COMPARATIVE ANALYSIS OF DATA MINING TECHNIQUES APPLIED TO WIRELESS SENSOR NETWORK DATA FOR FIRE DETECTION

Mirjana Maksimović[1], Vladimir Vujović[2]
[1]mirjana@etf.unssa.rs.ba, [2]vladimir.vujovic@ etf.unssa.rs.ba

Contribution to the state of the art
DOI: **10.7251/JIT1302065M** UDC: **519.816:004.735**

Abstract: Wireless sensor networks (WSN) are a rapidly growing area for research and commercial development with very wide range of applications. Using WSNs many critical events like fire can be detected earlier to prevent loosing human lives and heavy structural damages. Integration of soft computing techniques on sensor nodes, like fuzzy logic, neural networks and data mining, can significantly lead to improvements of critical events detection possibility. 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. Therefore, 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. This paper focuses on comparative analysis of various data mining techniques and algorithms and in that purpose three different experiments on WSN fire detection data are proposed and performed. The primary goal was to see which of them has the best classification accuracy of fuzzy logic generated data and is the most appropriate for a particular application of fire detection.

Keywords: Analysis, Data Mining, Fire Detection, WEKA, WSN

INTRODUCTION

With the advancement in sensors' technology sensor networks are increasingly finding its applications in many domains such as human activity monitoring [14], vehicle monitoring [8], vibration analysis [13], habitat monitoring [15], object tracking [3], environment monitoring [9, 10, 16] including critical events detections [1,17] etc. 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. To detect fire, one or a combination of sensors and a detection algorithm are needed where the sensors might be part of a wireless sensor network (WSN) or work independently [1].

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. This is the reason for exploring novel data mining techniques dealing with extracting knowledge from large continuous arriving data from WSNs [11]. 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. Sensor data mining is a relatively new area but it already reached a certain level of maturity.

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 a collection of automated and semi-automated techniques for modeling relationships and uncovering hidden patterns in large data repositories. It draws upon ideas from diverse disciplines such as statistics, machine learning, pattern recognition, database systems, information theory, and artificial intelligence [18]. 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 (Fig. 1).

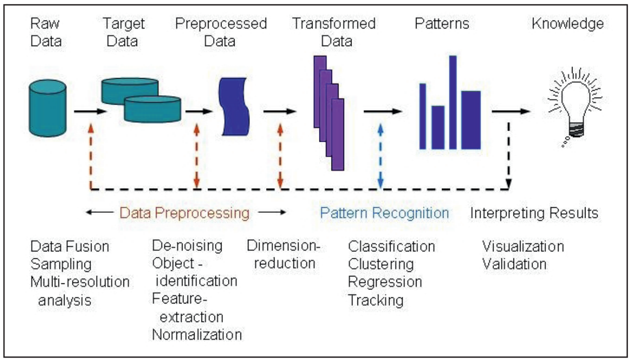


Figure 1 The overall process of knowledge discovery from data includes data preprocessing, data mining, and postprocessing of the data mining results

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 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 [4].

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 [11]. Using these prediction models the number of sensors that need to report their measurements is reduced by reducing both node activity and bandwidth. From analysis made in [11] it is observed that the techniques intended for mining sensor data at network side are helpful for taking real time decision as well as serve as prerequisite for development of effective mechanism for data storage, retrieval, query and transaction processing at central side. On the other hand centralized techniques are helpful in generating off-line predictive insights which in turn can facilitate real-time analysis.

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. 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.

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.

FIRE DETECTION – PREPARING THE INPUT FILES

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. 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. 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.

Preparing input for a data mining investigation usually consumes the bulk of the effort invested in the entire data mining process. However, simple application of 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 [22].

In this work three different experiments for fire detection will be presented based on similar approaches given in [2, 7, 19]. For the sake of clarity of machine learning domain the correlated sensor data used for a detection of fire are converted to nominal types [12]. Input data are defined as IF-THEN rules based on heuristic information that mainly comes from expert knowledge of the fire detection systems. 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 rest of the paper, FURIA (Fuzzy Unordered Rule Induction Algorithm) 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.

