IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 10, 2021, accepted May 2, 2021, date of publication May 12, 2021, date of current version May 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3079702

Border Trespasser Classification Using Artificial Intelligence

MOHSEN OTHMANI[1], MOHAMED HECHMI JERIDI[1], QING-GUO WANG[2], AND TAHAR EZZEDINE[1]

[1]Communication System Laboratory (SysCom), National Engineering School of Tunis, University of Tunis El Manar, Tunis 1002, Tunisia
[2]Institute of Artificial Intelligence and Future Networks, Beijing Normal University at Zhuhai, BNU-HKBU United International College, Zhuhai 519087, China

Corresponding author: Mohsen Othmani (mohsen.othmani@enit.utm.tn)

The work of Qing-Guo Wang was supported in part by the BNU Seed Fund, and in part by the UIC Start-up Fund under Grant R72021115.

ABSTRACT Monitoring the border is a very important task for national security. Wireless sensor networks (WSN) appear well suited in this application. This work aims to monitor a large-scale geographical framework that represents the borders of countries. Researchers take the Tunisian Algerian border as an example. This border is labeled by the illegal passage of intruders between the two countries. The task is to identify the intruders and study their kinematics based on speed, acceleration, and bearing. The appropriate types of sensors are determined according to the nature of intruders. Six classification techniques are compared which are: Naïve Bayes, Support Vector Machine (SVM), Multilayer Perceptron, Best First Decision Tree (BF-Tree), Logistic Alternating Decision Tree (LAD-Tree), and J48. The comparison of the performance of the classification techniques is provided in terms of correct differentiation rates, confusion matrices, and the time taken to build each model. Four different levels of cross-validation are used to validate the classifiers. The results indicate that J48 has achieved the highest correct classification rate with a relatively low model-building time.

INDEX TERMS Border surveillance, classification, machine learning, sensing, wireless sensor networks.

I. INTRODUCTION

Monitoring of border areas can be realized with WSN. In fact, due to the low cost, no maintenance convenience, absence of complex equipment requirements, etc..., the WSN is very suitable for monitoring in poor conditions or large area. The users can avoid wiring, network maintenance difficulties, waste of resources, etc.

Due to the speedy evolution of technology, wireless sensor networks have emerged as a promising research platform. A plethora of high-reliability fields such as health condition monitoring [1], Surveillance System for Railway Traffic [2], monitoring in secure assisted living internet of things environments [3], machinery manufacturing [4], mine safety monitoring requirements [5] are introducing the wireless sensor network. Besides, the recent growing emergence of the internet of things (IoT) and the importance of position in this context, has activated localization systems which become an essential part of life. The most proposed localization techniques for sensor networks use the geometric properties of the sensor network based on triangulation methods to deduce the locations of the sensors [6]–[10]. Another approach based on machine learning was adopted by several authors to resolve the localization problem. They work directly on the natural (non-Euclidean) coordinate systems provided by the sensor devices [11], [12]. One of the methods used is classification. According to Nguyen *et al.* [13], the localization problem can be modeled as a classification problem. They define a strategy based on three steps to achieve the localization of the sensor nodes:

- Define a set of classes {C1, C2...}, where each class Ci being a geographic region in the area of the sensor network.
- Determine the Training Data that will be used as training data for the classification procedure on class Ci ...
- Finally, execute the classification procedure to obtain a prediction model. This model is used to estimate for each given sensor node S and class Ci the membership of S in class Ci

In this study, the region of interest in the general framework is the border zone between neighboring countries, particularly, researchers focus on the border area between Tunisia

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchun Chen.

and Algeria as an example. This zone is labeled by the illegal passage of intruders between both countries. In fact, this borderline is a mountainous and forested area, and full of wild animals (such as the fox, the wolf, the pigs…) represents the preferred path for those who want to penetrate and smuggle from one country to another in an irregular way and especially for terrorists.

The Investigations conducted by the border police on this subject, as well as statistical information and military interventions on the ground, have shown that target people, who take the Tunisian-Algerian border illegally, are generally carrying ferrous weapons, move frequently at night using torches and they take animals for contraband purposes (ACP), such as donkeys, as a means of transportation to smuggle weapons and ammunition because of the difficulties of the geographical features of the area. Thus, the intruder object could be an unarmed person, a person carrying aferrous weapon (soldier), an ACP that transport some weapons, unarmed/armed person that uses a torch at night (the detection of light emitted by the torch will not be discussed in this paper).

