Energy Prediction in Smart Environments

Chao Chen[a,1] , Barnan Das[a] and Diane J. Cook[a]
[a] *School of Electrical Engineering and Computer Science*
Washington State University

Abstract. In the past decade, smart home environment research has found application in many areas, such as activity recognition, visualization, and automation. However, less attention has been paid to monitoring, analyzing, and predicting energy usage in smart homes, despite the fact that electricity consumption in homes has grown dramatically. In this paper, we extract the useful features from sensor data collected in the smart home environment and select the most significant features based on mRMR feature selection criterion, then utilize three machine learning algorithms to predict energy use given these features. To validate these algorithms, we use real sensor data collected from volunteers living in our smart apartment testbed. We compare the performance between alternative learning algorithms and analyze the results of two experiments performed in the smart home.

Keywords. Smart Environments, Machine Learning, Energy Prediction, Feature Selection.

Introduction

The recent emergence of the popularity of Smart Environments is the consequence of a convergence of technologies in machine learning, data mining and pervasive computing. In the smart home environment research, most attention has been directed toward the area of health monitoring and activity recognition [1, 2]. However, an aspect of home life that is often overlooked in this research is energy consumption. Based on recent reports [3], buildings are responsible for at least 40% of energy use in most countries. As an important part of buildings, household consumption of electricity has been growing dramatically. Thus, the need to develop technologies that improve energy efficiency and monitor the energy usage of the devices in household is emerging as a critical research area. The BeAware project [4] makes use of an iPhone application to give users alerts and to provide information on the energy consumption of the entire house. This mobile application can detect the electricity consumption of different devices and notify the user if the devices use more energy than expected. The PowerLine Positioning (PLP) indoor location system [5] is able to localize to sub-room level precision by using fingerprinting of the amplitude of tones produced by two modules installed in extreme locations of the home. Later work of this system [6] records and analyzes electrical noise on the power line caused by the switching of

[1] Corresponding Author：Phd student, The School of Electrical Engineering and Computer Science, Washington State University; Email: cchen@eecs.wsu.edu

significant electrical loads by a single, plug-in module, which can connect to a personal computer, then uses machine learning techniques to identify unique occurrences of switching events by tracking the patterns of electrical noise. The MITes platform [7] monitors the changes of various appliances in current electricity flow for the appliance, such as a switch from on to off by installing the current sensors for each appliance. Other similar work [8] also proposes several approaches to recognize the energy usage of electrical devices by the analysis of power line current. It can detect whether the appliance is used and how it is used.

In our study, we extend smart home research to consider the resident's energy usage, which will be used to support specific daily activities. The primary goal of this paper is not to improve the energy efficiency of the appliances in the household. The purpose of this paper is to validate our hypothesis that energy usage can be predicted based on sensor data that can be collected and generated by the residents in a smart home environment, including people’s activities, overall movement in the space, and frequency of sensor data events. The results of this work can be used to give residents feedback on energy consumption as it relates to various activities, with a possible eventual goal of automating some of the activities in a more energy-efficient manner.

In section 1, we introduce the CASAS smart environment system and describe the representation of the features we select for our experiment and section 2 and 3 present the mRMR feature selection and three machine learning methods with their advantages. Section 4 presents the results of our experiments and compares the performance between different machine learning approaches based on two group experiments.

1. CASAS Smart Environment System

The smart home environment testbed that we are using to recognize activity is a three bedroom apartment located at the university campus.

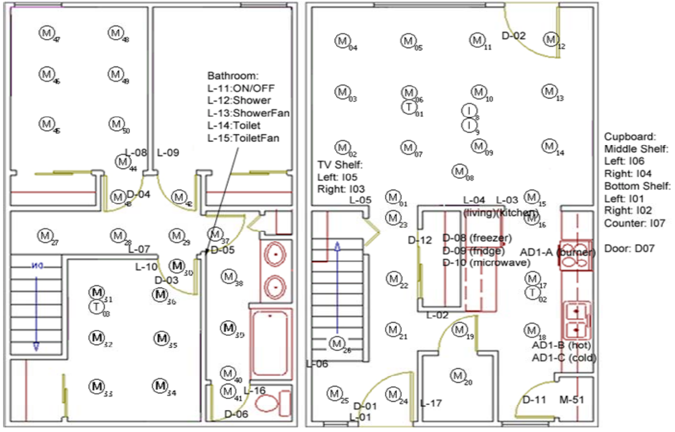

Figure 1. Three-bedroom smart apartment used for our data collection (motion (M), temperature (T), water (W), burner (B), telephone (P),and item (I)).

