Web.2016.

[5] Naive Bayes. Wikipedia, The Free Encyclopedia.Web.2016.

[6] M. Lichman, UCI machine learning repository, 2013. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Grammatical+Facial+Expressions

[7] Bayesian Networks. Wikipedia, The Free Encyclopedia.Web.2016.

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[9] L. Breiman, Random forests, Machine learning, vol. 45, no. 1, pp. 532, 2001.

[10] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of computer and system sciences, vol. 55, no. 1, pp. 119139, 1997.

[11] Logistic Regression. Wikipedia, The Free Encyclopedia.Web.2016.

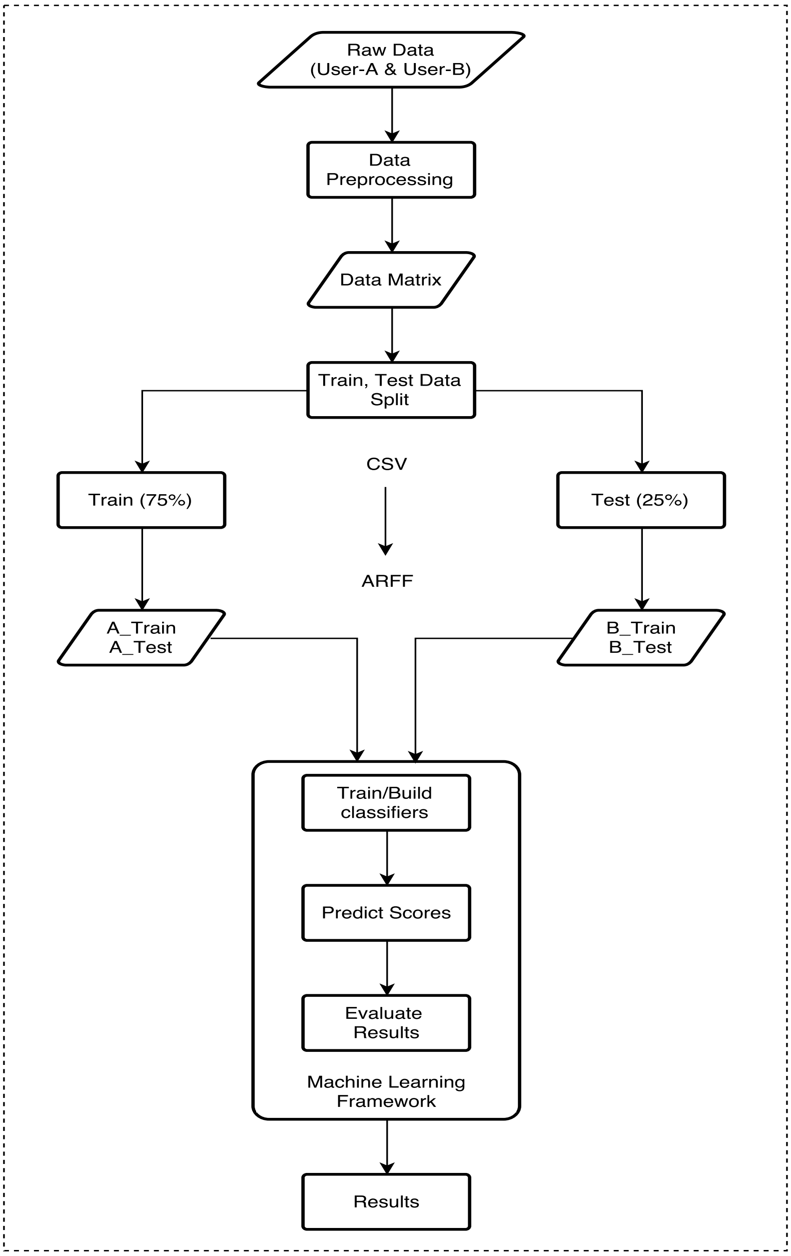
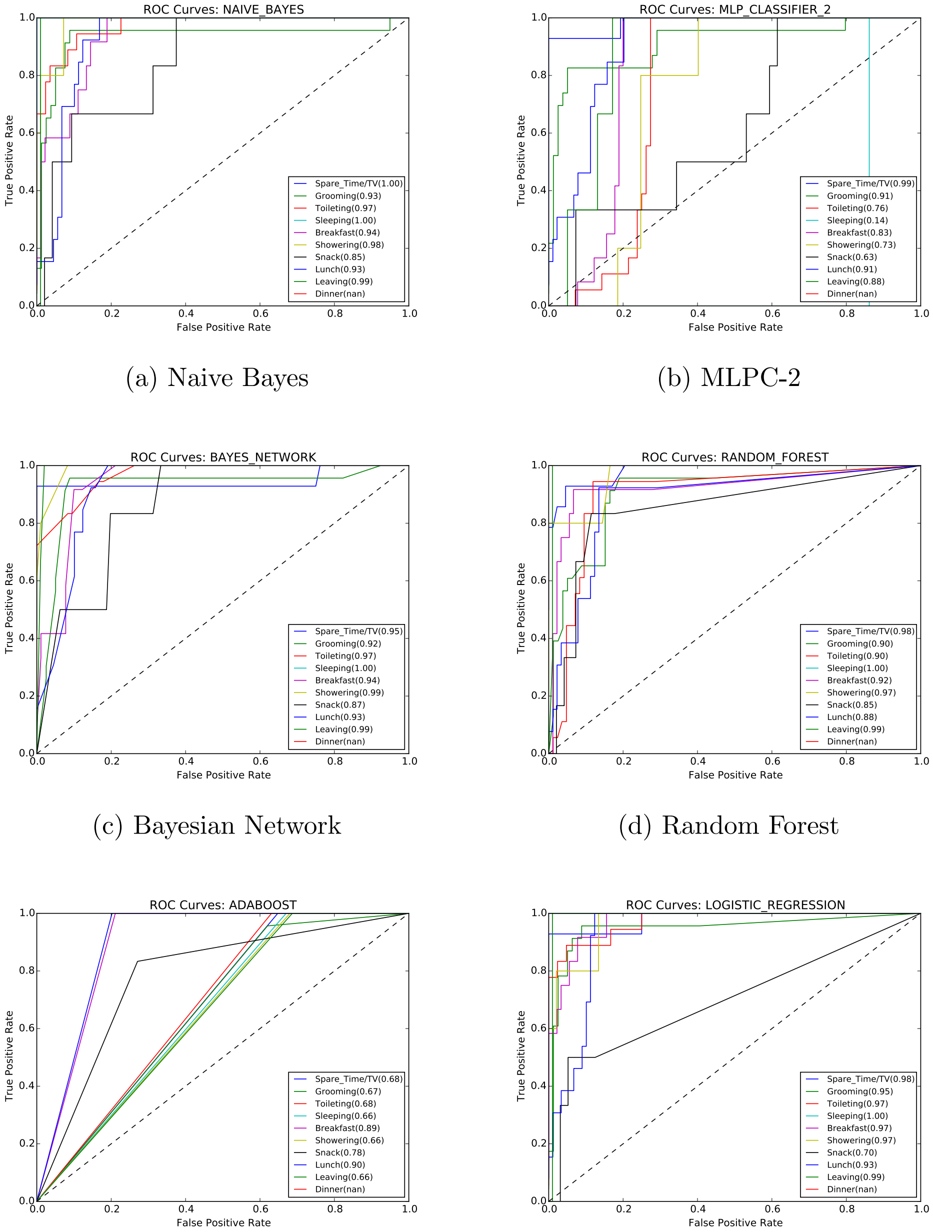


