Jestr

Journal of Engineering Science and Technology Review 7 (4) (2014) 89- 96

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Applying Fuzzy Logic and Data Mining Techniques in Wireless Sensor Network for Determination Residential Fire Confidence

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

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

Received 29 May 2014; Accepted 23 September 2014

Abstract

The main goal of soft computing technologies (fuzzy logic, neural networks, fuzzy rule-based systems, data mining techniques…) is to find and describe the structural patterns in the data in order to try to explain connections between data and on their basis create predictive or descriptive models. Integration of these technologies in sensor nodes seems to be a good idea because it can significantly lead to network performances improvements, above all to reduce the energy consumption and enhance the lifetime of the network. The purpose of this paper is to analyze different algorithms in the case of fire confidence determination in order to see which of the methods and parameter values work best for the given problem. Hence, an analysis between different classification algorithms in a case of nominal and numerical data sets is performed with the goal to realize which of applied techniques obtain higher accuracy and less error

Keywords: fuzzy logic, data mining, fire confidence, classification algorithms

1. Introduction

The advancement in sensors' technology has increased sensor networks applications in many domains such as industrial monitoring, building and home automation, medicine, environmental monitoring, urban sensor networks, intelligent transportation. These networks can also be used for security, military defense, disaster monitoring, prevention, etc [1]. Two main applications of wireless sensor networks (WSNs) to indoor environmental monitoring include: collecting information on environmental physical parameters in order to better control environmental systems such as HVAC (heating, ventilation, and air-conditioning), and emergency services such as fire and smoke detection [2]. A critical event, like fire can cause heavy structural damage to the indoor area and life threatening conditions so early residential fire detection is important for prompt extinguishing and reducing damages and life losses.

Sensor data mining is a relatively new area but it already reached a certain level of maturity. Data mining in sensor networks is the process of extracting application-oriented models and patterns with acceptable accuracy from a continuous, rapid, and possibly non ended flow of data streams from sensor networks [3]. But the extraction of useful knowledge from raw sensor data is a difficult task and traditional data mining techniques are not directly applicable to WSNs due to the distributed nature of sensor data and their special characteristics (the massive quantity and the high dimensionality), and limitations of the WSNs and sensor nodes [4]. This is the reason for exploring novel data mining techniques dealing with extracting knowledge from large continuous arriving data from WSNs [5]. The main goal of data mining techniques is to find and describe the structural patterns in the data in order to attempt to explain connections between data and create predictive models based on them. It draws upon ideas from diverse disciplines such as statistics, machine learning, pattern recognition, database systems, information theory, and artificial intelligence [6]. For such reasons, in recent years a great interest emerged in the research community in applying data mining techniques to the large volumes of sensor data. Input data for applying data mining techniques are presented in the form of a set of examples, and the output can be expressed in the predictive or descriptive form of the analyzed data structure. Thus, techniques used in data mining are categorized into two classes [2]:

- Predictive algorithms - these algorithms usually build a mapping function based on a set of input and output observations (regression modeling, decision trees, neural networks, K-nearest neighbor, and Bayesian learning algorithms).
- Descriptive algorithms - used for exploratory data analysis to discover individual patterns, such as associations, clusters, and other patterns that can be of interest to the user.

The process of a typical data mining project is shown in Fig. 1.

Data mining, as an iterative process of extracting hidden patterns from large data sets and a critical component of the knowledge discovery process, consists of three main steps:

- Pre-processing – includes cleaning of raw data in order to become suitable for mining (removing noises and abnormalities, handling too large data, identifying and removing irrelevant attributes, etc.)

- Data mining – the process of applying data mining algorithm that will produce patterns or knowledge.
- Post–processing – Among all discovered patterns or knowledge, it is necessary to discover ones that are useful for the application. For making the right decision there are many evaluation and visualization techniques that can be used.

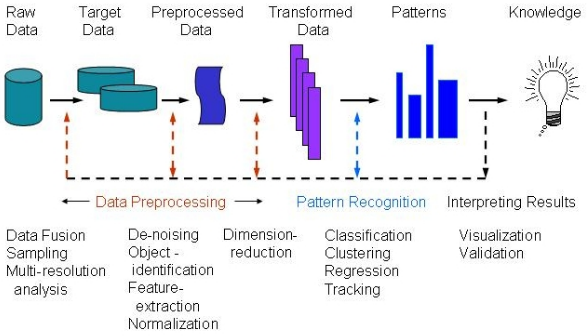


Fig. 1. The overall process of knowledge discovery from data includes data pre-processing, data mining, and post-processing of the data mining results.

