Machine Learning and Dataming Algorithms for Predicting Accidental Small Forest Fires

Vasanth Iyer*, S. Sitharama Iyengar†, N. Paramesh §, Garmiela Rama Murthy*, Mandalika B. Srinivas‡
*International Institute of Information Technology, Hyderabad, India - 500 032
†Louisiana State University, Baton Rouge, LA 70803, USA
§University of New South Wales, Sidney, Australia
‡Brila Institute of Technology & Science, Hyderabad Campus, Hyderabad-500078, India

vasanth,rammurthy{@research.iiit.ac.in},iyengar@csc.lsu.edu,paramesh@cse.unsw.edu.au,srinivas@bits-hyderabad.ac.in

Abstract—**Extracting useful temporal and spatial patterns from sensor data has been seen before, the technical basis of Machine learning with Data mining is studied with the evidence collected uniformly over many years and which allow using users' perspective in collected evidence. This model helps in probabilistically forecasting fires and help forest department in planing day to day schedules. Using a model to predict future events reliably one needs to collect samples from sensors and select a feature, which does have any particular bias. Due to practicable problems most of the collected data have 80% of attributes missing and the remaining has numeric values, which are hard to discretization. To adapt to such limitations, we use nominal data type, which allows better understanding of the temporal and spatial features, which are learnt. We encounter several practicable limitations as forest fires events are very rare and manual classification is extremely costly. Another is the unbalanced nature of the problem of the many forest fire events many are of the burnt area is very small and gives skewed distribution. Most of the examples naturally group into batches, which are collected from evidence satellite photography and collaborative reports from national parks departments. The second set of database was collected from the meteorological weather station about several weather observations, which are located very close to the reported fires. Finally, the compiling task is to serve as a filter and provide the user to vary the false alarm rate. We show by regression analysis of the compiled dataset that the forest fire classifier has a minimum false alarm rate when including temporal features. The machine learning algorithms successfully classifies accidental small fires with 85% reliably and large fires by a much lower accuracy of 30%.**

Index Terms—**Machine Learning, Datamining, Naive Bayes, Forest fires, Fire Weather Index (FWI), Temporal Patterns, WEKA machine learning framework.**

I. INTRODUCTION

Accidental small forest fire can lead to heavy loss of precious natural reserves in protected lands in which many different species thrive due to their balanced ecology. Tropical rain forests are also a factor in keeping the sensitive balance global warming trends seen recently due to heavy deforestation due to human needs. One of objective of Datamining is to allow modeling the users' perspective such as temporal properties, which are cause and effect of forest fires, which allows in reducing false detection. The hidden patterns are mined, which allows to find the underlying hidden structure of the data. This allows learning the concepts needed for forest fires classification. The features extracted of the predicted class by means of datamining allows to apply many machine learning algorithms to the transformed data. This framework forms the technical basis for the supervised and unsupervised classification.

Temporal query properties like weekday and weekends help probabilistically bias the predicted outcome of class variables to be classified. As a small human accidental fire or a possibility of occurrence of large natural fire disasters can further be classified according to the users choice. Attribute value transformations are equally important when formulating attribute dependencies within a weather class [4,5,6,7] nominal values such as cool, windy and high humidity for successful formulation of machine learning rules.

Following motivation the rest of the paper with is organized as follows. In Section III, we evaluate the performance of machine learning algorithms and develop a weak learner for temporal features. Section IV presents initial basis for user queries without significant error analysis, that is without any ranking criteria. In Section V, we evaluate Naive Bayes [1] with Tree based classifiers and compare the method on the task of accidental fire prediction. Section VI investigates alternative feature and computational aspects of the method respectively and explains the results. Section VII concludes the paper.

II. STATE OF THE ART

The historic information recorded by it does not reveal any hidden patterns to calculate the likelihood of forest fires. Classifiers model depends on accurate class conditional probabilities but in practice, samples are limited,

 ISBN: 978-1-61208-144-1

most of the estimates are approximate which further biases the sample search space. With limited samples, the Bayesian method does the better estimation of the class conditional probabilities, when compared to maximum likelihood method. The internal representation of the classifier data model uses a weighted term, and best evaluates the quantity purity of the labeled class. This provides classification of the same-labeled pattern with insufficient samples, with pure and impure groups and helps its internal ranking. The internal representation captures the hidden pattern of the training samples, once the hidden patterns are quantitatively verified with a base classifier such as Nave Bayes, these representative patterns are further classified by user specified attributes such as which month and which day the particular pattern has maximized the likelihood of a phenomenon such as fire event.

