FACTA UNIVERSITATIS
Series: **Working and Living Environmental Protection** Vol. 10, N° 1, 2013, pp. 37 - 51

MINING AND PREDICTING RATE-OF-RISE HEAT DETECTOR DATA

UDC 519.2:004.8:681.536.5

Mirjana Maksimović[1], Vladimir Vujović[1], Vladimir Milošević[2]

[1]Faculty of Electrical Engineering, East Sarajevo, Bosnia and Herzegovina
[2]Faculty of Technical Sciences, Novi Sad, Serbia

Abstract. *Extreme events like fire can cause massive damage to indoor areas and life threatening conditions. Early residential fire detection is important for life preservation, prompt extinguishing and reducing damage. To detect fire, one or a combination of sensors (heat detectors, smoke detectors, flame detectors) and a detection algorithm are needed. The sensors might be a part of a wireless sensor network (WSN) or work independently. One of the most frequently used heat detectors is the rate-of-rise heat detector. In this paper some of the data mining algorithms on simulation data of the rate-of-rise heat detector are applied. Data mining seems to be an effective technique for discovering useful knowledge from a large amount of data observed by many sensors. Prediction in sensor networks can be performed in the way that each sensor learns a local predictive model for the global target classes, using only its local input data. Only the predicted target class for each reading is then transmitted to the gateway or to the base station. One important class of such algorithms are predictors, which use the sensor inputs to predict some output function of interest. The purpose of the paper is to analyze different classification algorithms in the case of rate-of-rise heat detector to see which of the applied techniques led to higher accuracy and fewer errors.*

Key words: *Data Mining, J48, Naïve Bayes, Neural Network, Rate-of-rise Heat detector, SVM*

1. INTRODUCTION

Extreme events such as fire can cause massive damage to indoor areas and life threatening conditions. Early residential fire detection is important for prompt extinguishing and reducing damages and loss of life. To detect fire, one or a combination of sensors and a detection algorithm are needed. The sensors might be part of a wireless sensor network (WSN) or work independently [1]. The temperature sensors are probably the simplest and the most obvious sensors for fire detection. Heat detectors generally fall into two categories - fixed temperature heat detectors and more commonly, rate-of-rise heat detectors. A

Received June 11, 2013

fixed temperature heat detector utilizes a temperature sensing element which generates an alarm condition if the temperature within the protected area reaches a predetermined level (e.g. 60ºC or 90ºC). A fixed temperature trigger point should be selected which is most suitable for the situation in question. These detectors are used where high ambient temperatures exist or where sudden changes in temperature can occur e.g. kitchens, boiler rooms & foundries etc. A rate-of-rise heat detector includes a fixed temperature element as the ones mentioned above but in addition includes a temperature sensing element which can detect a sudden change in temperature. This is a device that responds when the temperature rises at a rate exceeding a predetermined value (e.g. 8.33 ºC/min, 9 ºC/min or 11 ºC/min - according to NFPA 72 standard [2]). This type of detector is more sensitive than a simple fixed temperature heat detector and as such is the choice for applications in which reliable performance and early warning are critical, but where the environment makes smoke detection unsuitable. In this paper simulated data of rate-of-rise heat detectors are considered with the aim of, by applying some of the data mining techniques, realizing which of them generate the best predictive model.

Sensor data mining is a relatively new area, which is now reaching a certain level of maturity. Sensor data brings numerous challenges with it in the context of data collection, storage and processing. Data mining is an iterative process of extracting hidden patterns from large data sets. A variety of data mining methods such as clustering, classification, frequent pattern mining, and outlier detection are often applied to sensor data in order to extract actionable insights. This data usually needs to be compressed and filtered for more effective mining and analysis. The main challenge is that conventional mining algorithms are often not designed for real time data processing. Therefore, new algorithms for sensor data stream processing need to perform the analytics in a single pass in real time. In addition, the sensor scenario may often require in-network processing, wherein the data is processed to higher level representations before further processing. This reduces the transmission costs, and the data overload from a storage perspective [3]. However, the simple application of the data mining technique to sensor data may not be as successful as expected because sensor data are mostly mere numerical values. Thus, contextual data should be incorporated in the database for data mining as well as sensor data [4]. In addition, several algorithms must be applied to the application before a suitable algorithm for the selected data types can be found. Therefore, the selection of a correct data mining algorithm depends not only on the goal of an application, but also on the compatibility of the data set.

The aim of this paper is to apply some of data mining techniques on rate-of-rise heat detector simulated data in order to see which of the chosen algorithms has the best classification and predictive accuracy and is the most appropriate for a particular case.

The rest of this paper is organized as follows. The simulation of rate-of-rise sensor data starting from the fire ignition is given in Section 2. The analysis of data mining techniques and the experimental results are given in Section 3 and Section 4, respectively. Finally, Section 5 gives the conclusion.