The knowledge discovery process needs to coordinate adaptive predictive analysis with real-time analysis and decision support systems. The ability to detect precursors and signatures of rare events and change from massive and disparate data in real time is a challenge [7]. 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. There are many predictive modeling methods available, including tree-based, rule-based, nearest neighbor, logistic regression, artificial neural networks, graphical methods, and support vector machines. These methods are designed to solve two types of predictive modeling tasks: classification and regression [5]. Using these prediction models the number of sensors that need to report their measurements is reduced by reducing both node activity and bandwidth.

Sensor data brings numerous challenges with it in the context of data collection, storage and processing and 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. On the one hand, massive volumes of disparate data, typically dimensioned by space and time, are being generated in real time or near real time. On the other hand, the need for faster and more reliable decisions is growing rapidly in the face of emerging challenges like fire. One critical path to enhanced threat recognition is through online knowledge discovery based on dynamic, heterogeneous data available from strategically placed wide-area sensor networks.

The massive streams of sensor data generated in some applications make it impossible to use algorithms that must store the entire data into main memory. Using data mining techniques in process of patterns discovery in large data sets it's not often so easy. A several algorithms must be applied to application before a suitable algorithm for selected data types can be found. Online algorithms provide an attractive alternative to conventional batch algorithms for handling such large data sets. The selection of a correct data mining algorithm depends on not only the goal of an application, but also on the compatibility of the data set.

The framework shown in Fig. 2 consists of two sub-frameworks [8]:

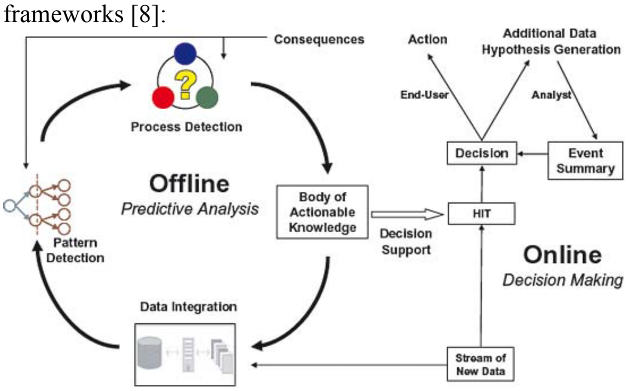


Fig. 2. A holistic approach to knowledge discovery

- Offline predictive analysis (data integration, pattern detection and process detection);
- Online decision making (includes online (real-time) analysis from models and observations, algorithmic efficiency for dynamic, distributed processing of sensor observations, resiliency, vulnerability, and impacts of observations, visualization and decision or policy aids models).

This paper focuses on comparative analysis of various data mining techniques and algorithms with primary goal to see which of them has the best classification accuracy and is the most appropriate for a particular application of fire detection uncovering useful information hidden in large quantities of sensor data. This kind of analysis provide an opportunity for data mining researchers to develop more advanced methods for handling some of the issues specific to sensor data [9].

The rest of this paper is organized as following. Second section presents data preparation file while third section provides an implementation of selected data mining techniques. The experimental results including comparative analysis of selected algorithms are shown in fourth section. Fifth section gives the conclusion.

2. Fire Detection – Preparing the Input Files

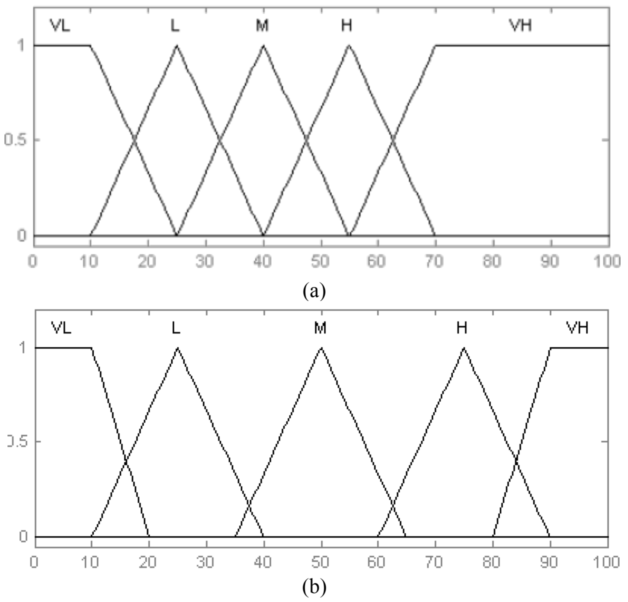
Preparing input for a data mining investigation usually consumes the most of the effort invested in the entire data mining process. Data cleaning is very labor-intensive procedure but it is absolutely necessary step for successful data mining.