The three proposed experiment were created with main goal to show how chosen algorithms predicting power depends on number of data and the fire detection method.

In first experiment, detection of fire is based on two heat detectors - fixed heat and rate of rise heat detector [7]. A fixed temperature heat detector utilizes a temperature sensing element which will generate an alarm condition if the temperature within the protected area reaches a predetermined level (e.g. 57 °C, 63 °C, 74 °C or 90 °C) while rate of rise heat detector is a device that responds when the temperature rises at a rate exceeding a predetermined value (e.g. 8.33 °C/min, 9 °C/min or 11 °C/min, according to NFPA 72 standard). Instead of using these crisp values, fuzzy logic proposes use of linguistic variables. Therefore, data obtained from those two temperature detectors 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. 2 and Fig. 3, respectively. 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.

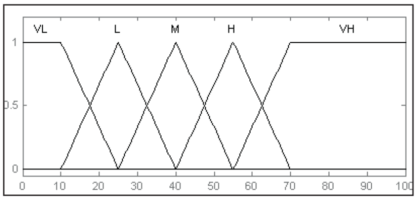


Figure 2 The membership function of input variable

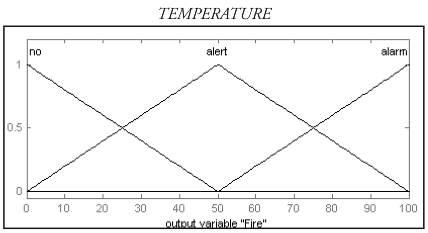


Figure 3 The membership function of input variable *temperature difference*

Possibility of *fire* is defined as output variable and is described with *no*, *alert* and *alarm* linguistic variables as it shown in Fig. 4. This linguistic variable represents the system's confidence in the presence of fire. For example, if the fire confidence is smaller than 50, the probability that there is no fire is higher. If the fire confidence value is higher than 80, there is more than 80% possibility that there is a fire.

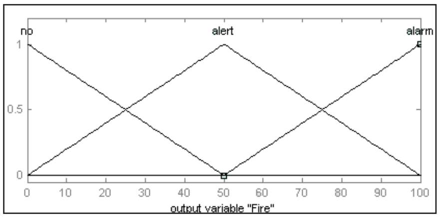


Figure 4 The membership function of output variable *fire*

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

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

Temperature difference	Temperature	Fire (class)
VL	VL	no
L	VL	no
M	VL	no
H	VL	alert
VH	VL	alarm
VL	L	no
L	L	no
M	L	alert
H	L	alert
VH	L	alarm

In the second experiment, detection of fire is based on two successively measured fixed heat temperature detector data [19] in function of additional variable time. Previous and current values of temperature are the same as in Fig. 2. Third input variable time is described with two linguistic variables: short (S) and long (L), according to ºC/min changes (Fig. 5).

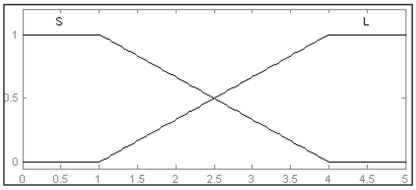


Figure 5 The membership function of input variable *time*

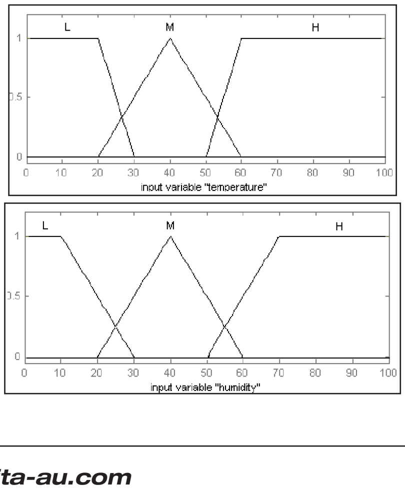
Output variable fire is the same as presented in Fig. 4.

In this case, there are 3 variables and the number of rules in the rule-base is 50 (5*5*2). Table 2 shows first 10 rules of the second experiment.