Figure 3.1: Flowchart representing the Methodology work flow.



(a) Naive Bayes

(b) MLPC-2

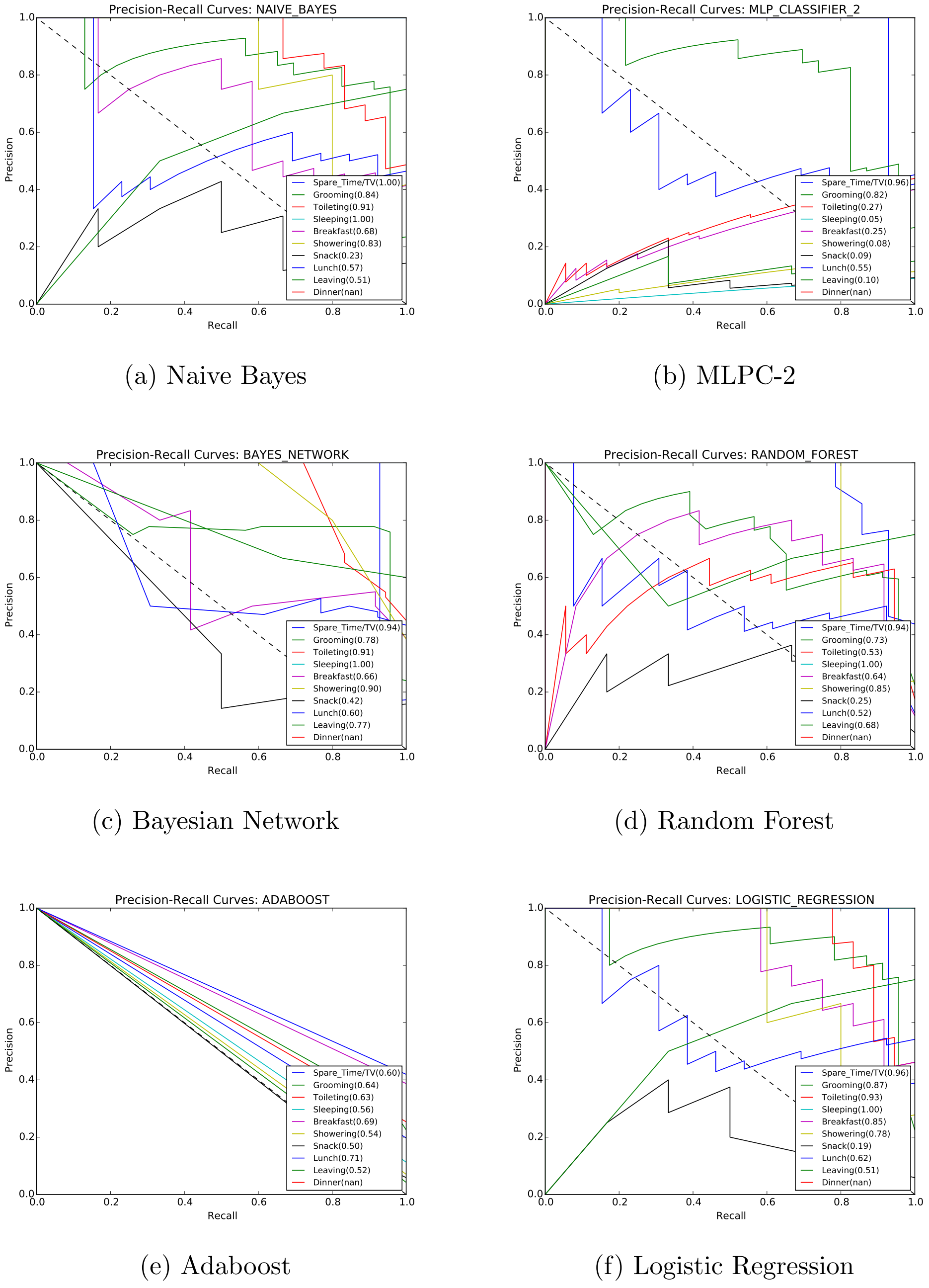
(c) Bayesian Network

(d) Random Forest

(e) Adaboost

(f) Logistic Regression

Figure 4.1: ROC curves for User A.