There are standard benchmarks for performance comparison of classifiers and Bayes gives the lowest error rate compared to others. We also study the kappa score, which compares our classifier with a J48 tree classifier for the same input data set and normalizes using the results of the confusion matrixes. A high kappa score is generally preferred for a classifier to be efficient, which needs using of good pre-processing algorithms. Since sensor data are, highly unreliable most of well-designed classifiers perform badly and cannot adapt to the sensor data stream. By using post-processing of miss-classified samples and identifying falsely classified data also called outliers, we further improve the reliability. From authors previous work [1], we have shown data aggregation eliminate redundancies and improves reliability in sensor network performance. The current ML algorithm focuses on event aggregation over a long period of time from user reports and collaborative sensor network stream, which have been further classified to a particular application. We study the effects of predicting forest fires in a given region using sensor aggregated data.

III. MACHINE LEARNING RULES

Consider the concept leaning, in particular the learner considers some finite hypothesis [6] space H defined over instance space X, in which the task is to learn some target concepts $c : X- > 0,1$. As we are building a fire event predictor from the sensed data, we assume that the network learner is given some sequence of training measurements $((X_1, d_1)...(x_m, d_m))$ where x_i is some instance from X and where d_i is the target value of $d_i = c(x_i)$. As we are learning from a knowledge base such as data repository the sequence of instances $(x_1...x_m$ is held fixed, so that the training data D can be written as the sequence of target values $D = d_1...d_m$.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{1}$$

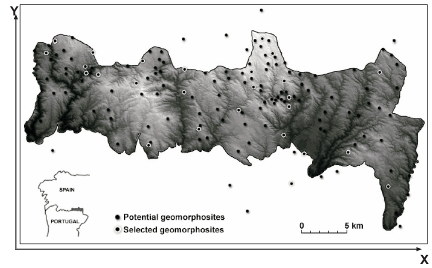

Fig. 1. location of inventorised geomorphosites in the montesinho natural park.

$$h_{MAP} = argmax_{h\epsilon H} P(h|D) \tag{2}$$

The assumptions for the concept learning follows:

- The training data D is noise free.
- The target concept c is contained in the hypothesis space H.
- We have no a priori reason to believe that any hypothesis is more probable than any others.

Since we assume noise free samples, the first hypothesis can be formed for the detection of forest fires [2] for an approximate target function V as shown in equation (3)

$$V(Fire\,Location) = XPos + YPos \tag{3}$$

where every incident of forest fire is documented and its location in terms of `<X, Y>` [2] are recorded. The learning algorithm uses a boosting [5,6] method to learn from the forest fire events and its corresponding correlated sensor measurements. The model does not use real-time sensor inputs and data samples to classify but on the other hand it uses recorded fire events and probabilistically predict the new sensor input closest to the already seen training sample (data aggregated over time) using Naive Bayes or Tree classifiers. This computational model can further post-processed using supervised learning to improve on the purity of the classes and detect any outliers which may create false alarms. The unreliability of accurately detection real and outlier events is an open problem in sensor networks. The above equation is dependent on `<X, Y>` positions in Figure 1 map, for grid of 10x10 it may need 100 combinations of every other dependent variable making the model unfriendly. The estimated formula of the above equation (3) can estimated in temporal terms for the Fire Event as shown in equation (4).

$$\begin{aligned} &\mathrm{V_{train}}(\mathrm{FireEvent_{Day\,of\,week}}) \leftarrow \\ &\hat{V}(FireEvent_{Day\,of\,week})\textit{Temporal Variable} + \\ &\hat{V}(FireEvent_{Day\,of\,week})\textit{Correlated measurements} \end{aligned} \tag{4}$$

$$\begin{aligned} \text{Temporal Variables} = \;&\textit{Month of the year} + \\ &\textit{Day of the week} \end{aligned} \tag{5}$$