2. THE FIRE DETECTION MODEL

The model used for the simulation is based on a set of functional relations for the temperature and velocity of fire gases in a ceiling jet proposed in [5] and on NFPA 72

standards [2]. Specifically, the heat release rate $Q[kW]$ of the fire is assumed to grow with time $t[s]$ according to a power-law relationship as follows:

$$Q = \alpha t^2 \tag{1}$$

where α is a constant for a particular fuel. To characterize the difference between fuels, the concept of "critical time" $t_c[s]$, is introduced to describe the fire intensity. Time t_c is defined as the time at which a power-law fire would reach a heat release rate of 1055 $[kW]$ [6]. In terms of t_c, Equation (1) becomes

$$t_c = \sqrt{\frac{1055}{\alpha}} \quad \text{or} \quad \alpha = \frac{1055}{t_c^2}[(Btu/s)/s^2 \text{ or } kW/s^2] \tag{2}$$

Fires are classified as being either slow-, medium-, or fast-developing from the time that established burning occurs until the fire reaches a heat release rate of 1000 [Btu/s] (1055 [kW]). Figure 1 and Table 1 present the power-law heat release rate.

Table 1 Power-law heat release rate

Fire Growth Rate	Growth Time $t_g[s]$	α $[kW/s^2]$
Slow	$t_g \geq 400$	$\alpha \leq 0.0066$
Medium	$150 \leq t_g < 400$	$0.0066 < \alpha \leq 0.0469$
Fast	$t_g < 150$	$\alpha > 0.0469$

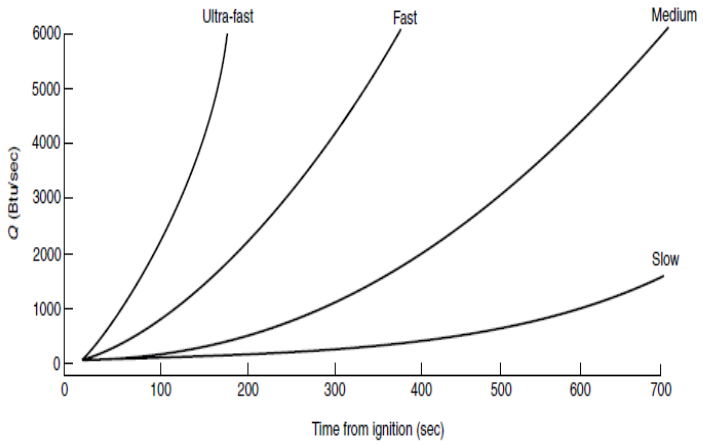


Fig. 1 Power-law heat release rate

For a height H (ceiling height or height above the fire), the temperature rise $\Delta T\,[^0C]$ and velocity u of fire gases at a radial distance from the fire plume axis $r[m]$ are given in dimensionless form as the reduced gas temperature ΔT_2^* and reduced gas velocity u_2^* respectively, by

$$\Delta T_2^* = \frac{\Delta T}{A^{\frac{2}{5}} \frac{T_a}{g} \alpha^{\frac{2}{5}} H^{\frac{-3}{5}}} = g\left(t_2^*, \frac{r}{H}\right) \tag{3}$$

$$u_2^* = \frac{u}{A^{\frac{1}{5}} \alpha^{\frac{1}{5}} H^{\frac{1}{5}}} = f\left(t_2^*, \frac{r}{H}\right) \tag{4}$$

where

$$\Delta T = T - T_a \tag{5}$$

the reduced time t_2^* is:

$$t_2^* = \frac{t}{A^{\frac{-1}{5}} \alpha^{\frac{-1}{5}} H^{\frac{4}{5}}} \tag{6}$$

and

$$A = \frac{g}{C_p T_a \rho_0} \tag{7}$$

In the above equations, T_a is the initial ambient temperature of the room (before the start of the fire), g is the gravitational constant, C_p and ρ_0 are the specific heat and density of air at the ambient conditions, respectively [6].

For t^2–fires authors of [5] developed the following specific relations for ΔT_2^*:

$$\Delta T_2^* = \begin{cases} 0, & t_2^* \le t_{2f}^* \\ \left\{ \dfrac{[t_2^* - t_{2f}^*]}{\left[0{,}188 + 0{,}313 \dfrac{r}{H}\right]} \right\}^{\frac{4}{3}}, & t_2^* > t_{2f}^* \end{cases} \tag{8}$$

where

$$t_{2f}^* = 0{,}954\left(1 + \frac{r}{H}\right) \tag{9}$$

$$D = 0{,}188 + 0{,}313\left(\frac{r}{H}\right) \tag{10}$$

In [2] a method for determining the application spacing for both fixed-temperature heat detectors and rate-of-rise heat detectors is provided. This method is only valid for use when detectors are to be placed on a large, flat ceiling. It predicts detector response to a geometrically growing flaming fire at a specific fire size. This method takes into account the effects of ceiling height, the radial distance between the detector and the fire, threshold fire size [critical heat release rate (QCR)], rate of fire development, and detector response time index. Annex B of [6] assumes a convective heat release rate fraction equal to 75 percent of the total heat release rate. For burning conditions that are substantially different (i.e. different convective heat release rate fraction), the last two equations in Annex B are replaced as follows [7]:

$$t_{2f}^{*} = 0,813\left(1+\frac{r}{H}\right) \tag{11}$$

$$D = 0,126 + 0,210\left(\frac{r}{H}\right) \tag{12}$$

The dimensionless gas velocity for t^2–fires was determined to be:

$$\frac{u_2^*}{(\Delta T_2^*)^{\frac{1}{2}}} = \begin{cases} \dfrac{3,87}{(9,115)^{0,5}}, & \dfrac{r}{H} \le 0,3 \\ 0,59\left(\dfrac{r}{H}\right)^{-0,63}, & \dfrac{r}{H} > 0,3 \end{cases} \tag{13}$$

The expression for the velocity with r/H of 0.3 is based on work [8]. The above expressions for ceiling gas temperature and velocity can be substituted into the heat transfer equation for the temperature of the thermal sensing element of the detector at $T_d(t)$ and can be integrated. Using the condition that the initial temperature of the thermal sensing element of this detector is equal to the ambient temperature ($T_d(0) = T_a$)), the equations for calculating the response of fixed temperature and the rate of temperature rise detectors are from Beyler [9]. Beyler's integration eliminates thousands of mathematical operations by eliminating the iterative solution to the heat transfer equation. The use of a computer program is still required if this technique is to be a common tool for fire protection engineers.

A detector activation temperature for fixed heat detectors can be calculated using the equation:

$$T_d(t) - T_d(0) = \left(\frac{\Delta T}{\Delta T_2^*}\right)\Delta T_2^*\left[1-\frac{(1-e^{-Y})}{Y}\right] \tag{14}$$

A rate of temperature change for rate-of-rise detectors is given in the next equation:

$$\frac{dT_d(t)}{dt} = \frac{\left[\dfrac{4}{3}\dfrac{\Delta T}{\Delta T_2^*}(\Delta T_2^*)^{\frac{1}{4}}(1-e^{-Y})\right]}{\left(\dfrac{t}{t_2^*}\right)D} \tag{15}$$

where

$$Y = \frac{3}{4}\left(\frac{u}{u_2^*}\right)^{\frac{1}{2}}\left(\frac{u_2^*}{\left(\Delta T_2^*\right)^{\frac{1}{2}}}\right)^{\frac{1}{2}}\left(\frac{\Delta T_2^*}{RTI}\right)\left(\frac{t}{t_2^*}\right)D \tag{16}$$

In equation (16), RTI is the response time index [$m^{1/2}s^{1/2}$] which is a measure of the thermal sensitivity of the detector. This is a property of the thermal sensing element itself without external influence such as affecting the convective heat transfer coefficient through the air speed. Equations (1) to (16) constitute a complete mathematical description of the transient response of a thermal detector to a t^2–fire. This is the basis for all

current designs of thermal detection systems. The procedure presented above can be used to estimate the response of rate-of-rise heat detectors for either design or analysis purposes. In this case, it is necessary to assume that the heat detector response can be modeled using a lumped mass heat transfer model. The user must determine the rate of temperature rise at which the detector will respond from the manufacturer's data. The user must use equation (15) instead of equation (14) in order to calculate the rate of change of the detector temperature. This value is then compared to the rate of change at which the chosen detector is designed to respond [2].

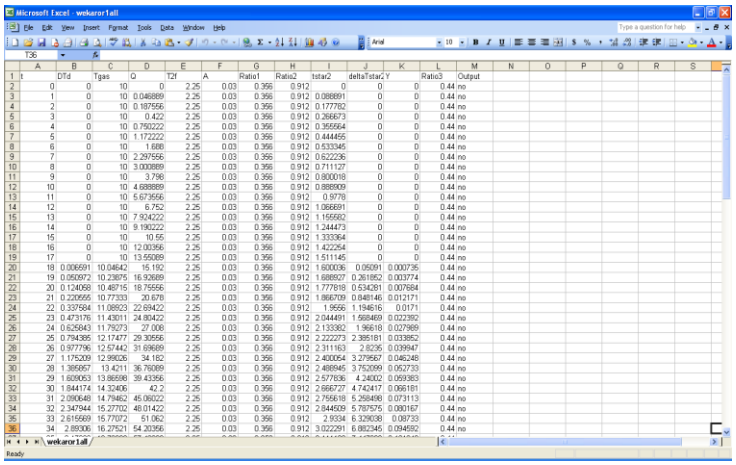
3. DATA MINING USING WEKA

WEKA stands for Waikato Environment for Knowledge Analysis [10]. WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms in WEKA can be applied directly to a dataset, which was performed in this paper. When WEKA is fired up, it is possible to choose among four different user interfaces: the Explorer, the Knowledge Flow, the Experimenter, and command-line interfaces. The easiest way to use WEKA is through a graphical user interface called Explorer. This gives access to all of its facilities using menu selection and form filling [11].