As shown in Figure 1, the smart apartment testbed has three bedrooms, one bathroom, a kitchen and a living/dining room[9]. In order to track the mobility of the inhabitants, the testbed is equipped with Insteon motion sensors placed on the ceiling. The circles in the figure indicate the positions of motion sensors. They allow tracking of the people moving across the apartment. In addition, the testbed also has installed temperature sensors along with custom-built analog sensors to provide temperature readings and usage of hot water, cold water and stove burner. A power meter records the amount of instantaneous and total power usage. Sensor data is captured using a customized sensor network and is stored in a SQL database. The sensor data gathered by CASAS for our study is expressed by the format shown in Table 1. These four fields (Date, Time, Sensor, ID and Message) are generated by the data collection module automatically.

Table 1. Raw Data from Sensors

Date	Time	Sensor ID	Message
2009-02-06	17:17:36	M45	ON
2009-02-06	17:17:40	M45	OFF
2009-02-06	11:13:26	T004	21.5
2009-02-06	11:18:37	P001	747W
2009-02-09	21:15:28	P001	1.929kWh

To provide real training data, we have collected data while two students in good health were living in the smart apartment. Our training data was gathered during several months and a large amount of sensor events were produced as training data. Each student had a separate bedroom and shared the downstairs living room in the smart apartment. All of our experimental data are produced by these two students' day to day lives, which guarantee that the results of this analysis are real and useful.

To predict energy power usage based on people's activities, we need to annotate sensor data with residents' activities. To improve the quality of the annotated data, we built an open source Python Visualizer, called PyViz, to visualize the sensor events. Figure 2 shows the user interface of PyViz for the CASAS project. PyViz can display events in real-time or in playback mode from a captured file of sensor event readings.

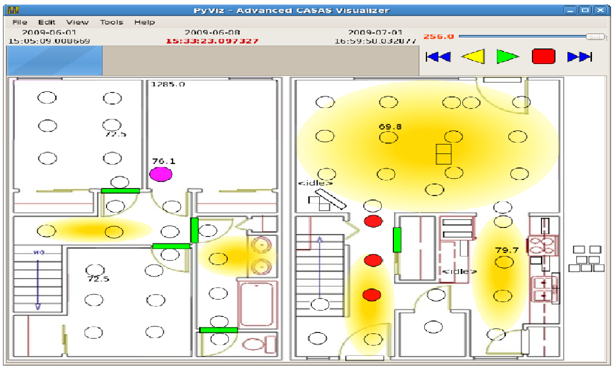

Figure 2. PyViz visualizer for the CASAS project.

With the help of PyViz, activity labels are optionally added at the end of each sensor event to mark the status of the activity. For our experiment, we selected six activities that the two volunteer participants regularly perform in the smart home environment to predict and classify activities. For our experiment, we selected seven activities from a series of activities that two volunteer participants perform in the smart home environment to predict and classify energy use in our smart home project. These activities are as follows:

- Bed to toilet
- Groom

- Breakfast
- Watch TV
- Work at computer
- Sleep
- Prepare meal (lunch,/dinner)

All of the activities that the participants perform have some relationship with the measurable features such as the time of day, the participants' movement patterns throughout the space, and the on/off status of various electrical appliances. We also postulate that the activity will have a measurable relationship with energy usage. For example, when performing a "cook" activity, the participants may open the refrigerator and use the stove or microwave oven, whereas when the participants are sleeping, most appliances will be idle.

Before making use of these learning algorithms, another important step is to extract useful features or attributes from the raw annotated data. We have considered some features that would be helpful in energy prediction. To address the goal of predicting energy, we discretize the energy readings using equal width binning [10], which is widely used in data exploration, preparation, and mining, and has been to preprocess continuous-valued attributes by creating a specified number of bins. In this paper, we discretize the target energy data (kWh) into several intervals (two classes, three classes, four classes and five classes, six classes) to assess the performance of our experiments. We performed two series of experiments. The different experiments are distinguished by the period of time before a power reading during which we collect and analyze the sensor readings. One experiment uses a one-hour span from 00:00 to 23:00. And another experiment makes use of a six-hour span (1 = morning, 2 = afternoon, 3 = evening, 4 = night).

The following is a listing of the features that we used in our energy prediction experiments.