Border surveillance based on wireless sensor networks can be an appropriate solution. However, it is a very critical task that requires some steps to accomplish its mission.

The remainder of this paper is structured as follows. Section 2 is reserved to define the user's requirements and study the kinematics model of intruders based on speed, acceleration, and bearing. Then, the authors study the sensing phase by determining the appropriate type of sensors that suits the nature of intruders, in section 3. In Section 4, a comparative study is conducted between six classification techniques: Naïve Bayes, SVM, Multilayer Perceptron, BF Tree, LAD Tree, and J48. The comparison of the performance of the classification techniques is provided in terms of correct differentiation rates, confusion matrices, and the time taken to build each model. Four different levels of cross-validation are used to validate the classifiers. Finally, Section 5 concludes the paper.

II. REQUIREMENTS AND PERFORMANCE METRICS

A. REQUIREMENTS

Based on the literature on border surveillance and intrusion detection and some real-world applications like ''Line in the sand'' [14], we set the following requirements:

- For detection phase: The deployed sensors must have a correct high detection probability P_D that will ensure a better estimate of the target presence. However, the time between target presence and its detection must be allowable latency $T_{D.}$
- For the classification phase: we must use the right classification method based on some hypothesis can classify each intruder in its correct category: this translates into the confusion matrix (Section 4) that reflects the probability of classifying an intruder i into his good class i (PCi,i) and vice versa (PCi,j).
- For the tracking phase: locating the intruder is the heart of the monitoring application. Indeed tracking system should give us the accurate coordinates (with acceptable tolerance) of where the target has entered as well as its current position, taking into account its speed and the detection latency.

Table 1 summarizes the acceptable performance requirements based on the realistic military system in the field and the state of the art.

TABLE 1. Performance requirements.

Metric	Value	Description
P_D	>0.95	Probability of detection
P_{FA}	<0.10	Probability of false alarm
T_D	<15	Detection latency (s)
$P_{Ci,j\|i=j}$	Table II	Probability of true classification
$P_{Ci,j\|i\neq j}$	Table II	Probability of misclassification
$(\hat{x}, \hat{y})$	$\in (x, y) \pm (2.5, 2.5)$	Position estimation error (m)

B. KINEMATIC MODEL

Our border monitoring focus on three targets classes: unarmed person, a person wearing metal (Soldier) and an ACP that transport a mass of ferrous.

In what follows, we study the kinematic movement models of targets (Fig. 1.a-c). We assume that each intruder has a specific (uniform, normal) distribution for speed, acceleration, and bearing.

The kinematics of an unarmed person results in a normally distributed speed (PS: N(5,1), Person speed (km/h)) and almost constant walking orientation (PB: N(0,1), Person bearing (rad)), and uniform acceleration (PA: U[−1,1], Person acceleration (m/s^2)). An armed person is likely to change direction frequently (SB: N(02), Soldier bearing (rad)), walking, running (SA: U[−3,3], Soldier acceleration (m/s^2)), or even sometimes crawling. Therefore, the kinematics of a soldier are distinguished by a uniform distribution of speed (SS: U[1,20], Soldier speed (km/h)) with obviously a wider normally distribution bearing than an unarmed person. As for the ACP which transport metals, their kinematics are characterized by an almost constant direction (ACPB: N(0, 1), ACP bearing (rad)) with a greater range of speed (ACPS: U[1,23], ACP speed (km/h) / ACPA: U[−4, 4], ACP acceleration (m/s^2)).

III. SENSING

The sensing phase plays a very important role to build our application to the desired standards. An ideal sensor should have low power dissipation and cost with high sensitivity, accuracy, and repeatability while being easy to use. Unfortunately, it is almost impossible to bring all of these features into one sensor and we have to choose application-specific sensors