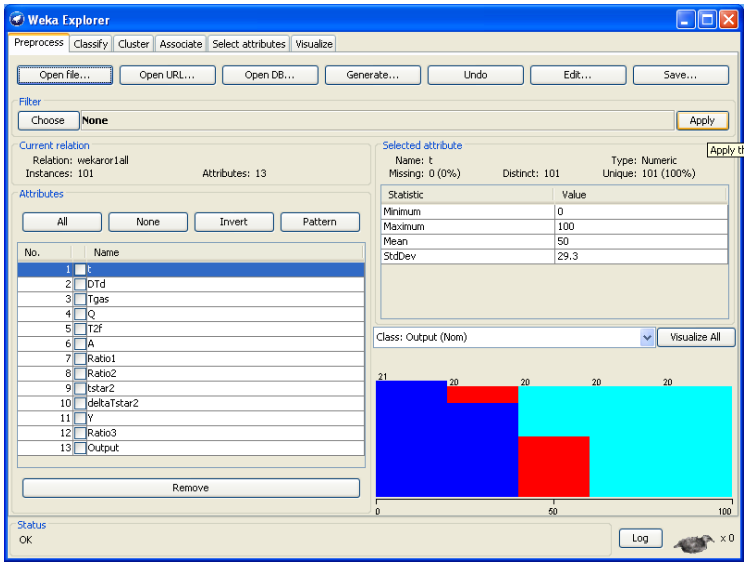
The WEKA Explorer interface contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

3.1. Preparing the data

Preparing input for a data mining investigation usually consumes the bulk of the effort invested in the entire data mining process. The solution of the equations presented above requires thousands of mathematical operations which are best solved by a computer. For the purpose of this paper, the program is written in MATLAB. The parameters used for simulation purposes, in order to apply some of the data mining algorithms on them, include: the initial ambient temperature $T_a = 10$ [^{0}C], the height above the fire $H = 4$ [m], rate-of-rise heat detectors designed to activate at a nominal rate of temperature rise of $ROR = 9$ [^{0}C/min], the spacing between detectors $S = 9.1$ [m], the response time index $RTI = 98$ [$m^{1/2}s^{1/2}$], the critical time $t_c = 150$ [s] (a fast developing fire), the upper limit value of the heat release rate $Q = 5000$ [kW] and simulation time is $t_{sim} = 600$ [s]. For the given parameters, the simulation program written in MATLAB shows that the detector activates at 88.2 [s] after fire ignition with a heat release rate of 365 [kW]. The simulation results are then exported to Excel .csv form (Fig. 2 a)) and later imported to a WEKA data mining tool (Fig. 2 b)). The input to the WEKA package can be in .arff format or Excel .csv. For the experiment described above, there are 13 attributes and 101 instances. For a numeric attribute, it is possible to see its minimum and maximum values, mean, and standard deviation (Fig. 2 b).



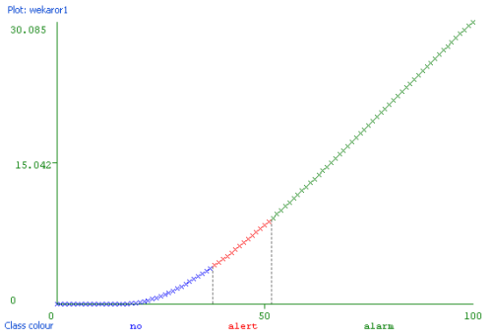
a)



b)

Fig. 2 a) .csv file of sensor data, b) data imported in WEKA

In this paper, some of the data mining techniques are applied incorporating contextual data with sensor data, as the 13th attribute - class. Data from the first 101s after fire ignition are used and it is assumed that the values under approximately *ROR* = 4 [^{0}C/min] belong to class *no*, the values above approximately *ROR* = 9 [^{0}C/min] mean *alarm* and the values in between belong to class *alert* (as is shown in Fig. 3 and 4 using the Visualize tab of WEKA Explorer interface).