Table 2 The 2nd fire data test (first 10 rules)

Previous temperature	Current temperature	time	Fire (class)
VL	VL	S	no
VL	VL	L	no
VL	L	S	no
VL	L	L	no
VL	M	S	alert
VL	M	L	no
VL	H	S	alert
VL	H	L	alert
VL	VH	S	alarm
VL	VH	L	alert

The third experiment considers that fire detection is not based only on the temperature values but also on the CO, humidity and light levels, similar as in [2]. Therefore, proposed fire detection logic in this case takes four linguistic variables as input – *temperature*, *humidity*, *light* and *CO*. The linguistic values for all four input variables are classified into *low* (L), *medium* (M), and *high* (H) (Fig. 6). Output variable *fire* is the same as in previous two experiments.



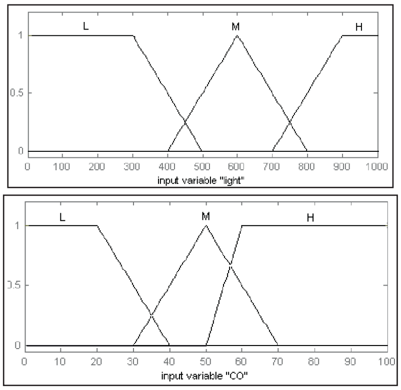


FIGURE 6 THE MEMBERSHIP FUNCTIONS OF INPUT VARIABLES *TEMPERATURE, HUMIDITY, LIGHT* AND *CO*

With 4 variables each of which can take 3 values, the number of rules in the rule-base is 81 (3*3*3*3). Table 3 shows first 10 rules for third fire detection scenario.

TABLE 3 THE 3RD FIRE DATA TEST (FIRST 10 RULES)

Temperature	Humidity	Light	CO	Fire (class)
L	L	L	L	no
L	L	L	M	alert
L	L	L	H	alert
L	L	M	L	no
L	L	M	M	alert
L	L	M	H	alarm
L	L	H	L	no
L	L	H	M	alert
L	L	H	H	alarm
L	M	L	L	no

For further analysis Excel .csv data files are formed based on data given in Tables 1, 2 and 3. The next step is their exporting to WEKA data mining tool [20] in order to apply chosen classification algorithms presented in the rest of the paper.

CLASSIFICATION ALGORITHMS IMPLEMENTATIONS

Implementations of chosen classification algorithms are performed in WEKA, which is a collection of machine learning algorithms for data mining tasks. The algorithms in WEKA can be applied directly to a 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. The purpose is to apply the learning algorithms and then to choose the best one for prediction purposes [21].

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.

The output of the simulator proposed in this paper is used to learn the difference between a subject that is *no*, *alert* and *alarm*. 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 [21].

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

$$P = \frac{TP}{TP + FP} \quad (1) \qquad R = \frac{TP}{TP + FN} \quad (2)$$

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 [21].

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 [21]. 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.

In simulation process presented in this paper four widely used classification algorithms [21] are implemented for comparative analysis with FURIA on given fire data sets. Thus, the comparative analysis is based on following algorithms:

- FURIA
- OneR
- J48 decision tree
- Naive Bayes
- Neural Network classifier

FURIA

FURIA (Fuzzy Unordered Rule Induction Algorithm) is a fuzzy rule-based classification method proposed in 2009 by Hühn and Hüllermeier [6]. FURIA 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 [6].

OneR

OneR is classifier with one parameter – the minimum bucket size for discretization. It generates a one-level decision tree expressed in the form of a set of rules that all test one particular attribute. OneR is a simple, cheap method that often comes up with quite good rules for characterizing the structure in data. In any event, it is always a good plan to try the simplest things first. The idea of OneR is to make rules that test a single attribute and branch accordingly. Next step is to use the class that occurs most often in the training data and to determine the error rate of the rules counting the errors that occur on the training data (the number of instances that do not have the majority class) [21].

Pseudocode of OneR algorithm is:

```
For each attribute,
    For each value of that attribute, make a rule as follows:
        count how often each class appears
        find the most frequent class
        make the rule assign that class to this attribute value.
    Calculate the error rate of the rules.
Choose the rules with the smallest error rate.
```

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 [21].