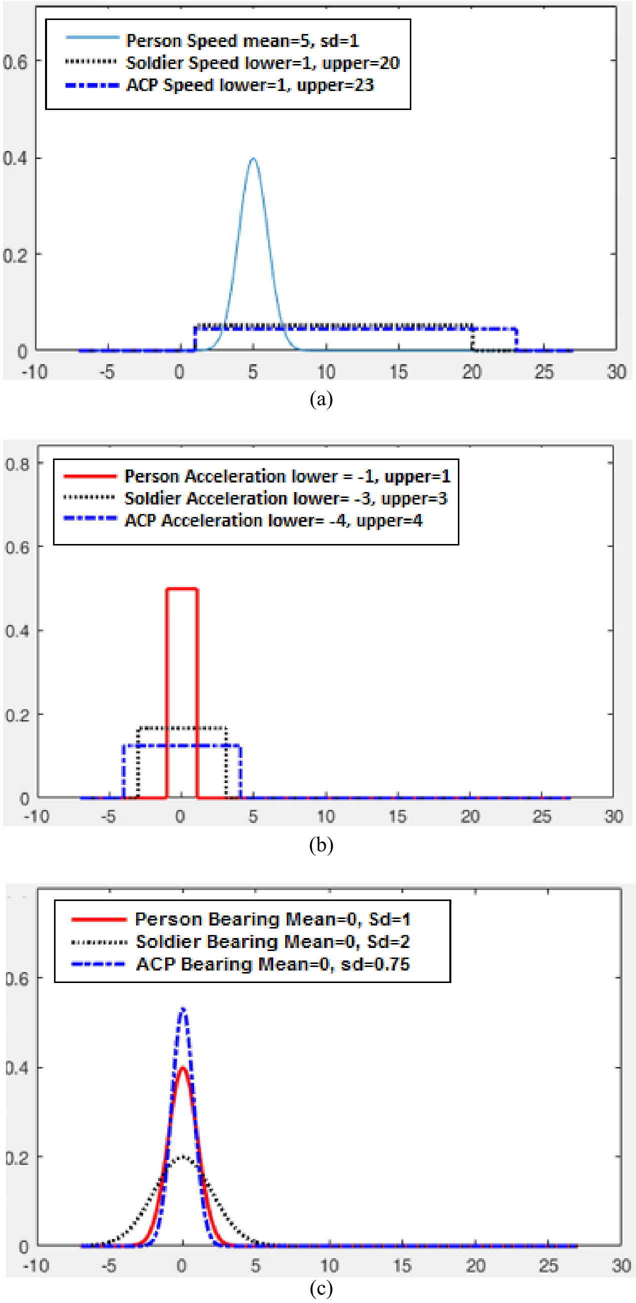


FIGURE 1. a. Targets Speed motion distribution. b. Targets acceleration motion distribution. c. Targets Bearing motion distribution.

in the design phase of WSNs. Based on the study [15] on the design of the detection device, researchers can classify a sensor according to several points of view: Power supply, output type, and measurement properties. Below, the figures which summarize these classifications, help us to choose right ones.

Other sensing principles used in WSNs are based on Physical properties, Motion properties, Contact properties, Presence, Biochemical, and Identification, are summarized in [16]. Also, Table 2 summarizes the strengths and weaknesses of the sensors given each metric. We can select a set of sensors for the detection phase according to the nature of the considered intruders in our context.

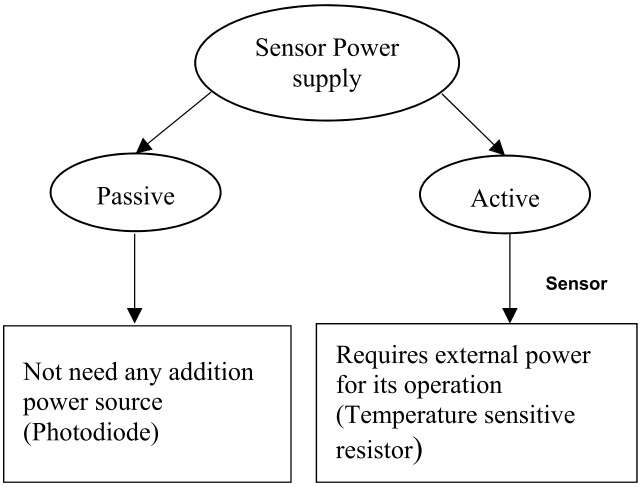

Sensor Power supply
Passive
Active
Sensor
Not need any addition power source (Photodiode)
Requires external power for its operation (Temperature sensitive resistor)

FIGURE 2. Sensor classification: Power supply.