 ISBN: 978-1-61208-144-1

$$\text{Correlated measurement} = \textit{temperature} + \textit{humidity} + \textit{wind} + \textit{rain} \tag{6}$$

$$\text{Classfires} = \{\text{accidental; small, medium, large}\} \tag{7}$$

A. Estimating training values with sample data

Sample datasets are based on UCI forest fire repository. The equation representing the Bayes probability model of the hypothesis is given in equation (1). In our case the hypothesis to be maximized is shows in equation (2) for a four class classification, as shown in equation (7). The assumption here is that the training set D is a unbiased representation to learn the concept c and can estimate the inputs x_i. The previously defined dependent variable **Fire Location**, which is used to estimate given the independent correlated measurements and its relation to the temporal attributed are given in equations (4), (5) and (6). The target concepts are present in the training samples and, we like to see the influence of adding sensor measurements to further accurately learn the concepts of the human induced accidental fires versus the more natural accruing types of the medium and large fires. For the sake of clarity of machine learning domain we convert the correlated sensor data to nominal [5] types, as illustrated below.

$$temperature = \{cool; mild; hot\} \tag{8}$$

$$humidity = \{low; medium; high\} \tag{9}$$

$$wind = \{true; false\} \tag{10}$$

The model estimation of the the target function with weights w_1, w_2 as shown allows to minimize the training error, where x_1, x_2 are temporal and correlated measurements.

$$\hat{V} = w_1 x_1 + w_2 x_2 \tag{11}$$

The learning algorithm needs to define the best fit for the given hypothesis and adjust the weights to minimizing the error and miss classifications.

$$E \equiv \sum (V_{train}(FireEvent) - \hat{V}(FireEvent))^2 \tag{12}$$

B. Algorithm complexity

Search space consists of all the possible patterns of the features, given our data model, $3*3*2*4 = 72$ possibilities for each rule when using attributes 3 for temperature, 3 for humidity, 2 for wind for 4 classes of fire categories. As there are 517 rules from the collected dataset instances the complete search space [5] will have ${}^nP_r = 72^{522} \approx 10^{969}$ different possibilities. To minimize the complexity of search space, we can further cut down on the sample instances by using spatial clustering and removing any redundancies in similar features. Given the `<X,Y>` positions, we can cluster into groups the possible fires types into accidental small fires and others which have medium and larger burnt area as large fires.

FIRE TYPES	RECORDED
Accidental (AF)	247
Small (SF)	175
Medium (MF)	71
Large (LF)	24

TABLE II
TARGET VARIABLE OCCURRENCES.

As measuring ambient phenomena are correlated, we expect them to be independent, then all the i.i.d's can be aggregated to ${}^nC_r = {}^{517}C_{72} \approx 10^{89}$ possibilities, were in this case the combination is calculated. As these methods are used with pre-processing to reduce data overloads in the model further real valued dataset search space optimization is possible. Domain knowledge as in the case of WSN can be used practically to reduce the complexity of machine learning algorithms if well studies and calibrated. To judge the affectiveness of the model and the classification effectiveness, we initially rely on real-valued numeric model such as [1] to estimate the errors. In contrast to the previous approach, we use nominal values as defined in equations (8), (9) and (10) to build a tree classifier and further reduce errors.

IV. NAIVE BAYES

One can use Naive Bayes [5], which by design presumes the class densities, which have been determined and accurate. The model calculates the class conditional probabilities of the input feature vectors. To understand the underlying skewed structure of the dataset, we further create thresholds for accidental small fires compared to medium and large fires as shown in Table II. So we have the four possible values for the target variable as shown in equation (7).

A. User query

To validate the model let us predict the outcome of a peak month, from the dataset [2] August has significant number of reported fires compared to other months. Estimating the probabilities of fire events given the attribute values for the class

$$? = \{Month = August; Day = Monday\}$$
$$\{Temprature = Cool; Humidity = High; Wind = True\}$$

The estimated class conditional densities for the independent variables temperature, humidity and wind conditions are calculated using temporal attributes **month** for the dataset are shown in Table II. The datasets further is explored using two temporal variables, which are **month** and the **day of the week** as shown in Table I and Table IV. The temporal variables introduced into the dataset helps gain the insight of users' dependencies with fire prediction model.

$$g_i(x) = P(\omega_i \| \mathbf{x}) = \frac{p(\mathbf{x} \| \omega_i) P(\omega_i)}{\sum_{i=0}^{i=4} p(x \| \omega_j) P(\omega_j)} \tag{13}$$

 ISBN: 978-1-61208-144-1

Burnt Area(hectors)	AUG	MON	TEMP	HUMIDITY	WINDY	PRIOR PROB	PREDICTOR VAR
GT 1h	▶ 0.34	▶ 0.14	▶ 0.46	▶ 0.17	▶ 0.42	▶ 0.47	▶ 57%
GT 1h LEQ 10h	0.39	0.15	0.35	0.13	0.38	0.33	25.0%
GT 10h LEQ 50h	0.30	0.14	0.43	0.19	0.50	0.13	17.0%
GT 50h	0.33	0.08	0.16	0.08	0.37	0.04	0.02%

TABLE I
POSTERIORS PROBABILITIES FOR BACKGROUND WEATHER DATA FOR THE PEAK MONTH AUGUST.