Naïve Bayes

The Naïve Bayes classifier, 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. A simple Bayesian classifier, Naïve Bayes Classifier (based on Bayes' theorem.), 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 [5]. Naïve Bayes gives a simple approach, with clear semantics, for representing, using, and learning probabilistic knowledge and it can achieve impressive results [21].

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 [21].

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

COMPARATIVE ANALYSIS OF SIMULATION RESULTS

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.

1st experiment

Attributes evaluation of data used in 1st experiment are shown in Table 4.

TABLE 4. ATTRIBUTES EVALUATION – 1ST EXPERIMENT

Attribute	*InfoGainAttributeEval*	GainRatioAttributeEval
Temperature	0.248	0.107
Temperature difference	0.602	0.259

Applying FURIA classifier to existing rules shown in Table 1, 25 rules are generalized into only 3 (Table 5).

TABLE 5 THE FIRE DATA TEST OBTAINED USING FURIA IN 1ST EXPERIMENT

Temperature difference	**Temperature**	**Fire (class)**
VL	/	no
VH	/	alarm
/	VH	alarm

J48 decision tree for presented fire data in 1st experiment is shown in Fig. 7. The attribute with the maximum gain ratio, as it is showed in Table 4, is *temperature difference* and it is selected as the splitting attribute.

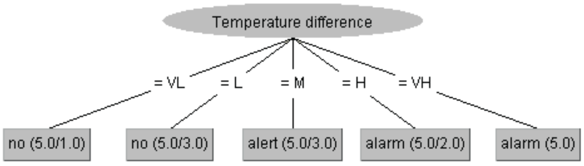


FIGURE 7 J48 DECISION TREE – 1ST EXPERIMENT

Classifiers evaluation is presented in Table 6.

TABLE 6 CLASSIFIER EVALUATION – 1ST EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	60	40	0.6	0.314	0.51	0.6	0.506	0.704
OneR	40	60	0.4	0.297	0.4	0.4	0.395	0.552
J48	40	60	0.4	0.297	0.4	0.4	0.395	0.682
NB	48	52	0.48	0.266	0.436	0.48	0.455	0.661
NN	56	44	0.56	0.25	0.516	0.56	0.531	0.731

From Table 4 it can be seen that FURIA has the best prediction model. It generated a model with 60% correctly classified instances (CCI), a precision of 51% (0.51) and the classification above the ROC curve area (0.704> 0.5).

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 [21]. The results are shown in Table 7.

Table 7 Confusion matrices – 1st experiment

FURIA			
Predicted class			
a	b	c	Real class
4	0	3	*a=no*
0	**0**	7	*b=alert*
0	0	**11**	*c=alarm*

OneR			
Predicted class			
a	b	c	Real class
5	2	0	*a=no*
3	**0**	4	*b=alert*
1	5	**5**	*c=alarm*

J48			
Predicted class			
a	b	c	Real class
5	2	0	*a=no*
3	**0**	4	*b=alert*
1	5	**5**	*c=alarm*

Naïve Bayes			
Predicted class			
a	b	c	Real class
5	2	0	*a=no*
3	**0**	4	*b=alert*
1	3	**7**	*c=alarm*

Neural Network			
Predicted class			
a	b	c	Real class
5	2	0	*a=no*
1	**1**	5	*b=alert*
2	1	**8**	*c=alarm*

Applying Resample filter on data given in Table 1, the balance of data distribution is significantly improved what affect the results of the applied algorithms. In other words, it is possible to generate model with more precise predictions. Results obtained by applying above mentioned algorithms on re-sampled data are shown in next tables. Table 8 shows the predictive accuracy of the classifier on the re-sampled data. From Table 8 it can be seen that J48 decision tree and OneR classifiers have the best prediction models. On re-sampled data they generated a model with 80% correctly classified instances (CCI) and precision of 82.8% (0.828).

Confusion matrices of re-sampled data are presented in Table 9.