In this work different experiments for fire confidence determination will be presented. The proposed experiment were created with main goal to show how chosen algorithms predicting power depends on type of data and the fire detection method.

Early detection of critical events, like residential fire, is crucial for life saving and reduction of potential damages so WSN should be able to detect if fire has occurred or is about to. But just like many other human-recognizable events, the phenomenon fire has no real meaning to a sensor node. Therefore, suitable techniques that would allow describing events in ways that sensor nodes would be able to "understand" are needed [10]. One of them is fuzzy technique. What makes fuzzy logic suitable for use in WSNs is that it can tolerate unreliable and imprecise sensor readings, it is much closer to human way of thinking than crisp logic and compared to other classification algorithms based on probability theory, fuzzy logic is much more intuitive and easier to use [4]. It allows using linguistic variables whose values are not numbers but words or sentences in a natural or artificial language. Fuzzy rules are conditional statements in the form of IF-THEN which:

- Require less computational power than conventional mathematical computational methods,
- Require few data samples in order to extract the final result,
- and the most important, it can be effectively manipulated since they use human language to describe problems (based on heuristic information that mainly comes from expert knowledge of the system) and making the creation of rules simple, independently of the previous knowledge in the field of fuzzy logic.

Input data in the first experiment are defined as IF-THEN rules based on heuristic information that mainly comes from expert knowledge of the fire detection systems. In other words, detection of fire is based on two successively measured values of heat detector. A 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. 57 ºC, 63 ºC, 74 ºC or 90 ºC). Instead of using these crisp values, fuzzy logic proposes use of linguistic variables. Therefore, data obtained from temperature detector according to fuzzy technique and above mentioned thresholds, for the purpose of the experiment are described with values: *very low* (VL), *low* (L), *medium* (M), *high* (H) and *very high* (VH) and presented with membership functions shown in Fig. 3 a). Due to their simple formulas and computational efficiency, both triangular and trapezoidal membership functions have been used extensively, especially in real-time implementations as it is fire detection.



(a)

(b)

Fig. 3. The membership functions of a) input variables *previous temperature* and *current temperature*, b) output variable *fire confidence*

Confidence of *fire* is defined as output variable and is described with *very low* (VL), *low* (L), *medium* (M), *high* (H) and *very high* (VH) linguistic variables as it shown in Fig. 3 b). This linguistic variable represents the system's confidence in the presence of fire.

With 2 variables each of which can take 5 values, the number of rules in the full fuzzy rule-base of the first experiment is 25 (5*5). Tab. 1 shows first 10 rules for 1st experiment.

Table 1. The 1st fire data test (first 10 rules)

Previous temperature	Current temperature	Fire confidence (class)
VL	VL	VL
VL	L	L
VL	M	M
VL	H	M
VL	VH	H
L	VL	VL
L	L	L
L	M	M
L	H	H
L	VH	VH

To test the proposed approach a sensor node prototype is created, precisely a prototype of Sensor Web node. This sensor node was exposed to temperature changes created to simulate an environment of temperature rise in real time. Thus, in the second experiment, detection of fire is based on 26 successively measured fixed heat temperature detector crisp data shown in Fig. 4.

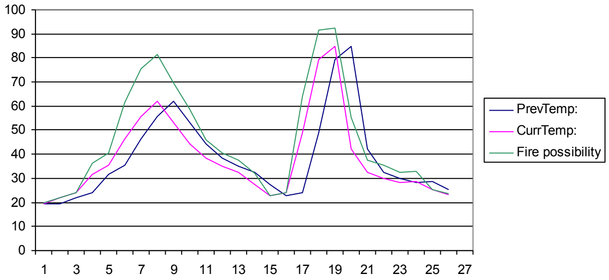


Fig. 4. Measured values of *previous* and *current temperature* and obtained *fire confidence* using rules presented in 1st experiment.

For the sake of clarity of machine learning domain the correlated output sensor data used for a detection of fire are converted to nominal types. First 10 measured values and converted output are presented in Tab. 2.