FIRE TYPE	MONTH=AUG
Accidental	0.004
Small	0.002
Medium	0.001
Large	0.00004

TABLE III
LIKELIHOOD OF FIRES FOR THE MONTH OF AUGUST.

DAYS	ACCIDENTAL	SMALL	MEDIUM	LARGE
MON	▶ 35	▶ 27	▶ 10	▶ 2
TUE	28	21	11	4
WED	22	24	5	3
THU	30	21	9	1
FRI	42	31	12	0
SAT	42	24	11	7
SUN	48	27	13	7
TOTAL	▶ 247	▶ 175	▶ 71	▶ 24

TABLE IV
POSTERIORS PROBABILITIES FOR TEMPORAL FEATURE DAY OF THE WEEK.

Substituting the corresponding highlighted values from Table 1 through to Table IV in the above equation (13), we get the posterior probability of accidental small fire

$$\hat{fire_{accidental}} = \frac{0.0007547}{0.003565} = 57\% \tag{14}$$

$$\hat{fire_{Small}} = \frac{0.000333}{0.003565} = 25\% \tag{15}$$

$$\hat{fire_{medium}} = \frac{0.000223}{0.003565} = 0.17\% \tag{16}$$

$$\hat{fire_{large}} = \frac{0.000000287}{0.003565} = 0.02\% \tag{17}$$

The posterior probabilities for the month of August for the data collected in Portugal [2], the likelihood of accidental small fires are very high. From cross-validating from the known fact that in summer the likelihood of wild fires are higher the Bayes rule is able to classify the dataset for accidental and small fires with high accuracy. We use a simulation framework in the next sections to further prove our intial conclusion from the datasets, it is shown that the training time for Naive Bayes scales linearly in both the number of instances and number of attributes.

V. TREE CLASSIFIER

In this section, we will focus on the domain rules, which are applicable to the learning system. Tree classifiers lend itself to use ML rules [6] when searching the hypothesis by further branching on specific attributes. The design of such a classifier needs to sort the weights or entropies [5] of the attributes, which is the basis of its classification effectiveness.

ID3 is a popular tree classifier algorithm, to implement ID3 as illustrated in Figure 2 and Table X with our attributes. Let (S) be a collection of samples then using the tree algorithm, which uses entropy to split its levels is given by

$$Entropy(S) = \sum_{i=0}^{i=c} p(i) \log_2 p(i) \tag{18}$$

Let us assume a collection (S) has 517 samples [2] with 248, 246, 11 and 12 of $accidental, small, medium, large$ fires respectively. The total entropy calculated from equation (18) is given by

$$\begin{aligned} Entropy(S) = {} & \frac{248}{517} \log_2 \frac{248}{517} + \\ & \frac{246}{517} \log_2 \frac{246}{517} + \\ & \frac{11}{517} \log_2 \frac{11}{517} + \\ & \frac{12}{517} \log_2 \frac{12}{517} = \\ & 1.23 \end{aligned}$$

A. Attribute selection

ID3 uses a statistical property called information gain to select the best attribute. The gain measures how well the attribute separates training targeted examples, when classifying them into fire events. The measure of purity that we will use is called information and is measured in units called bits. It represents the expected amount of information that would be needed to specify whether a new instance should be classified accidental, small, medium or large fires, given that the example reached that node. The gain of an attribute is defined by and illustrated in Table V. Using the calculated attribute for information gain we show that **temp** attribute is used before the **wind** attribute to split the tree after the tree root.