- One-hour Time Window (*00–23*) or Six-hour Time Window (*1–4*)—This feature represents the length of the time in one training instance.
- Times of different activities occurring during the time window (a_1–a_7) — This feature is represent by a 6-tuple: ($a_1, a_2, a_3, a_4, a_5, a_6, a_7$). For example, if activity 1 and 3 both happen twice in the time window, the vector is (2, 0, 2, 0, 0, 0, 0).
- Times of different motion sensors activated during the time window (m_1–m_{51}) — With the same idea as the second feature, this feature is also expressed by 51-tuple:($m_1, m_2, \ldots, m_{51}$). And m_i stands for the times of number *i* motion sensor activated.
- Times of major appliances opened and closed during the time window.
- Total power values (w) of major appliances used during the time window - By analyzing power data (w) from the power meter, we can compute which major device was used because every appliance has a recognizable upward or downward shift in power usage. This attribute records the total amount of power values (w) that are generated by major devices in the time window.

- Target Value: Values (kWh) of Energy — Through the power meter, we can attain the readings (kWh) of energy used in the time window.

2. Feature Selection

After feature extraction, our algorithm will generate a large number of attributes or features to describe a particular activity. However, some of these features are redundant and irrelevant and thus suffer from the problem of a drastic raise in computational complexity and classification errors [11]. One of the most popular feature selection approaches is the Max-Relevance method [12], which selects the features with the highest relevance to the target class c. Using Max-Relevance, features x_i are selected individually to provide the largest mutual information $I(x_i; c)$ with the target class c, reflecting the largest dependency on the target class. The definition of mutual information $I(x_i; c)$ is as following:

$$\mathrm{I(x_i; y)} = \iint \mathrm{P(x_i, y) log \frac{p(x_i,y)}{p(x_i)p(y)} dxdy} \quad (1)$$

However, the results of Cover [13] show that combinations of individually good features do not necessarily lead to good classification performance. To solve this problem, Peng et al. [14] proposed a heuristic minimum-Redundancy-Maximum-Relevance (mRMR) selection framework, which selects features mutually far away from each other that still maintain high relevance to the classification target. This method has proven to be more efficient than Max-Relevance selection. In mRMR, Max-Relevance is used to determine the mean value of all mutual information values between an individual feature $\mathrm{x_i}$ and class c:

$$\max \mathrm{D(S, c)}, \ \mathrm{D} = \frac{1}{|\mathrm{S}|} \sum_{\mathrm{x_i} \in \mathrm{S}} \mathrm{I(x_i; c)} \quad (2)$$

Because the Max-Relevance features may have a very high possibility of redundancy, a minimal redundancy condition can be added to select mutually exclusive features:

$$\min \mathrm{R\,(S)}, \ \mathrm{R} = \frac{1}{|\mathrm{S}|^2} \sum_{\mathrm{x_i, x_j} \in \mathrm{S}} \mathrm{I(x_i; x_j)} \quad (3)$$

Thus, the operator $\Phi(\mathrm{D, R})$ is the mRMR criterion combining the above two constraints:

$$\max \Phi(\mathrm{D, R}), \Phi = \mathrm{D - R} \quad (4)$$

Because its outstanding property that select promising features, we use mRMR feature selection method before applying the machine learning algorithms to the data sets.

3. Prediction Model

The field of machine learning [15] is capable to learn and recognize complex patterns based on sensor data. In our study, we make use of three popular machine learning methods to predict energy usage given the optimal features. we selected: a Bayes Net Classifier, an Artificial Neural Network Classifier, and a LogitBoost ensemble learning method. We later describe experiment results and test these four algorithms on the data collected in the CASAS smart home apartment testbed.

3.1. Bayes Net

Bayes belief networks [16] belong to the family of probabilistic graphical models. They represent a set of conditional independence assumptions by a directed acyclic graph, whose nodes represent random variables and edges represent direct dependence among the variables and are drawn by arrows by the variable name. Unlike the naïve Bayes classifier, which assumes that the values of all the attributes are conditionally independent given the target value, Bayesian belief networks apply conditional independence assumptions only to subsets of the variables. They can be suitable for small and incomplete data sets and they incorporate knowledge from different sources. After the model is built, they can also provide fast responses to queries.