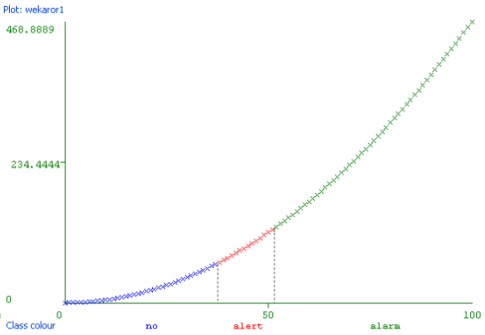


Fig. 3 Rate of temperature rise as a function of time **Fig. 4** Heat release rate as a function of time

By choosing the *Select Attributes* tab it is possible to evaluate the given attributes. *InfoGainAttributeEval* evaluates the attributes by measuring their information gain with respect to the class. It discretizes numeric attributes first using the MDL-based discretization method (it can be set to binarize them instead). This method can treat *missing* as a separate value or distribute the counts among other values in proportion to their frequency. *GainRatioAttributeEval* evaluates attributes by measuring their gain ratio with respect to the class (Table 2.) [11]. Looking at Table 2 it can be concluded that attributes T2f (Eq. 11), A (Eq.7), Ratio1 (Eq. 4), Ratio2 (Eq. 3), Ratio3 (Eq. 13) are constants and have a value of 0 so they can be removed from the attribute list by using the *Remove* filter.

Table 2 Attribute evaluation

Attribute	*InfoGainAttributeEval*	*GainRatioAttributeEval*
t	1.298	0.391
DTd	1.35	0.45
Tgas	1.288	0.424
Q	1.217	0.403
T2f	0	0
A	0	0
Ratio1	0	0
Ratio2	0	0
tstar2	1.3	0.392
deltaTstar2	1.288	0.424
Y	1.389	0.446
Ratio3	0	0

The *ReliefFAttributeEval* is instance-based: It samples instances randomly and checks neighboring instances of the same and different classes. It operates on discrete and continuous class data. Parameters specify the number of instances to sample, the number of neighbors to check, whether to weigh neighbors by distance, and an exponential function that governs how rapidly weights decay with distance [11]. The *ReliefFAttributeEval* obtained results are shown in Table 3:

Table 3 Attribute evaluation using *ReliefAttributeEval*

Attribute	*ReliefAttributeEval*
Y	0.711
DTd	0.688
deltaTstar2	0.688
Tgas	0.688
tstar2	0.76
t	0.778
Q	0.621

3.2. Classification algorithms

The main advantage of using WEKA is the application of learning methods to a dataset and analyzing its output to extract information about the data. These learning methods are called classifiers [12]. In this paper the classifiers from WEKA are used in order to analyze the classification accuracy of simulation data. Classification here means the problem of correctly predicting the probability that an example has a predefined class from a set of attributes describing the example. In addition, the learning algorithms in WEKA can be applied and then the best one can be used for prediction purposes. The following algorithms were used for classification purposes in this work:

Algorithm 1: Decision Tree classifier

Algorithm 2: Naïve Bayes

Algorithm 3: Neural Network classifier

Algorithm 4: Support Vector Machines – SVM

The applied algorithms will briefly be described in the remainder of the paper. More information about them can be found in [11, 12].

The output of the simulator is used to learn about the difference between a subject that is *no*, *alert* and *alarm*. The question of predicting performance based on limited data is an interesting, and still controversial, one. The repeated cross-validation technique is probably the method of choice in most practical limited-data situations. Different testing strategies can be used to train and test based on the given datasets. In the averaging process, the given dataset is divided into two parts. One part is first used to train the classification algorithm. So the percentage of data to be used for training purposes should be specified first (using the option *Percentage Split* in WEKA). Then concepts learned during the training process are used to test the remaining data. For these experiments a 10-fold cross validation testing technique is used. During the process the data set is divided into 10 subsets and the classification algorithms are fed with these subsets of data. The left-out subsets of the training data are used to evaluate classification accuracy. When seeking an accurate error estimate, it is standard procedure to repeat the cross-validation process 10 times (10 times tenfold cross-validation) and average the results. This involves invoking the learning algorithm 100 times on datasets that are all nine-tenths the size of the original. Getting a good measure of performance is a computation-intensive undertaking [11].

3.2.1. Decision Tree Classifier

The decision tree classifier is a tree-based classifier which selects a set of features and then compares the input data with them. Learned patterns are represented as a tree where

the nodes in the tree embody decisions based on the values of attributes and the leaves of the tree provide predictions. The main advantage of a decision tree classifier is its classification speed. WEKA uses the J48 decision tree which is an implementation of the C 4.5 algorithm [12]. The attribute with the maximum gain ratio, as shown in Table 2, is DTd – the rate of temperature change - and it is selected as the splitting attribute (Fig. 5). The numbers given in parentheses are the number of instances assigned to that node number followed by incorrectly classified instances.