Table 9 Confusion matrix of re-sampled data – 1st experiment

FURIA			
Predicted class			
a	b	c	Real class
9	1	0	*a=no*
6	**0**	1	*b=alert*
3	1	**4**	*c=alarm*

OneR			
Predicted class			
a	b	c	Real class
10	0	0	*a=no*
2	**5**	0	*b=alert*
1	2	**5**	*c=alarm*

J48			
Predicted class			
a	b	c	Real class
10	0	0	*a=no*
2	**5**	0	*b=alert*
1	2	**5**	*c=alarm*

Naïve Bayes			
Predicted class			
a	b	c	Real class
10	0	0	*a=no*
3	**3**	1	*b=alert*
1	2	**5**	*c=alarm*

Neural Network			
Predicted class			
a	b	c	Real class
10	0	0	*a=no*
2	**2**	3	*b=alert*
0	1	**7**	*c=alarm*

Table 8 Classifier evaluation on re-sampled data - 1st experiment

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	52	48	0.52	0.29	0.456	0.52	0.454	0.584
OneR	80	20	0.8	0.111	0.828	0.8	0.794	0.844
J48	80	20	0.8	0.111	0.828	0.8	0.794	0.826
NB	72	28	0.72	0.157	0.72	0.72	0.702	0.871
NN	76	24	0.76	0.138	0.784	0.76	0.76	0.80

From the results presented above it can be concluded that FURIA has the best prediction power on initial model of 1st experiment while on re-sampled data OneR and J48 have shown the highest predicting percentage.

2nd experiment

Attributes evaluation of data presented in 2nd experiment are shown in next table.

Table 10. Attributes evaluation – 2nd experiment

Attribute	*Info Gain Attribute Eval*	Gain Ratio Attribute Eval
Previous temperature	0.0388	0.0167
Current temperature	1.0114	0.4356
time	0.043	0.043

Presented results show that the major impact to output variable (*fire*) has *current temperature* value.

Applying FURIA classifier to existing rules shown in Table 2, 50 rules are generalized into 7 presented in Table 11.

Table 11. The fire data test obtained using FURIA in 2nd experiment

Previous temperature	Current temperature	time	Fire (class)
/	VL	/	no
/	L	/	no
/	M	L	no
/	H	L	alert
/	M	S	alert
/	VH	/	alarm
/	H	S	alarm

J48 decision tree for presented fire data in 2nd experiment is shown in Fig. 8. The attribute with the maximum gain ratio, as it is showed in Table 10, is *current temperature* and it is selected as the splitting attribute.

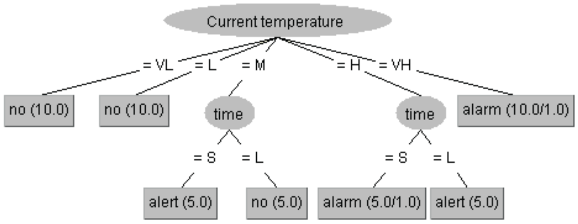


Figure 8 J48 decision tree – 2nd experiment

Classifiers evaluation is presented in Table 12.

Table 13. Confusion matrices – 2nd experiment

FURIA			
Predicted class			
a	b	c	Real class
25	0	0	*a=no*
0	**10**	2	*b=alert*
0	0	**13**	*c=alarm*

OneR			
Predicted class			
a	b	c	Real class
22	3	0	*a=no*
5	**6**	1	*b=alert*
0	4	**9**	*c=alarm*

J48			
Predicted class			
a	b	c	Real class
25	0	0	*a=no*
0	**10**	2	*b=alert*
0	0	**13**	*c=alarm*

Naïve Bayes			
Predicted class			
a	b	c	Real class
24	1	0	*a=no*
4	**4**	4	*b=alert*
0	4	**9**	*c=alarm*

Neural Network			
Predicted class			
a	b	c	Real class
25	0	0	*a=no*
0	**12**	0	*b=alert*
0	0	**13**	*c=alarm*