3.2. Artificial Neural Network

Artificial Neural Networks (ANNs) [17] are abstract computational models based on the organizational structure of the human brain. The most common learning method for ANNs, called backpropagation, which performs a gradient descent within the solution's vector space to attempt to minimize the squared error between the network output values and the target values for these outputs. Although there is no guarantee that an ANN will find the global minimum and the learning procedure may be quite slow, ANNs can be applied to problems where the relationships are dynamic or non-linear and capture many kinds of relationships that may be difficult to model by other machine learning methods. In our experiment, we choose Multilayer-Perceptron algorithm based on the backpropagation learning method to recognize activities.

3.3. LogitBoost Ensemble

AdaBoost [18], an abbreviation for Adaptive Boosting, is the most commonly used boosting algorithm. AdaBoost fits an additive logistic regression mode $\mathrm{F(x)} = \sum_{\mathrm{m=1}}^{\mathrm{M}} \mathrm{c_m f_m(x)}$ to the training data. It should minimize the expectation of an exponential loss function $\mathrm{E(e^{-yF(x)})}$. However, the performance of AdaBoost will be harmed by noisy data because the expectation of an exponential loss function changes exponentially. To address this problem, LogitBoost [19] minimizes the expectation of the loss function by using Newton-like steps to fit an additive logistic regression model to directly optimize a binomial log-likelihood $(-log(1 + e^{-2yF(x)})$. The property of LogitBoost turns out to change linearly with the output error and be less sensitive to the noisy data.

4. Experiment Results

We performed two series of experiments. The first experiment uses a one-hour time span and the time window of the second experiment is a six-hour time span (morning, afternoon, evening, night). The testing tool we use, called Weka [20], provides an implementation of learning algorithms that we can easily apply to our own dataset. Using Weka, we assessed the classification performance of our three selected machine learning algorithms using 3-fold cross validation. In the mRMR experiment, we have used the subset size of twenty features for the experiment. Due to space constraints, we just provide the results of the Multilayer-Perceptron algorithm to compare the performance of the two time windows and the effect of the mRMR feature selection.

4.1. Comparison of Average Accuracy

As shown in Figure 3, the highest accuracy is around 90% for both datasets to predict the two-class energy usage and the lowest accuracy is around 60% for the six-class case in both datasets. These results also show that the higher accuracy will be found when the precision is lower because the accuracy of all three methods will drop from about 90% to around 60% with an increase in the number of energy class labels. By testing the different machine learning algorithms, we can get the best algorithm for our energy prediction task. In comparison of the performance between the different algorithms, LogitBoost proves to be the best for both datasets. That is because LogitBoost can generate a powerful classifier by combing the performance of many "weak" classifiers. For the two other algorithms, it is very difficult to judge which one is better than others on the accuracy performance from the experimental results.

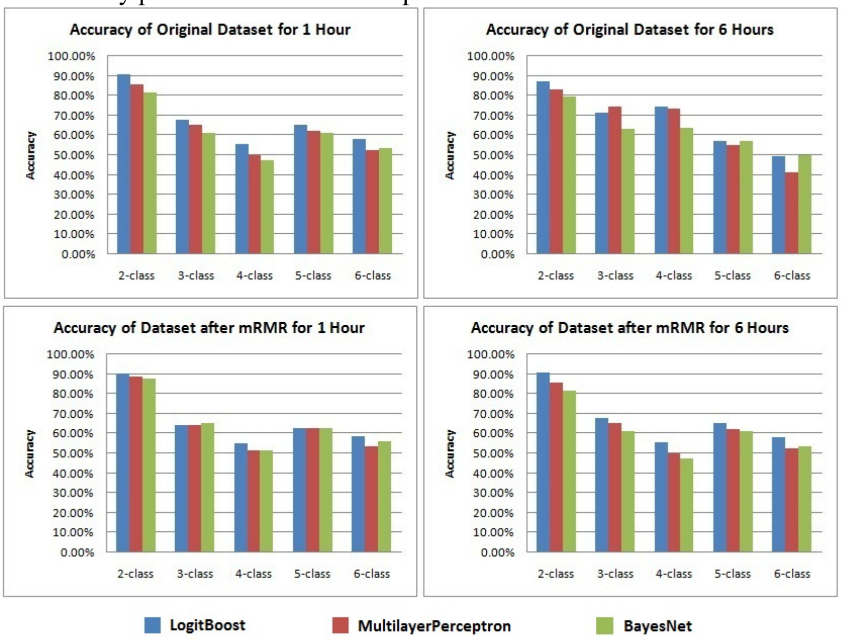

Figure 3. Comparison of accuracies of different algorithms under the following settings: (top-left) original dataset for 1 hour time window; (top-right) original dataset for 6 hours time window; (bottom-left) dataset after applying mRMR for 1 hour time window; (bottom-right) dataset after applying mRMR for 6 hour time window.