(a) Naive Bayes

(b) MLPC-2

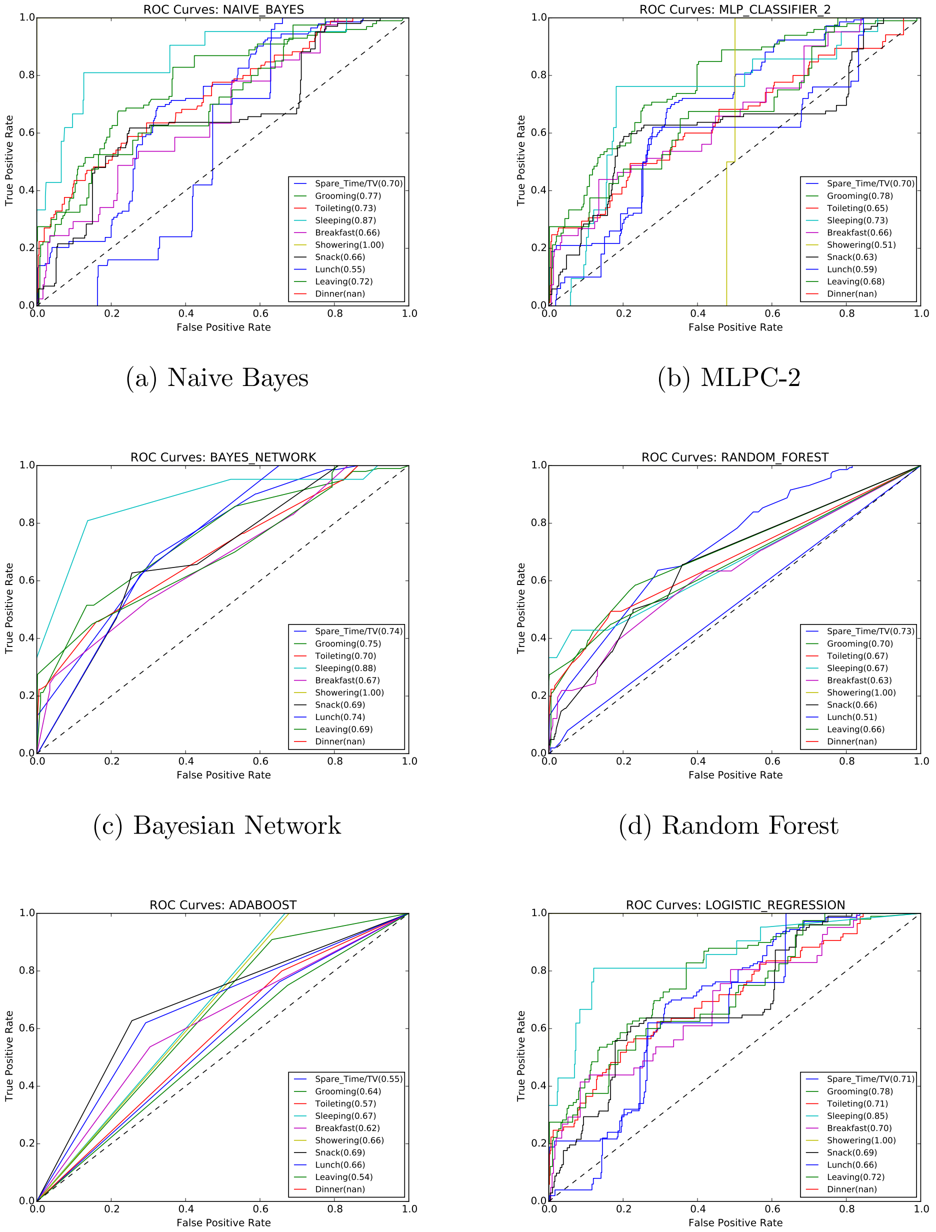
(c) Bayesian Network

(d) Random Forest

(e) Adaboost

(f) Logistic Regression

Figure 4.2: PR curves for User A.