$$Gain(S, A) = Entropy(S) - \sum_{i=0}^{i=c} \frac{S_v}{|S|} Entropy(S_v) \tag{19}$$

$$\begin{aligned} Entropy(S_{Hot}) = {} & \frac{9}{36} \log_2 \frac{9}{36} + \\ & \frac{23}{36} \log_2 \frac{23}{36} + \\ & \frac{3}{36} \log_2 \frac{3}{36} + \\ & \frac{1}{36} \log_2 \frac{1}{36} = \end{aligned}$$

$$1.282$$

$$Entropy(S_{Medium}) = \frac{23}{96}\log_2\frac{23}{96} + \frac{65}{96}\log_2\frac{65}{96} + \frac{3}{96}\log_2\frac{3}{96} + \frac{5}{96}\log_2\frac{5}{96} = 1.175$$

$$Entropy(S_{Cool}) = \frac{117}{269}\log_2\frac{117}{269} + \frac{146}{269}\log_2\frac{146}{269} + \frac{2}{269}\log_2\frac{2}{269} + \frac{4}{269}\log_2\frac{4}{269} = 1.05$$

$$Entropy(temp) = \frac{43}{517} * 1.282 + \frac{139}{517} * 1.175 + \frac{335}{517} * 1.05 = 1.08$$

$$Gain(S, temp) = 1.23 - 1.08 = 0.192$$

$$Entropy(S_{HIGH}) = \frac{162}{249}\log_2\frac{162}{249} + \frac{72}{249}\log_2\frac{72}{249} + \frac{8}{249}\log_2\frac{8}{249} + \frac{7}{249}\log_2\frac{7}{249} = 1.1952$$

$$Entropy(S_{LOW}) = \frac{68}{133}\log_2\frac{68}{133} + \frac{59}{133}\log_2\frac{59}{133} + \frac{2}{133}\log_2\frac{2}{133} + \frac{4}{133}\log_2\frac{4}{133} = 1.24$$

$$Entropy(wind) = \frac{361}{517} * 1.1952 +$$

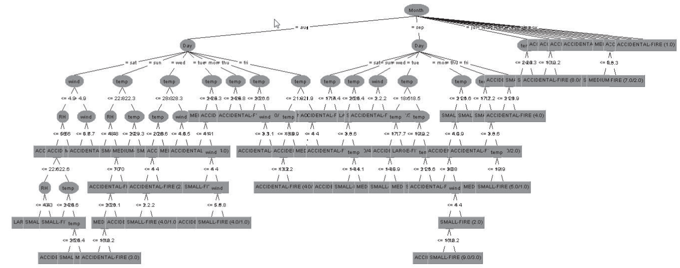

Fig. 2. Tree classifier and attribute view.

Month	Temp	Wind
Not shown	info: 1.08	info: 1.20
Not shown	gain: 1.23-1.08 = 0.192	gain: 1.23-1.08 = 0.025

TABLE V
GAIN RATIO CALCULATION FOR TREE IN FIGURE 2.

$$\frac{156}{517} * 1.24 = 1.20$$

$$Gain(S, wind) = 1.23 - 1.20 = 0.025$$

The internal tree representation for m attributes from n samples will have a complexity of $O(\lg n)$, with increasing inputs, given by parameter n, the height of the tree will not grow linearly as in the case of Naive Bayes. On the other hand complexity of building a tree will be $O(mn \lg n)$

VI. SIMULATION

Open-source workbench called WEKA [3] is a useful tool to quantify and validate results, which can be duplicated. WEKA can handle numeric attributes well, so we use the same values for the weather data from the UCI [4] repository datasets. The class variable has to be a nominal one, to allow WEKA [3], we convert all fire types to ″0″ or ″1″. Where ″0″ is of accidental small fire and ″1″ is for large fires making it a two class classifier, the results are shown as confusion matrix in Table VIII and Table IX. Naive Bayes correctly classifies accidental and small fires(209 out of 247) were as the J48 Tree classifier does far more, 219 out of 247. As WEKA uses kappa [3] stats for evaluating the training sets, a standard score of $> 60\%$ means training set is correlated, using J48 simulation, we get 53.56% just below the standard. The comparison on results shows that tree classifier does better than Naive Bayes by 25% overall and equally well for accidental and small fires as shown in Table VI and Table VII, when randomly tested it falls just short of the expected 60%. Therefore using sensor network measurements accidental and small fires can be predicted reliably.

WEKA Stats	Results	Summary
Correctly Classified Instances	267	51.64%
Incorrectly Classified Instances	250	48.35%
Kappa statistic	0.1371	
Mean absolute error	0.3022	
Root mean squared error	0.3902	
Relative absolute error	94.86%	
Root relative squared error	97.84%	
Total Number of Instances	517	

TABLE VI
EVALUATION ON TRAINING SET FOR NAIVE BAYES.