4.2. Comparison of Two Time Window Performance

To compare the performance of the one-hour time window and the six-hour time window, we apply the multilayer-perceptron algorithm to these two datasets. As shown in Figure 4, in most situations, the results of the six-hour group have a better performance than the one-hour group. We hypothesize that energy use varies dramatically among the time periods of morning, afternoon, evening and night. For example, at night , most appliance s are idle when the residents go to sleep. Machine learning methods have the ability to capture this kind of difference to build a more accurate prediction model.

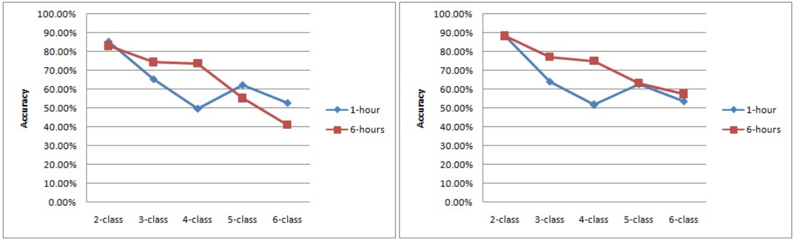

Figure 4. Comparison of accuracy performance between two time windows: (left) original dataset, (right) dataset after mRMR.

4.3. Comparison of Without and With Feature Selection Performance

Figures 5 and 6 show the performance based on a multilayer-perceptron algorithm with and without mRMR feature selection. Figure 6 shows the result of making use of logarithmic time as a scale to compare the different performances. There has been a remarkable improvement in the running time performance of the algorithms after using feature selection. Meanwhile as seen in Figure 5, the classification accuracy is almost the same or a slight better than the performance without feature selection. The function of feature selection can improve the time performance without reducing the accuracy performance in the original data set.

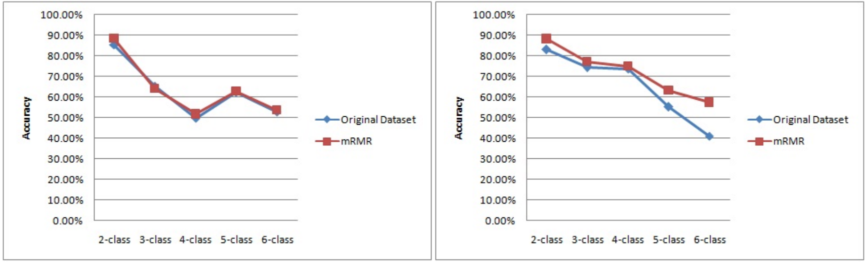

Figure 5. Comparison of accuracy performance without and with mRMR for: (left) one-hour time window, (right) six-hour time window.

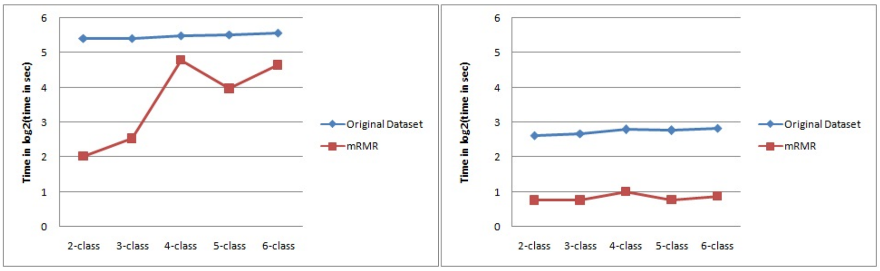

Figure 6. Comparison of running time performance without and with mRMR for: (left) one-hour time window, (right) six-hour time window.

4.4. MRMR Feature Analysis

By applying mRMR feature selection, we select several of the most related features and the least-redundancy features among the candidate features. By taking into consideration various experiment results, the most useful features for classifying energy labels are as follows:

- Time Window
- Activity (1-7)
- Motion (2-5) — Living Room
- Motion (11-13) — Living Room
- Motion17 — Kitchen
- Motion20 — Storage Room
- Motion34 — Bedroom1
- Motion (38-40) — Bathroom1
- Motion(45, 46, 47) — Bathroom2
- Total power values (w) of major appliances used

Inspecting the results from applying the mRMR feature selection method, all the activity features and time windows have been selected by mRMR. The result validates our hypothesis that the energy usage of the resident has a strong relationship with their regular activities. mRMR also selects some specific motion sensor features for each room. Thus, we note that the machine learning algorithms can track the people's activities in the different rooms. Because the major appliances consume a lot of energy, there is a linear relationship with energy usage in the whole smart home and energy major appliances consume. That is why total power values used by major appliances used is also an important feature selected by mRMR.