Table 2. The 2nd fire data test (first 10 measurements)

Previous temperature	Current temperature	Fire confidence	Fire confidence–nominal (class)
19.4	19.4	19.98	VL
19.4	21.95	22.03	L
21.95	23.87	23.96	L
23.87	31.51	36.37	L
31.51	35.44	40.35	M
35.44	46.54	61.41	H
46.54	55.49	75.53	H
55.49	62.08	81.37	VH
62.08	53.17	69.49	H
53.17	44.27	58.6	M

For further analysis Excel .csv data files are formed based on data given in Tables 1 and 2. The next step is their exporting to WEKA data mining tool [11] in order to apply chosen predictive models which can be built to estimate physical parameters (temperature, fire confidence) or descriptive algorithms for finding the relationship between different variables.

3. Temperature sensors' data mining - Classification algorithms implementations

The massive streams of sensor data which could be generated in fire detection applications make it impossible to use algorithms that must store the entire data into main memory. For that purpose, on full rule-base consisted of fuzzy rules for detection of fire, presented in the Tab. 1, FURIA (Fuzzy Unordered Rule Induction Algorithm) [12] will be applied. Other four chosen algorithms will be compared to results obtained using FURIA with aim to realize which of them generate the best prediction models uncovering useful information hidden in large quantities of sensor data in a case of fire detection.

Clustering is probably the most frequently used data mining algorithm, used as exploratory data analysis or included in other data mining techniques like regression or classification. In simulation process presented in this paper four widely used classification algorithms [13] are implemented for comparative analysis with FURIA on given fire data sets. Thus, the comparative analysis is based on following algorithms:

- FURIA
- J48 decision tree
- Naive Bayes
- Support Vector Machines (SVM)
- Neural Network classifier

3.1. FURIA

FURIA (Fuzzy Unordered Rule Induction Algorithm) is a fuzzy rule-based classification method which extends the well-known RIPPER algorithm preserving its advantages, such as simple and comprehensible rule sets. In addition, FURIA includes a number of modifications and extensions. It obtains fuzzy rules instead of the usual strict rules, as well as an unordered rule set instead of the rule list. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method. The idea is to generalize the existing rules until they cover the example [12].

3.2. Decision Tree Classifier

WEKA uses the J48 decision tree which is an implementation of the C 4.5 algorithm. The decision tree classifier is a tree based classifier which selects a set of features and then compares the input data with them and its main advantage is classification speed. Learned patterns are represented as a tree where nodes in the tree embody decisions based on the values of attributes and the leaves of the tree provide predictions [13].

3.3. Naïve Bayes

The Naïve Bayes classifier (based on Bayes' theorem.) for each class value, estimates the probability that a given instance belongs to that class. It is a statistical classifier and performs probabilistic prediction, i.e., predicts class membership probabilities. The Naïve Bayes Classifier has comparable performance with decision tree and selected neural network classifiers. Each training example can incrementally increase/decrease the probability that a hypothesis is correct - prior knowledge can be combined with observed data. Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision [14]. Naïve Bayes gives a simple approach, with clear semantics, for representing, using, and learning probabilistic knowledge and it can achieve impressive results [13].

3.4. Support vector machine (SVM)

The SVM 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 attempt to split the positive examples from the negative examples i.e., low from high. If the dimensionality of the input data is high then the SVM takes more time for training [15].

3.5. 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 [13]. Weakness of neural network classifier can be summed up into: long training time, requiring a number of parameters typically best determined empirically, and poor interpretability. On the other side strength of this classifiers are: high tolerance to noisy data, ability to classify untrained patterns, well-suited for continuous-valued inputs and outputs, successful on a wide array of real-world data, etc.

3.6. Measures for estimation the strength and the accuracy of a classification/predictive models

There are many methods and measures for estimation the strength and the accuracy of a classification/predictive model. 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. Some of the common methods for classifier evaluation are holdout set, Multiple Random Sampling and Cross-validation [13].

The output of the simulator proposed in this paper is used to learn the difference between a subject that is VL, L, M, H and VH. For these experiments averaging and 10-fold cross validation testing techniques are used. During the process the data set is divided into 10 subsets. Then 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 (that is 10 times tenfold cross-validation) and average the results. This involves invoking the learning algorithm 100 times on data sets that are all nine-tenths the size of the original. Getting a good measure of performance is a computation-intensive undertaking [13].