WEKA Stats	Results	Summary
Correctly Classified Instances	373	72.14%
Incorrectly Classified Instances	144	27.85%
Kappa statistic	0.5356	
Mean absolute error	0.1938	
Root mean squared error	0.3113	
Relative absolute error	60.83%	
Root relative squared error	78.04%	
Total Number of Instances	517	

TABLE VII
EVALUATION ON TRAINING SET FOR J48 TREE CLASSIFIER.

A. Simulation analysis

WEKA attribute statistics and its effective correlation score. Table VI and Table VII show kappa and other comparison statistics for Naive Bayes and J48 tree classifier.

B. Error analysis

Equation (12) specifies the model error and the Confusion matrix from the simulation score are shown in Table VIII and Table IX, upper bound of small fire(AF+SF) has over 80% accuracy for J48-Tree and 61% for Naive Bayes. The corresponding baseline performances including all fires categories is 72.1% for J48-Tree and Naive Bayes is 51.64%, which is due to large fires not correlated.

1) Correlation of attributes: From statistical point of view if the attributes have similar values then it creates high bias creating what is called over-fitting error during learning. In our case **temp** and **humidly** may have similar values and needs to be avoided and substituted with a suitable attribute. To pre-process and analyze, we use all the available in the dataset and WEKA provides the attribute selection as illustrated in Table X.

We use the attribute selection wizard of WEKA to find out the best match. The analysis shows from Table X that

	LF	MF	SF	AF
LF	0	1	7	16
MF	0	5	12	54
SF	0	7	53	115
AF	0	0	38	209

TABLE VIII
CONFUSION MATRIX FOR NAIVE BAYES USING TRAINING SET.

	LF	MF	SF	AF
LF	7	0	7	10
MF	0	29	15	27
SF	1	7	118	49
AF	0	5	23	219

TABLE IX
CONFUSION MATRIX ON TRAINING SET FOR J48 TREE CLASSIFIER.

Number of folds (%)	No.	Attribute
10(100 %)	1	month
1(10 %)	2	day
0(0 %)	3	temp
0(0 %)	4	RH
0(0 %)	5	wind

TABLE X
ATTRIBUTE SELECTION 10 FOLD CROSS-VALIDATION (STRATIFIED)

the Month(100%), Day(10%) and Wind(0%) are highly dependent on the precision. As most of the attributes are nominal it lends more to a tree classifier, which are more flexibility in handling nominal types by design.

VII. CONCLUSION AND FUTURE WORK

The future research work will focus on how to rank sensor queries with high reliability which otherwise be biased due to unverifiable outliers present in the form of noise, spikes and false positives in the time-series data. The training sample sorting allows to weigh the precession versus relevant evidence based on the ranking criteria, such has F-scores and correlated Fire Weather Index (FWI) to further compare the likelihood of predicting large fire events reliably. The statistical analysis of the data collection helps in exploring the higher and lower bounds of the FWI ranges and its corresponding robustness to predict large fires using our implemented algorithms.

VIII. ACKNOWLEDGEMENT

One of the authors like to thank Shailesh Kumar of Google, Labs, India for suggesting the machine learning framework WEKA and UCI for providing the most needed datasets on forest fires. Dean P.J. Narayanan has been a mentor in introducing dimension reduction and attribute selection. The first author like to express appreciation and support from Dr. S.S Iyengar for funding this work under his LSU research grant. Authors also like to thank the anonymous reviewer's comments which has improved the final quality of the submission.

REFERENCES

[1] Vasanth Iyer, S.S. Iyengar, G. Rama Murthy, and M.B. Srinivas. INSPIRE-DB: Intelligent Networks Sensor Processing of Information using Resilient Encoded-Hash DataBase. SENSORCOMM 2010, The Fourth International Conference on Sensor Technologies and Applications, Venice, pp. 363-368.

[2] Paulo Cortez and Anibal Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. Department of Information Systems-R&D Algoritmi Centre, University of Minho, Portugal.

[3] WEKA Machine learning software. $http : //www.cs.waikato.ac.nz/\ ml/weka$ [Accessed May 15th, 2011].

[4] Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. [Accessed May 20th, 2011].

[5] Ian H. Witten and Eibe Frank. Datamining, Pratical machine learning. Elsevier 2005.

[6] Tom M. Mitchell. Machine Learning. MaGRAW-Hill Publications 1997.