Analyzing these results, we see that machine learning methods can be used as a tool to predict energy usage in smart home environments based on the human's activity and mobility. However, the accuracy of these methods is not as high as we anticipated when the energy data has been divided into more than three classes. There are several reasons that lead to low performance of these algorithms. One reason is that some of the major devices are difficult to monitor and predict, such as the floor heater, which may rely on the outside temperature of the house. Another reason is that there is not an obvious cycle of people's activities. An additional factor we can't ignore is that there is some noise and perturbation motion when the sensors record data and transfer them into the database. Finally, the sensor data we collect is not enough to predict energy precisely. As a result, we will collect more kinds of sensor data to improve the performance of the prediction.

5. Conclusions

In this work we introduced CASAS, an integrated system of collecting sensor data and applied machine learning in the smart home environment. To predict energy precisely, we extracted features from real sensor data in a smart home environment and used an equal width binning method to discretize the value of the features, then utilized mRMR feature selection to choose the most important features. To assess the performance of the three machine learning methods, we performed two group experiments based two time window, analyzed the results of the experiments and provided the explanation of those results.

In our ongoing work, we plan to further investigate new and pertinent features to predict the energy more accurately. To improve the accuracy of energy prediction, we intend to install more sensitive sensors to capture more useful information in the smart home environment. We are also planning to apply different machine learning methods to different environments in which different residents perform similar activities. This will allow us to analyze whether the same pattern exists across residents and environments. In our next step we will analyze the energy usage data itself to find trends and cycles in the data viewed as a time series.

References

[1] G. Singla, et al., Recognizing independent and joint activities among multiple residents in smart environments, *Journal of Ambient Intelligence and Humanized Computing*, 2010, pp 1–7.

[2] E. Kim, et al., Human Activity Recognition and Pattern Discovery, *IEEE Pervasive Computing* **9**(2010), pp 48–53.

[3] *Energy Efficiency in Buildings*, www.wbcsd.org, 2009.

[4] *Beware*, www.energyawareness.eu/beaware, 2010.

[5] S. N. Patel, et al., PowerLine Positioning: A Practical Sub-Room-Level Indoor Location System for Domestic Use, in *UbiComp 2006: Ubiquitous Computing*, 2006, pp 441-458.

[6] S. N. Patel, et al., At the flick of a switch: Detecting and classifying unique electrical events on the residential power line, in *UbiComp 2007: ubiquitous computing*, 2007, pp 271.

[7] E. Tapia, et al., The design of a portable kit of wireless sensors for naturalistic data collection, *Pervasive Computing*, 2006, pp 117–134.

[8] G. Bauer, et al., Recognizing the Use-Mode of Kitchen Appliances from Their Current Consumption, *Smart Sensing and Context*, 2009, pp 163–176.

[9] D. J. Cook and M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, *Methods of information in medicine*, **48**(2009), 480.

[10] H. Liu, et al., Discretization: An enabling technique, *Data Mining and Knowledge Discovery*, **6**(2002), 393–423.

[11] R. E. Bellman, *Adaptive control processes - A guided tour,* Princeton University Press, Princeton, New Jersey, 1961.

[12] M. A. Hall, *Correlation-based feature selection for machine learning*, Citeseer, 1999.

[13] T. M. Cover, The best two independent measurements are not the two best, *IEEE Transactions on Systems, Man, and Cybernetics*, **4**(1974), 116-117.

[14] H. Peng, et al., Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(2005), 1226-1238.

[15] T. Mitchell, *Machine Learning*, AMcGraw Hill, New York, 1997.

[16] I. Rish, An empirical study of the naive Bayes classifier, in *IJCAI-01 workshop on "Empirical Methods in AI"*, 2001.

[17] S. F. Zornetzer, *An introduction to neural and electronic networks: Morgan Kaufmann*, 1995.

[18] Y. Freund and R. E. Schapire, Experiments with a New Boosting Algorithm, in *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp 148-156.

[19] J. Friedman, et al., Additive Logistic Regression: a Statistical View of Boosting, *Annals of Statistics*, **28**(1998), pp 2000.

[20] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations: Morgan Kaufmann*, 1999.