(a) Naive Bayes

(b) MLPC-2

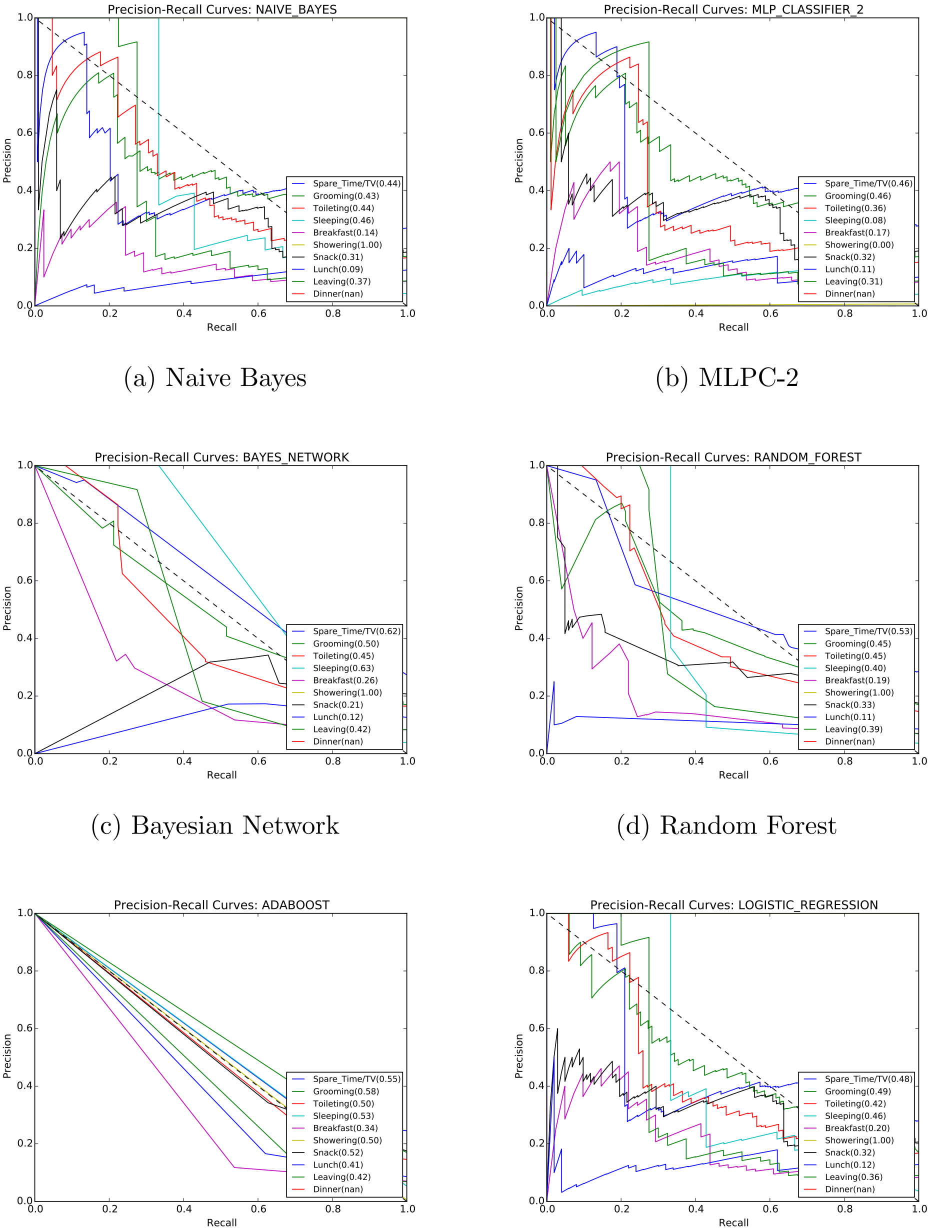
(c) Bayesian Network

(d) Random Forest

(e) Adaboost

(f) Logistic Regression

Figure 4.3: ROC curves for User B.