Table 12. Classifiers evaluation – 2nd experiment

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	96	4	0.96	0.014	0.965	0.96	0.96	0.97
OneR	74	26	0.74	0.151	0.752	0.74	0.742	0.794
J48	96	4	0.96	0.014	0.965	0.96	0.96	0.96
NB	74	26	0.74	0.14	0.715	0.74	0.724	0.936
NN	100	0	1	0	1	1	1	1

Table 14 Classifiers evaluation on re-sampled data – 2nd experiment

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	92	8	0.92	0.031	0.92	0.92	0.92	0.941
OneR	92	8	0.92	0.033	0.923	0.92	0.919	0.944
J48	98	2	0.98	0.007	0.981	0.98	0.98	0.981
NB	90	10	0.9	0.041	0.907	0.9	0.898	0.964
NN	98	2	0.98	0.007	0.981	0.98	0.98	0.989

Results presented in Tables 12 and 13 show that Neural network classifier has the best prediction model. It generated a model with 100% correctly classified instances (CCI).

Results obtained by applying above mentioned algorithms on re-sampled data are given in Tables 14 and 15.

Table 15. Confusion matrices on re-sampled data – 2nd experiment

FURIA			
Predicted class			
a	b	c	Real class
22	0	0	*a=no*
0	**13**	2	*b=alert*
0	2	**11**	*c=alarm*

OneR			
Predicted class			
a	b	c	Real class
22	0	0	*a=no*
0	**14**	1	*b=alert*
0	3	**10**	*c=alarm*

J48			
Predicted class			
a	b	c	Real class
22	0	0	*a=no*
0	**14**	1	*b=alert*
0	0	**13**	*c=alarm*

Naïve Bayes			
Predicted class			
a	b	c	Real class
22	0	0	*a=no*
0	**14**	1	*b=alert*
0	4	**9**	*c=alarm*

Neural Network			
Predicted class			
a	b	c	Real class
22	0	0	*a=no*
0	**14**	1	*b=alert*
0	0	**13**	*c=alarm*

From obtained results it can be seen that Neural Network classifier and J48 decision tree generate the best prediction models on initial and on re-sample data but it is important to note that in this case, other algorithms also have high predictive power.

3rd experiment

Attributes evaluation of data presented in 3rd experiment are shown in Table 16.

Table 16. Attributes evaluation – 3rd experiment

Attribute	Info Gain Attribute Eval	Gain Ratio Attribute Eval
Temperature	0.21741	0.13717
Humidity	0.0042	0.00265
Light	0.08299	0.05236
CO	0.38509	0.24296

Presented results show that the major impact to output variable (*fire*) has *CO* and *temperature*.

Applying FURIA classifier to existing rules shown in Table 3, 81 rules are generalized into 13 presented in Table 17.

Table 17 The fire data test obtained using FURIA in 3rd experiment

Temperature	Humidity	Light	CO	Fire (class)
L	L	/	L	no
M	/	L	L	no
L	M	/	L	no
L	H	L	/	no
L	/	/	M	alert
/	/	M	L	alert
H	/	L	L	alert
M	/	H	L	alert
M	/	L	M	alert
/	/	/	H	alarm
H	/	/	M	alarm
M	/	H	M	alarm
H	/	H	/	alarm

J48 decision tree for presented fire data is shown in Fig. 9. The attribute with the maximum gain ratio, as it is showed in Table 1, is *CO* and it is selected as the splitting attribute.

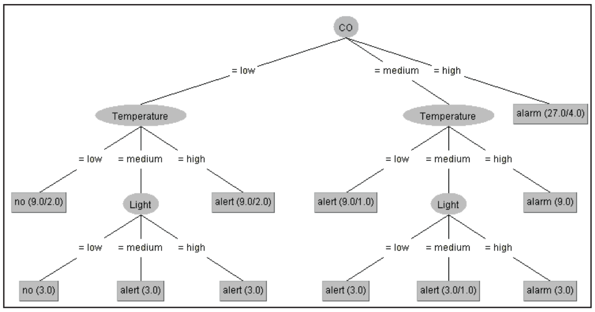


FIGURE 9 J48 DECISION TREE – 3^{RD} EXPERIMENT

Classifiers evaluation is presented in Table 18.