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

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

TP, FP and FN used in Eq. (1) and Eq. (2) are the numbers of true positives, false positives and false negatives, respectively. These measures can be also used in case of larger number of classes, which in this case are seen as a series of problems with two classes. It is convenient to introduce these measures using a confusion matrix. A confusion matrix contains information about actual and predicted results given by a classifier. However, it is hard to compare classifiers based on two measures, which are not functionally related [13]. If a single measure to compare different classifiers is needed, the F-measure is often used:

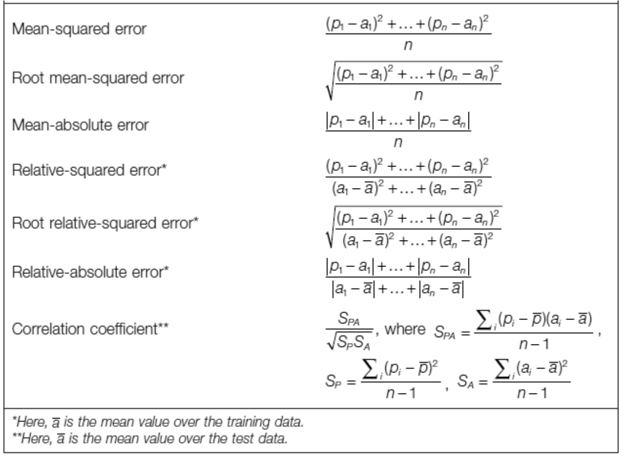
$$FM = \frac{2 \cdot P \cdot R}{P + R} \tag{3}$$

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.

In addition, it is possible to evaluate attributes by measuring their information gain with respect to the class using Info-Gain Attribute Evaluation and measuring their gain ratio with respect to the class using Gain-Ratio Attribute Evaluation [13]. Information gain is biased towards multivalued attributes while gain ratio tends to prefer unbalanced splits in which one partition is much smaller than the others.

There are many other methods and measures for estimation the strength and the accuracy of a classification/predictive model. Performance measures for numeric prediction are given in Tab. 3.

Table 3. 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.

In rest of the paper comparative analysis, using FURIA as base predictive model will be performed.

4. Simulation results

Implementations of chosen classification algorithms are performed in WEKA, which is a collection of machine learning algorithms for data mining tasks [11]. The algorithms in WEKA can be applied directly to previous formed data sets as it is used in this paper. The main advantage of using WEKA is to apply the learning methods to a data set and analyze its output to extract information about the data. These learning methods are called classifiers. In simulation process the classifiers from WEKA in order to analyze the classification accuracy of simulation data are used. 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 [13]. The purpose is to apply the learning algorithms and then to choose the best one for prediction purposes.

Simulation results (performances and classifier error) of above described experiments and chosen algorithms are shown in rest of the paper. It will be shown which of applied algorithms has the highest percentage of correct classified instances (CCI), the minimal of incorrect classified instances (ICI), the highest precision (P) and the classification above ROC curve area in function of chosen experiment and its number of data.

Attributes evaluation of data given in Tab. 1 are shown in Tab. 4.

Table 4. Attributes evaluation – 1st experiment

Attribute	*InfoGainAttributeEval*	*GainRatioAttributeEval*
Previous temperature	0.0684	0.0294
Current temperature	2.0215	0.8706

Applying FURIA classifier to existing rules shown in Tab. 1, 25 rules are generalized into only 5 (Tab. 5). It can be seen that in rules generated by FURIA output depends only of *current temperature* values.

Table 5. The fire data test obtained using FURIA in 1st experiment

Previous temperature	Current temperature	Fire confidence (class)
/	VL	VL
/	L	L
/	M	M
/	H	H
/	VH	VH

J48 decision tree for presented fire data in 1st experiment is shown in Fig. 5. The attribute with the maximum gain ratio, as it is showed in Tab. 4, is *Current temperature* and it is selected as the splitting attribute.

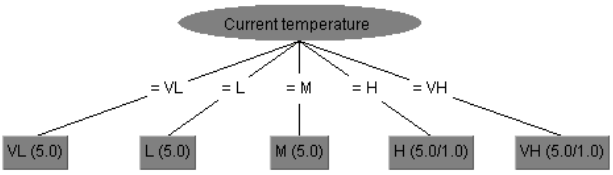


Fig. 5. J48 decision tree – 1st experiment

Classifiers evaluation is presented in Tab. 6.