(a) Naive Bayes

(b) MLPC-2

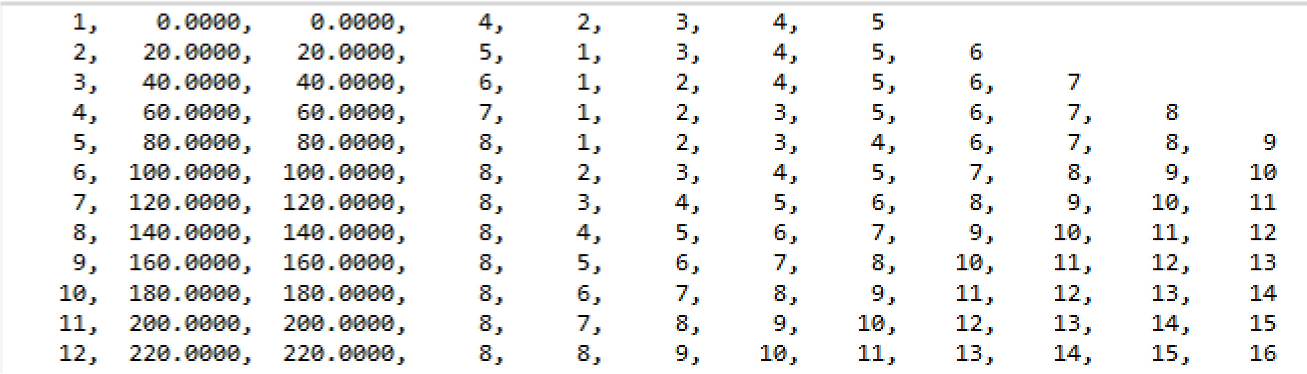
(c) Bayesian Network

(d) Random Forest

(e) Adaboost

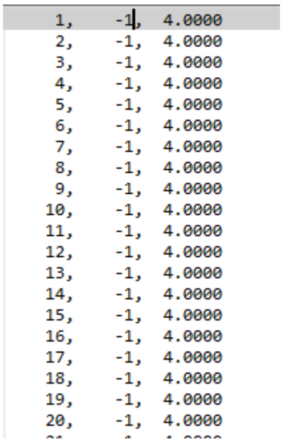
(f) Logistic Regression

Figure 4.4: PR curves for User B.



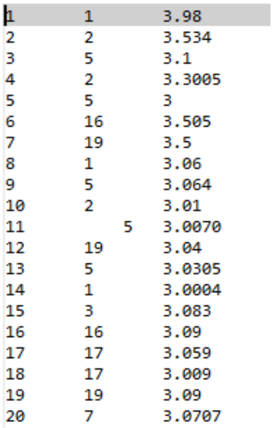
```
    1,     0.0000,     0.0000,      4,      2,      3,      4,      5
    2,    20.0000,    20.0000,      5,      1,      3,      4,      5,      6
    3,    40.0000,    40.0000,      6,      1,      2,      4,      5,      6,      7
    4,    60.0000,    60.0000,      7,      1,      2,      3,      5,      6,      7,      8
    5,    80.0000,    80.0000,      8,      1,      2,      3,      4,      6,      7,      8,      9
    6,   100.0000,   100.0000,      8,      2,      3,      4,      5,      7,      8,      9,     10
    7,   120.0000,   120.0000,      8,      3,      4,      5,      6,      8,      9,     10,     11
    8,   140.0000,   140.0000,      8,      4,      5,      6,      7,      9,     10,     11,     12
    9,   160.0000,   160.0000,      8,      5,      6,      7,      8,     10,     11,     12,     13
   10,   180.0000,   180.0000,      8,      6,      7,      8,      9,     11,     12,     13,     14
   11,   200.0000,   200.0000,      8,      7,      8,      9,     10,     12,     13,     14,     15
   12,   220.0000,   220.0000,      8,      8,      9,     10,     11,     13,     14,     15,     16
```

Figure 4.5: HEED Protocol Nodes and Positions



```
    1,     -1,   4.0000
    2,     -1,   4.0000
    3,     -1,   4.0000
    4,     -1,   4.0000
    5,     -1,   4.0000
    6,     -1,   4.0000
    7,     -1,   4.0000
    8,     -1,   4.0000
    9,     -1,   4.0000
   10,     -1,   4.0000
   11,     -1,   4.0000
   12,     -1,   4.0000
   13,     -1,   4.0000
   14,     -1,   4.0000
   15,     -1,   4.0000
   16,     -1,   4.0000
   17,     -1,   4.0000
   18,     -1,   4.0000
   19,     -1,   4.0000
   20,     -1,   4.0000
```

Figure 4.6: Node parameters before TDMA



```
1       1       3.98
2       2       3.534
3       5       3.1
4       2       3.3005
5       5       3
6       16      3.505
7       19      3.5
8       1       3.06
9       5       3.064
10      2       3.01
11          5   3.0070
12      19      3.04
13      5       3.0305
14      1       3.0004
15      3       3.083
16      16      3.09
17      17      3.059
18      17      3.009
19      19      3.09
20      7       3.0707
```

Figure 4.7: Node parameters after TDMA