In case of third experiment, Neural Network classifier has the best prediction model. It generated a model with 85.1% correctly classified instances (CCI), a precision of 85.9% (0.859) and the classification above the ROC curve area (0.951> 0.5). Confusion matrices are presented in Table 19.

TABLE 19 CONFUSION MATRICES – 3^{RD} EXPERIMENT

FURIA			
Predicted class			
a	b	c	Real class
8	3	1	*a=no*
2	**16**	13	*b=alert*
0	3	**35**	*c=alarm*

OneR			
Predicted class			
a	b	c	Real class
0	10	2	*a=no*
0	**19**	12	*b=alert*
0	15	**23**	*c=alarm*

J48			
Predicted class			
a	b	c	Real class
9	2	1	*a=no*
0	**24**	5	*b=alert*
0	4	**34**	*c=alarm*

Naïve Bayes			
Predicted class			
a	b	c	Real class
3	8	1	*a=no*
0	**25**	6	*b=alert*
0	6	**32**	*c=alarm*

Neural Network			
Predicted class			
a	b	c	Real class
9	3	0	*a=no*
1	**27**	3	*b=alert*
0	5	**33**	*c=alarm*

Table 20 shows the predictive accuracy of the classifier on the re-sampled data. Obtained results show that all classifiers applied on re-sampled data have significantly better accuracy compared to results presented in Table 1. From Table 20 it can be seen that Neural Network classifier again has the best prediction model. On re-sampled data it generated a model with 93.8% correctly classified instances (CCI), a precision of 94% (0.94) and the classification above the ROC curve area (0.997> 0.5).

Results shown in confusion matrices of re-sampled data in Table 21 are also better than ones presented in Table 19.

TABLE 21 CONFUSION MATRICES OF RE-SAMPLED DATA – 3^{RD} EXPERIMENT

FURIA			
Predicted class			
a	b	c	Real class
14	2	2	*a=no*
1	**13**	5	*b=alert*
0	2	**42**	*c=alarm*

OneR			
Predicted class			
a	b	c	Real class
14	1	3	*a=no*
9	**10**	0	*b=alert*
2	8	**34**	*c=alarm*

J48			
Predicted class			
a	b	c	Real class
14	1	3	*a=no*
2	**15**	2	*b=alert*
0	2	**42**	*c=alarm*

Naïve Bayes			
Predicted class			
a	b	c	Real class
12	6	0	*a=no*
2	**15**	2	*b=alert*
0	5	**39**	*c=alarm*

TABLE 18 CLASSIFIERS EVALUATION – 3^{RD} EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	72.8	27.1	0.728	0.203	0.732	0.728	0.716	0.847
OneR	51.8	48.1	0.519	0.344	0.457	0.519	0.482	0.587
J48	82.7	17.3	0.827	0.116	0.826	0.827	0.828	0.832
NB	74.1	25.9	0.741	0.184	0.778	0.741	0.723	0.872
NN	85.1	14.8	0.852	0.096	0.859	0.852	0.853	0.951

Neural Network			
Predicted class			
a	b	c	Real class
17	1	0	*a=no*
1	**17**	1	*b=alert*
0	2	**42**	*c=alarm*

Table 20 Classifier evaluation on re-sampled data – 3rd experiment

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	85.1	14.8	0.852	0.121	0.852	0.852	0.849	0.947
OneR	71.6	28.4	0.716	0.117	0.747	0.716	0.724	0.8
J48	87.6	12.3	0.877	0.092	0.875	0.877	0.875	0.929
NB	81.5	18.5	0.815	0.078	0.843	0.815	0.822	0.95
NN	93.8	6.1	0.938	0.03	0.94	0.938	0.939	0.997

Obtained results show that Neural Network classifier generates the best prediction models on initial and on re-sample data.