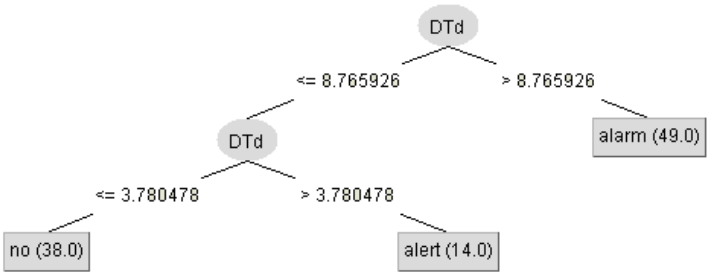


Fig. 5 J48 decision tree

3.2.2. Naïve Bayes

Naïve Bayes gives a simple approach, with clear semantics, for representing, using, and learning probabilistic knowledge. It can achieve impressive results [11]. The Naïve Bayes classifier produces probability estimates rather than hard classifications. For each class value, it estimates the probability that a given instance belongs to that class. Most other types of classifiers can be coerced into yielding this kind of information if necessary.

3.2.3. Neural Network Classifier

The neural network classifier is used for many pattern recognition purposes. It uses the backpropogation algorithm to train the network. The accuracy of the neural network classifiers does not depend on the dimensionality of the training data [12].

3.2.4. Support Vector Machines

The support vector machine classifiers work by generating functions from the input training data. This function is used as a classification function. They operate by finding a hypersurface in the space of possible inputs. This hypersurface will attempt to split the positive examples from the negative examples i.e., no fire from the alarm. If the dimensionality of the input data is high, then the SVM takes more time for training [12].

4. RESULTS

Using the above mentioned algorithms certain simulations were performed and the following results were obtained.

4.1. Summary of accuracy

There are many methods and measures for estimating the strength and the accuracy of a classification/predictive model. Performance measures for numeric predictions are given in Table 4 while simulation results for each of the applied algorithms are presented in Table 5.

Table 4 Performance measures for numeric prediction

Mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
Root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
Mean-absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
Relative-squared error*	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$
Root relative-squared error*	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
Relative-absolute error*	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
Correlation coefficient**	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_P=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$

*Here, $\bar{a}$ is the mean value over the training data.
**Here, $\bar{a}$ is the mean value over the test data.

Table 5 Performance measures for the applied algorithms

J48	
Correctly Classified Instances	99
Incorrectly Classified Instances	2
Kappa statistic	0.9671
Mean absolute error	0.0132
Root mean squared error	0.1149
Relative absolute error	3.2657 %
Root relative squared error	25.5854 %
Coverage of cases (0.95 level)	98.0198 %
Mean rel. region size (0.95 level)	33.3333 %
Total Number of Instances	101

Naïve Bayes	
Correctly Classified Instances	98
Incorrectly Classified Instances	3
Kappa statistic	0.9515
Mean absolute error	0.0201
Root mean squared error	0.1362
Relative absolute error	4.9655 %
Root relative squared error	30.3256 %
Coverage of cases (0.95 level)	98.0198 %
Mean rel. region size (0.95 level)	33.9934 %
Total Number of Instances	101

Neural Network classifiers	
Correctly Classified Instances	99
Incorrectly Classified Instances	2
Kappa statistic	0.9671
Mean absolute error	0.0216
Root mean squared error	0.0864
Relative absolute error	5.3369 %
Root relative squared error	19.2383 %
Total Number of Instances	101

Support Vector Machine	
Correctly Classified Instances	98
Incorrectly Classified Instances	3
Kappa statistic	0.9511
Mean absolute error	0.2288
Root mean squared error	0.284
Relative absolute error	56.6056 %
Root relative squared error	63.2487 %
Total Number of Instances	101

The main measure is the classification accuracy which is the number of correctly classified instances in the test set divided by the total number of instances in the test set. Getting a good measure of performance is a computation-intensive undertaking.

In applications with only two classes, two measures named Precision and Recall are usually used. Their definitions are:

$$P = \frac{TP}{TP + FP} \tag{17}$$

$$R = \frac{TP}{TP + FN} \tag{18}$$

TP, FP, TN, and FN used in Eq. (17) and Eq. (18). They are the numbers of true positives, false positives, true negatives, and false negatives, respectively. These measures can be also used in the case of a larger number of classes, which in this case are seen as a series of problems with two classes. However, it is hard to compare classifiers based on two measures, which are not functionally related. If a single measure to compare different classifiers is needed, the F-measure is often used:

$$FM = \frac{2 \cdot P \cdot R}{P + R} \tag{19}$$

Another measure is the receiver operating characteristic (ROC). It is a term used in signal detection to characterize the tradeoff between hit rate and false-alarm rate over a noisy channel. ROC curves depict the performance of a classifier without regard to class distribution or error costs. They plot the true positive rate on the vertical axis against the true negative rate on the horizontal axis [11].