Table 6. Classifier evaluation – 1st experiment

	CCI (%)	*ICI (%)*	*TP*	*FP*	*P*	*R*	*FM*	*ROC*
FURIA	92	8	0.92	0.018	0.928	0.92	0.92	0.951
J48	92	8	0.92	0.018	0.928	0.92	0.92	0.935
NB	92	8	0.92	0.018	0.928	0.92	0.92	0.904
SVM	92	8	0.92	0.018	0.928	0.92	0.92	0.928
NN	84	16	0.84	0.043	0.859	0.84	0.842	0.929

From Tab. 6 it can be seen that first four classifiers generate the same prediction model with 92% correctly classified instances (CCI), a precision of 92.8% (0.928) but the FURIA has the best classification above the ROC curve area (0.951).

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 [13]. The results are shown in Tab. 7. FURIA, J48, Naïve Bayes and SVM have the same confusion matrix.

Table 7. Confusion matrices – 1st experiment

FURIA, J48, NB, SVM						***Neural Network***					
Predicted class						***Predicted class***					
a	***b***	***c***	***d***	***e***	***Real class***	***a***	***b***	***c***	***d***	***e***	***Real class***
5	0	0	0	0	*a=VL*	**4**	0	1	0	0	*a=VL*
0	**5**	0	0	0	*b=L*	0	**4**	1	0	0	*b=L*
0	0	**5**	1	0	*c=M*	0	0	**5**	1	0	*c=M*
0	0	0	**4**	1	*d=H*	0	0	0	**4**	1	*d=H*
0	0	0	0	**4**	*e=VH*	0	0	0	0	**4**	*e=VH*

Performance measures for applied algorithms according to Tab. 3 are given in Tab. 8.

Table 8. Performance measures for applied algorithms

FURIA	
Correctly Classified Instances	23
Incorrectly Classified Instances	2
Kappa statistic	0.9
Mean absolute error	0.032
Root mean squared error	0.1789
Relative absolute error	9.82 %
Root relative squared error	43.83 %
Coverage of cases (0.95 level)	92 %
Mean rel. region size (0.95 level)	20 %

J48	
Correctly Classified Instances	23
Incorrectly Classified Instances	2
Kappa statistic	0.9
Mean absolute error	0.064
Root mean squared error	0.2
Relative absolute error	19.64 %
Root relative squared error	49 %
Coverage of cases (0.95 level)	92 %
Mean rel. region size (0.95 level)	26.4 %

Naïve Bayes	
Correctly Classified Instances	23
Incorrectly Classified Instances	2
Kappa statistic	0.9
Mean absolute error	0.2634
Root mean squared error	0.3369
Relative absolute error	80.85 %
Root relative squared error	82.55 %
Coverage of cases (0.95 level)	100 %
Mean rel. region size (0.95 level)	100 %

Support Vector Machine	
Correctly Classified Instances	23
Incorrectly Classified Instances	2
Kappa statistic	0.9
Mean absolute error	0.2496
Root mean squared error	0.3284
Relative absolute error	76.6 %
Root relative squared error	80.46 %
Coverage of cases (0.95 level)	96 %
Mean rel. region size (0.95 level)	84.8 %

Neural Network	
Correctly Classified Instances	21
Incorrectly Classified Instances	4
Kappa statistic	0.7992
Mean absolute error	0.0866
Root mean squared error	0.209
Relative absolute error	26.57 %
Root relative squared error	51.21 %
Coverage of cases (0.95 level)	92 %
Mean rel. region size (0.95 level)	35.2 %

If fire confidence of 2nd experiment is presented as a class and with nominal values according to Fig. 3, applying above mentioned classifiers next results are obtained.
The attribute with the maximum gain ratio is again *Current temperature* and it is selected as the splitting attribute in J48 decision tree (Fig. 6)

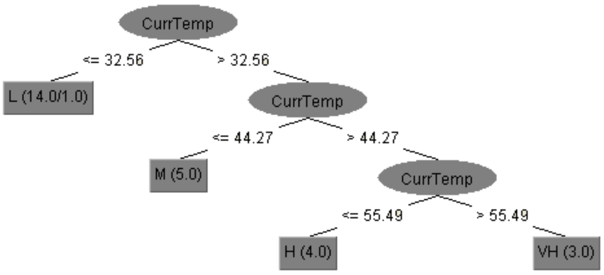


Fig. 6. J48 decision tree – 2nd experiment

Classifiers evaluation in 2nd experiment is presented in Tab. 9.