CONCLUSION

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. The main purpose of sensors network for fire detection is to collect the monitoring original data, and provide basic information and decision support for monitoring centre. Also, data mining algorithm has to be sufficiently fast to process high-speed arriving data. In sensor networks, data are distributed by nature. The sensor scenario may often require in-network processing, wherein the data is processed to higher level representations before further processing. In other words, 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. 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 in a case of fire and to see which of applied techniques has the best prediction performances in order to reduce node activity and bandwidth.

FURIA was used as a base prediction model and it has shown the best prediction power in initial model of 1st experiment while on re-sampled data OneR and J48 obtained the highest predicting percentage. Neural Network classifier and J48 decision tree generated the best prediction models on initial and on re-sample data in 2nd experiment where all algorithms have shown high predictive power. Obtained results in 3rd experiment show that Neural Network classifier generates the best prediction models on initial and on re-sample data.

It can be seen that Neural Network classifier showed better predicting power on larger data set while in the case of small data set, simpler classifier like OneR or FURIA showed quite good results. 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.

Our future work will be based on measuring and combining real data from different sensors (temperature, humidity, light and CO) 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 (*alert* and *alarm*) to the central server. This process by reducing the number of inputs, deleting redundancy, and improving the system speed and correct rate would decrease the potential network traffic and prolong network life span making early fire detection possible.

Authorship statement

Author(s) confirms that the above named article is an original work, did not previously published or is currently under consideration for any other publication.

Conflicts of interest

We declare that we have no conflicts of interest.

LITERATURE

[1] Bahrepour, M., et al. (2009). Use of AI Techniques for Residential Fire Detection in Wireless Sensor Networks, AIAI-2009 Workshops Proceedings, pp. 311-321, Thessaloniki, Greece.

[2] Bolourchi, P. and Uysal, S. (2013). Forest Fire Detection in Wireless Sensor Network Using Fuzzy Logic, Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), Madrid, Spain.

[3] Chauhdary, S.H., et al. (2009). EOATR: Energy efficient object tracking by auto adjusting transmission range in wireless sensor network, *Journal of Applied Sciences*, 9(24), 4247-4252.

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

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

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

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

[8] Kargupta, H., et al. (2010). A Vehicle Data Stream Mining System for Ubiquitous Environments, *Ubiquitous Knowledge Discovery Lecture Notes in Computer Science*, 6202:235-254.

[9] Kumar, D. (2011). Monitoring forest cover changes using remote sensing and GIS: A global prospective, *Research Journal of Environmental Sciences*, 5 (2), 105-123.

[10] Lee, L.T. and Chen, C.W. (2008). Synchronizing sensor networks with pulse coupled and cluster based approaches, *Information Technology Journal*, 7(5), 737-745.

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

[12] Manjunatha, P., et al. (2008). Multi-Sensor Data Fusion in Cluster based Wireless Sensor Networks Using Fuzzy Logic Method, IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, India.

[13] McGarry, K. and MacIntyre, J. (2001). Data mining in a vibration analysis domain by extracting symbolic rules from RBF neural networks, Proceedings of the 14th International Congress on Condition Monitoring and Engineering Management, Manchester, U.K.

[14] Perkowitz, M., et al. (2004). Mining models of human activities from the Web, Proceedings of the 13th International Conference on World Wide Web, New York, NY.

[15] Rozyyev, A., et al. (2011). Indoor child tracking in wireless sensor network using fuzzy logic technique, *Research Journal of Information Technology*, 3(2), 81-92.

[16] Sabri, N., et al. (2011). Wireless sensor actor network based on fuzzy inference system for greenhouse climate control, *Journal of Applied Sciences*, 11(17), 3104-3116.

[17] Sathik, M.M., et al. (2010). Fire Detection Using Support Vector Machine in Wireless Sensor Network and Rescue Using Pervasive Devices, *International Journal of Advanced Networking and Applications*, 02(02), 636-639.

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

[19] Tripathy, M.R., et al. (2010). Energy Efficient Fuzzy Logic Based Intelligent Wireless Sensor Network, Progress In Electromagnetics Research Symposium Proceedings, Cambridge, USA.

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

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

[22] Yabukil, N., et al. (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.

Submitted: October 29, 2013.
Accepted: December 9, 2013.