Table 6 Classifier evaluation

	CCI (%)	*ICI (%)*	*TP*	*FP*	*P*	*R*	*FM*	*ROC*
J48	98.01	1.98	0.98	0.011	0.98	0.98	0.98	0.985
Naïve Bayes	97.02	2.97	0.97	0.005	0.976	0.97	0.971	0.999
Neural Network	98.01	1.98	0.98	0.011	0.98	0.98	0.98	1
Support Vector Machine	97.02	2.97	0.97	0.009	0.972	0.97	0.971	0.989

From Table 6 it can be seen that all the applied algorithms have good predicting performances but the J48 and Neural Network generate the best prediction model among the others. They generated a model with 98% correctly classified instances (CCI), a precision of 98% (0.98) but the Neural Network has a classification above the ROC curve area of 1, which makes it slightly better than J48.

4.2. Classifier Error

The following figures show classifier errors – output vs. predicted output for implemented algorithms. The crosses represent correctly classified and the squares incorrectly classified instances.

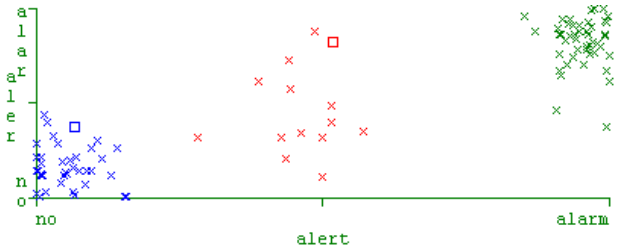


Fig. 6 The classifier error for J48

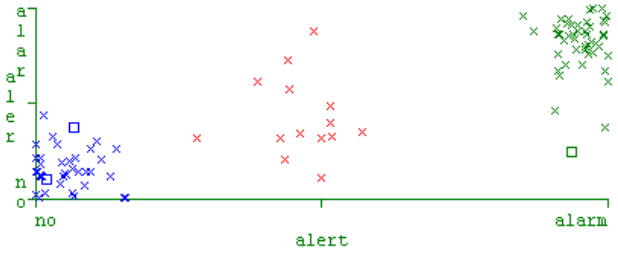


Fig. 7 The classifier error for Naïve Bayes

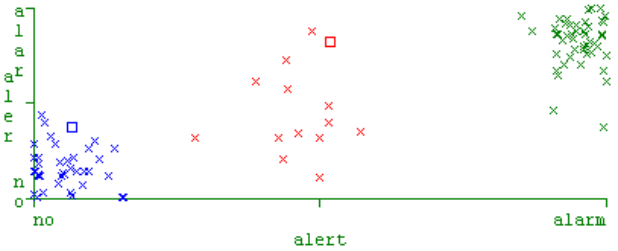


Fig. 8 The classifier error for the Neural Network

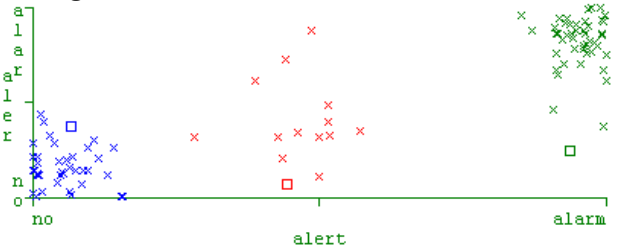


Fig. 9 The classifier error for the Support Vector Machine

4.3. Confusion Matrix

In multiclass prediction, the result on a test set is often displayed as a two-dimensional *confusion matrix* with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements [11]. The obtained confusion matrices with information about actual and predicted results given by the applied classifiers are presented in Table 7.

Table 7 Confusion matrices

J48			
Predicted class			
a	***b***	***c***	***Real class***
37	1	0	*a=no*
0	**13**	1	*b=alert*
0	0	**49**	*c=alarm*

Naïve Bayes			
Predicted class			
a	***b***	***c***	***Real class***
36	2	0	*a=no*
0	**14**	0	*b=alert*
0	1	**48**	*c=alarm*

Neural Network			
Predicted class			
a	***b***	***c***	***Real class***
37	1	0	*a=no*
0	**13**	1	*b=alert*
0	0	**49**	*c=alarm*

Support Vector Machine			
Predicted class			
a	***b***	***c***	***Real class***
37	1	0	*a=no*
1	**13**	0	*b=alert*
0	1	**48**	*c=alarm*

5. CONCLUSION

The goal of predictive modeling is to build a model that can be used to predict - based on known examples collected in the past - future values of a target attribute.

In this paper the model used for simulation is based on the NFPA 72 standard which provides prescriptive solutions that favor active fire suppression. Predictive modeling is performed using data mining algorithms on a small amount of rate-of-rise heat detector simulation data. From the classification results presented above, it can be seen that J48 has two misclassified instances, one instance of class *no* is classified as *alert* and one instance of class *alert* is classified as *alarm*. In the case of Naïve Bayes, three instances are not classified properly, two instances of class *no* are recognized as *alert* and one of class *alert* is classified as *alarm*. The Neural Network has the same percent of correctly classified instances as J48, while the Support Vector Machine has three incorrectly classified instances, one of class *no* is classified as *alert*, one *alert* instance is classified as *no* and one *alarm* instance is wrongly recognized as *alert*. It can be concluded that the J48 Tree classifier model and Neural Network classifier have a higher level of classification accuracy than the Naïve Bayes algorithm and Support Vector machine, which makes them more appropriate for the case of the rate-of-rise heat detector prediction model. The advantage of the Neural Network classifier is the classification above the ROC curve area of 1, which makes it slightly better than J48.