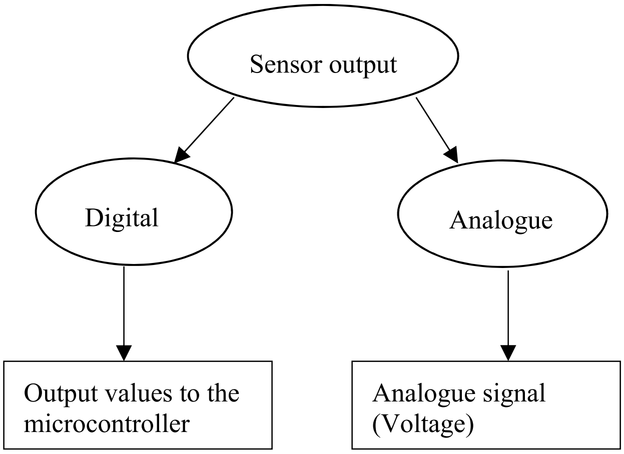

Sensor output
Digital
Analogue
Output values to the microcontroller
Analogue signal (Voltage)

FIGURE 3. Sensor classification: type of sensor output.

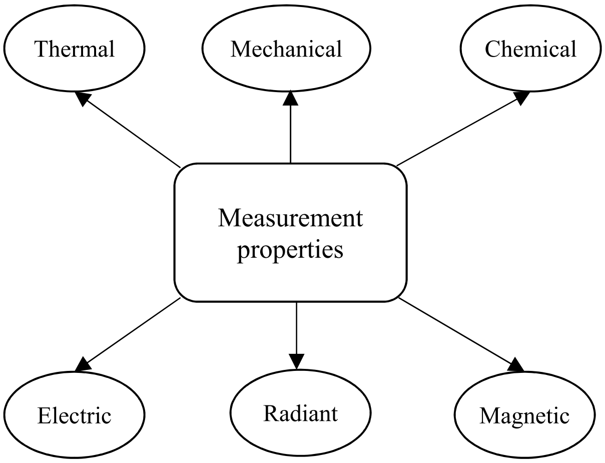


FIGURE 4. Sensor classification: Measurement properties.

Some metrics are summarized by [14] as **M1:** Orientation invariant, **M2:** No special packaging, **M3:** Reasonable signal processing, **M4:** Established, **M5:** Long-Range, **M6:** No line-of-sight, **M7:** Co-locatable and **M8:** Passive operation.

TABLE 2. Sensors ranking by different metrics.

	M1	M2	M3	M4	M5	M6	M7	M8	Total
Optical	NO	NO	NO	OK	OK	NO	OK	OK	4
Seismic	NO	NO	NO	OK	OK	OK	NO	OK	4
Thermal	NO	NO	OK	OK	OK	NO	OK	OK	5
Electric	NO	NO	OK	NO	NO	OK	NO	NO	2
Magnetic	OK	OK	OK	OK	NO	OK	OK	OK	7
Radar	NO	OK	OK	OK	OK	OK	NO	NO	5
Chemical	OK	NO	NO	NO	NO	NO	OK	OK	3

In each cell of Table 2, we write 'OK' for each type of sensor operating in an energy domain and we write 'NO' in the opposite case

To provide an example of a border surveillance scenario (the north-west border of Tunisia with Algeria), the target object may be an armed person, a person wearing a metal tool (a soldier carrying an iron arm), or an ACP transporting a mass of ferrous. From the results of Table (2), we can model in general, the moving ferromagnetic object as a magnetic dipole in motion. Therefore, magnetic sensors are the most powerful and allow us to detect soldiers and ACP. More precisely, we use a passive binary magnetic sensor: a passive sensor because of the risk area deployment in addition to the inability to provide a source of electrical energy. Each node sends just one bit of information about the presence or absence of a target in its sensing range. We don't need any computationally complex time-frequency domain signal processing. As for the unarmed person who is not detected, we choose the radar sensors.

IV. CLASSIFICATION

All classification simulations are made under the software Weka3.8. Weka is free software composed of a set of classes and algorithms written in Java. It implements the main data mining algorithms and it can be installed on different operating systems such as versions for Unix and Windows [17].

Simulations on Weka can be done either through a user interface or through the command line or also through the use of classes provided inside Java programs.

Classification is a data mining algorithm that guides and defines the way adopted to identify the correct class label of a new data instance. Often, the classification procedure allocates a large percentage of the data as a training data set and considers the remaining data for the testing phase. In the context of this case study, the attribute's data are the acceleration, the speed, and the bearing. The objective of the classification model is to predict the correct class label (person, soldier, ACP carrying a mass of iron). We apply 6 classification schemes and we make comparisons between them: Naïve Bayes, Support Vector Machine (SVM), Multilayer Perceptron, BF Tree, LAD Tree, and j48. We use 4500 instances, 4 attributes speed, acceleration, bearing, and intruder.