Table 9. Classifier evaluation – 2nd experiment

	CCI (%)	***ICI (%)***	***TP***	***FP***	***P***	***R***	***FM***	***ROC***
FURIA	84.61	15.38	0.846	0.064	0.82	0.846	0.822	0.947
J48	84.61	15.38	0.846	0.055	0.828	0.846	0.827	0.868
NB	65.38	34.61	0.654	0.123	0.658	0.654	0.654	0.867
SVM	50	50	0.5	0.439	0.271	0.5	0.351	0.523
NN	80.76	19.23	0.808	0.104	0.808	0.808	0.791	0.899

From Table 9 it can be seen that FURIA and J48 generate the same prediction model with 84.6% correctly classified instances (CCI), a precision of 82% (0.928) but the FURIA has the best classification above the ROC curve area (0.947).

For a more detailed analysis of the class attribute distribution, in Table 10 are shown confusion matrices of the 2nd experiment.

Table 10. Confusion matrices – 2nd experiment

FURIA					
Predicted class					
a	***b***	***c***	***d***	***e***	***Real class***
0	1	0	0	0	*a=VL*
0	**13**	0	0	0	*b=L*
0	0	**5**	0	0	*c=M*
0	0	2	**2**	0	*d=H*
0	0	0	1	**2**	*e=VH*

J48					
Predicted class					
a	***b***	***c***	***d***	***e***	***Real class***
0	1	0	0	0	*a=VL*
0	**13**	0	0	0	*b=L*
0	0	**4**	1	0	*c=M*
0	0	0	**2**	2	*d=H*
0	0	0	0	**3**	*e=VH*

NB					
Predicted class					
a	***b***	***c***	***d***	***e***	***Real class***
0	1	0	0	0	*a=VL*
1	**11**	1	0	0	*b=L*
0	1	**2**	2	0	*c=M*
0	0	1	**2**	1	*d=H*
0	0	0	1	**2**	*e=VH*

SVM					
Predicted class					
a	***b***	***c***	***d***	***e***	***Real class***
0	1	0	0	0	*a=VL*
0	**13**	0	0	0	*b=L*
0	5	**0**	0	0	*c=M*
0	4	0	**0**	0	*d=H*
0	1	1	1	**0**	*e=VH*

Neural network					
Predicted class					
a	***b***	***c***	***d***	***e***	***Real class***
0	1	0	0	0	*a=VL*
0	**12**	1	0	0	*b=L*
0	1	**4**	0	0	*c=M*
0	0	2	**2**	0	*d=H*
0	0	0	0	**3**	*e=VH*

Performance measures for applied algorithms according to Tab. 3 are given in next table.

Table 11. Performance measures for applied algorithms in 2nd experiment

FURIA	
Correctly Classified Instances	22
Incorrectly Classified Instances	4
Kappa statistic	0.7642
Mean absolute error	0.0579
Root mean squared error	0.1965
Relative absolute error	20.39 %
Root relative squared error	52.25 %
Coverage of cases (0.95 level)	96.15 %
Mean rel. region size (0.95 level)	24.61 %

J48	
Correctly Classified Instances	22
Incorrectly Classified Instances	4
Kappa statistic	0.7673
Mean absolute error	0.0763
Root mean squared error	0.2505
Relative absolute error	26.88 %
Root relative squared error	66.6 %
Coverage of cases (0.95 level)	84.61 %
Mean rel. region size (0.95 level)	29.23 %

Naïve Bayes	
Correctly Classified Instances	17
Incorrectly Classified Instances	9
Kappa statistic	0.488
Mean absolute error	0.1399
Root mean squared error	0.3384
Relative absolute error	49.27 %
Root relative squared error	89.98 %
Coverage of cases (0.95 level)	80.76 %
Mean rel. region size (0.95 level)	26.92 %

Support Vector Machine	
Correctly Classified Instances	13
Incorrectly Classified Instances	13
Kappa statistic	0.0479
Mean absolute error	0.2815
Root mean squared error	0.3756
Relative absolute error	99.16 %
Root relative squared error	99.88 %
Coverage of cases (0.95 level)	96.15 %
Mean rel. region size (0.95 level)	80 %

Neural Network	
Correctly Classified Instances	21
Incorrectly Classified Instances	5
Kappa statistic	0.7059
Mean absolute error	0.1344
Root mean squared error	0.2516
Relative absolute error	47.34 %
Root relative squared error	66.89 %
Coverage of cases (0.95 level)	96.15 %
Mean rel. region size (0.95 level)	48.46 %

5. Conclusion

The main purpose of sensors networks for fire detection is to collect the monitored original data, and provide basic information and decision support for monitoring center. Also, data mining algorithm has to be sufficiently fast to process high-speed arriving data. The sensor scenario may often require in-network processing, wherein the data is processed to higher level representations before further processing. On this way, individual nodes access and process local information and in order to achieve a collective decision, they must communicate to neighbor nodes, to send local and partial models and negotiate a common decision. In this case, whole data cannot be stored and must be processed immediately by their compressing and filtering for more effective mining and analysis in order to generate actionable insights from massive, disparate and dynamic data, in real time or near real time. This reduces the transmission costs, and the data overload from a storage perspective.