Future work will be based on combining data from different sensors considering large amounts of data. Prediction in the sensor networks can be performed in a way that each sensor learns a local predictive model for the global target classes, using only its local input data. By classifying the large dataset, at the sensor nodes level, normal values can be discarded and only the anomaly values will be transmitted to the central server. Thus, the number of sensors that need to report their measurements will be reduced by reducing

both node activity and bandwidth. It should also lead to faster response time of detectors which will directly influence the response time of the fire protection system entirely.

Also, directions of further research will be based on the EN54 standard which tends to provide performance requirements that favor passive fire protection. The purpose of the research will be to analyze the pros and cons between NFPA 72 and EN 54 standards in this particular application case.

REFERENCES

1. Bahrepour M., Meratnia N. and Havinga P.J.M., (2009) Use of AI Techniques for Residential Fire Detection in Wireless Sensor Networks, AIAI-2009 Workshops Proceedings, pp. 311-321, Thessaloniki, Greece
2. NFPA 72, National Fire Alarm Code®, 1999 Edition and later
3. Aggarwal C. C, (2013) Managing and Mining Sensor Data, Springer US
4. Yabuki1 N., Yoshida Y., Tsukamoto S. and Fukuda T., (2011) Data Storage and Data Mining of Building Monitoring Data with Context, Proceedings of the 28th International Symposium on Automation and Robotics in Construction (ISARC2011), pp. 377-378, Seoul, Korea
5. Heskestad G. and Delichatsios M., (1977) Environments of Fire Detectors - Phase I: Effect of Fire Size, Ceiling Height and Material, Volume II "Analysis", National Technical Information Service (NTIS) Springfield, VA 22151.
6. Yuen Walter W., Chow W.K., (2005) A Monte Carlo Approach for the Design of Thermal Fire Detection System, NFPA – Fire Technology, Vol. 41, No. 1, pp. 93-104
7. Antono Ir. M.H., Automatic Fire Detector Spacing, MSFPE Introductions an Appendix, i. e. of NFPA72,b. AppendixB
8. Zukoski E. E., Kubota T., and Cetegan B., (1980) Entrainment in Fire Plumes, Fire Safety Journal, Vol. 3, pp. 107-121.
9. Beyler C., (1984) A Design Method For Flaming Fire Detection, Fire Technology, Vol. 20, No. 4, pp. 5-16
10. WEKA data mining tool, Available: www.cs.waikato.ac.nz/ml/WEKA
11. Witten I. H., Frank E. and Hall M.A., (2011) Data mining: practical machine learning tools and techniques, Morgan Kaufmann, Amsterdam
12. Natajaran J., (2006) Simulation of sensor responses of advanced security systems, Master Thesis, University of Texas at Arlington.

DUBINSKA ANALIZA I PREDIKCIJA PODATAKA TERMODIFERENCIJALNOG DETEKTORA TOPLOTE

Ekstremne situacije, kao što je požar, mogu prouzrokovati masivna oštećenja objekata i situacije opasne po ljudske živote. Rana detekcija požara je važna zbog spašavanja ljudskih života, brzog gašenja požara i smanjenja štete. Za detekciju požara obično se koristi jedan ili kombinacija više senzora (detektori toplote, dima, plamena) i algoritam detekcije. Senzori mogu biti dio bežičnih senzorskih mreža ili raditi samostalno. U ovom radu primjenili smo neke od algoritama dubinske analize podataka na simulacijske podatke termodiferencijalnog detektora toplote. Dubinska analiza podataka (data mining) se pokazala kao efikasna tehnika u pogledu otkrivanja korisnog znanja iz velike količine podataka prikupljene od strane mnogobrojnih senzora. Predikcije u senzorskim mrežama mogu biti izvedene na način da svaki senzor uči lokalni predikcijski model koristeći jedino sopstvene lokalne ulazne podatke. Samo ciljne prediktivne klase svakog očitanja se šalju ka gateway-u ili baznoj stanici. Važna klasa svakog algoritma jesu prediktori, koji koriste senzorske ulazne podatke za predikciju izlazne funkcije od interesa. Cilj rada je bio izvesti analizu nekoliko različitih klasifikacijskih algoritama kako bi ustanovili koji od njih daje najbolje rezultate u slučaju termodiferencijalnog detektora toplote, okarakterisani visokom tačnošću i malom greškom.

Ključne reči: *dubinska analiza podataka, J48, naivni Bayes, neuronske mreže, termodiferencijalni detektor, SVM*