The aim of this paper was to make a comparative analysis between different classification algorithms, applied on nominal and real time recorded fire data, and to see which of applied techniques has the best prediction performances in order to reduce sensor node activity and bandwidth. For evaluation of classification methods next measures can be used: accuracy, speed, time to construct the model (training time), time to use the model (classification/prediction time), robustness (handling noise and missing values), scalability, interpretability, understanding and insight provided by the model and other measures (e.g., goodness of rules).

For simulation results the standard measures for evaluation of the accuracy of the predictive model (the number of correctly and incorrectly classified examples, TP, FP, recall, precision, F-measure, the area of the ROC curve) were applied while confusions matrix was used for a more detailed analysis of the class attribute distribution. According to chosen evaluation measures, FURIA, which was used as a base prediction model, has shown the best prediction power in both performed experiments. J48 decision tree algorithm generated second best results with same classification accuracy as FURIA but with less classification accuracy above the ROC curve area. Even applied data mining techniques are efficient, none of them can be considered as unique or general solution. On the contrary the selection of a correct data mining algorithm depends of an application and the compatibility of the observed data set. Thus, each situation should be considered as a special case and choice of adequate predictor or classifier should be performed very carefully based on empirical arguments.

Real time data set used in those experiments is just an example, and for getting better and more accurate results the larger data sets should be used. Our future work will be based on measuring and combining real data from different sensors and selecting the best prediction model for the given application classifying large data set at the sensor node level, discarding normal values and transmitting only anomaly values to the central server what will decrease the potential network traffic and prolong network life span making early fire detection possible.

References

[1] F. Ali, A Middleware to Connect Software Applications with Sensor Web, The International Journal of Technology, Knowledge and Society, Vol. **6**, No. 5, ISSN 1832-3669, pp. 27-35, (2010)

[2] S. Wu, D. Clements-Croome, Understanding the indoor environment through mining sensory data—A case study, Energy and Buildings **39**, pp. 1183–1191, (2007)

[3] C.C. Aggarwal, Managing and Mining Sensor Data, Springer US, (2013)

[4] M. Maksimović, V. Vujović, V. Milošević, Mining and predicting rate of rise heat detector data, Facta Universitatis, Series: Working and Living Environmental Protection, Vol. **10**, No 1, pp. 37 – 51, (2013)

[5] A. Mahmood et al. Mining Data Generated by Sensor Networks: A Survey, Information Technology Journal, **11(11):** 1534-1543, (2012)

[6] P.N. Tan, Knowledge Discovery from Sensor Data, Sensors Magazine (Cover story), (2006)

[7] J. Gama, M.M. Gaber, Learning from Data Streams, Processing Techniques in Sensor Networks, Springer-Verlag Berlin Heidelberg, (2007)

[8] A. R. Ganguly, O. A. Omitaomu, R. M. Walker, Knowledge Discovery from Sensor Data, for Security Applications, Chapter 12, Processing Techniques in Sensor Networks, Springer (2007)

[9] M. Maksimović, V. Vujović, Comparative analysis of data mining techniques applied on wireless sensor network data for fire detection, Journal of Information Technology and Applications, **3**(2013) 2:65-77

[10] K. Kapitanova et al., Using fuzzy logic for robust event detection in wireless sensor networks, Ad Hoc Netw., (2011)

[11] WEKA data mining tool, Available: www.cs.waikato.ac.nz/ml/WEKA

[12] J. Hühn, E. Hüllermeier, FURIA: An Algorithm For Unordered Fuzzy Rule Induction, Data Mining and Knowledge Discovery, **19**, 293-319, (2009)

[13] I. H. Witten et al., Data mining: practical machine learning tools and techniques, Morgan Kaufmann, Amsterdam, (2011)

[14] J. Han et al., Data Mining: Concepts and Techniques, Morgan Kaufmann, USA, (2012)

[15] J. Natajaran, Simulation of sensor responses of advanced security systems, Master Thesis, University of Texas at Arlington, (2006)