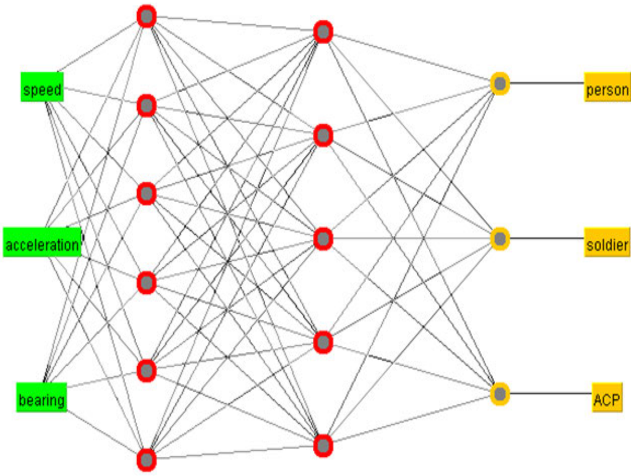


FIGURE 5. Sensor Multi-Layer Perceptron Neural Network (MLPNN).

A. NAÏVE BAYES

A Naive Bayesian classifier is a supervised machine learning algorithm. It is based on a simple assumption of independence between all pairs of variables. It is used in many applications such as text classification: in [18], the authors sought to improve the Naive Bayes by proposing an auxiliary feature method and then adjust the corresponding conditional probability in order to improve classification accuracy. Also, N. F. Rusland *et al.* [19] test the Naïve Bayes algorithm for e-mail spam filtering on two datasets. They proved that the type of email and the number of instances of the dataset influences the performance of Naïve Bayes. Numerous other applications that make Naïve Bayes a distinguished and preferred algorithm among machine learning tools: [20] for Banking and Insurance Domain, [21] for Healthcare Applications, etc...

In this context, the classification problem amounts to estimating the probability of each class of intruders Ci (i = Unarmed person, a soldier, an ACP carrying an iron) knowing a vector of characteristics F (speed, acceleration, bearing).

Applying Bayes' theorem gives us:

$$P\left(c_i \,|\, F\right) = \frac{P\left(c_i\right) P(F|c_i)}{P(F)} \tag{1}$$

```
speed  <  2.43374
|    bearing  <  -1.64715
|    |    speed  <  1.00491
|    |    | acceleration  <  3.05419: soldier (432.0/13.0)
|    |    | acceleration  >=3.05419: ACP (2.0/0.0)
|    |    speed  >=1.00491: ACP (4.0/0.0)
|    bearing  >  -1.64715
|    |    bearing  <  1.26336
|    |    |    speed  <  0.99699
|    |    |    |   speed  < -0.99857: ACP (193.0/0.0)
|    |    |    |   speed  >=  -0.99857
|    |    |    |   |  bearing  <  -1.17276
|    |    |    |   |  | acceleration < -3.0358: ACP (3.0/0.0)
|    |    |    |   |  | acceleration >= -3.0358
|    |    |    |   |  | |   speed <  0.24572
|    |    |    |   |  | |   |  acceleration < 2.89835
|    |    |    |   |  | |   |  |  bearing  <  -1.35932
|    |    |    |   |  | |   |  |    |    |    |    |    |    |    |
|    |    |    |   |  | |   |  |    |    |    |    |    |    |    |
|    |    |    |   |  | |   |  |    |    |    |    |    |    |    |
|    |    |    |   |  | |   |  |    |    |    |    |    |    |    |
|    |    |    |   |  | acceleration >=3.15651 ACP (2.0/0.0)
|    |    |    |   |  speed  >= 0.78152: ACP (2.0/0.0)
|    |    |    |   speed  >=  0.82241
|    |    |    |   |  speed  <   0.95759: soldier (12.0/0.0)
|    |    |    |   |  speed  >=  0.95759: soldier (1.0/1.0)
|    |    |    speed  >= 1.04962: ACP (8.0/0.0)
|    bearing  >=  1.86172
|    |    acceleration  <   3.03331: soldier (360.0/6.0)
|    |    acceleration  >= 3.03331: soldier (3.0/1.0)
speed  >=  2.43374
|    bearing  <   0.74787:  ACP(7.0 / 0.0)
|    bearing  >= 0.74787:  person(1347.0 / 1.0)
```

FIGURE 6. Best first decision tree.

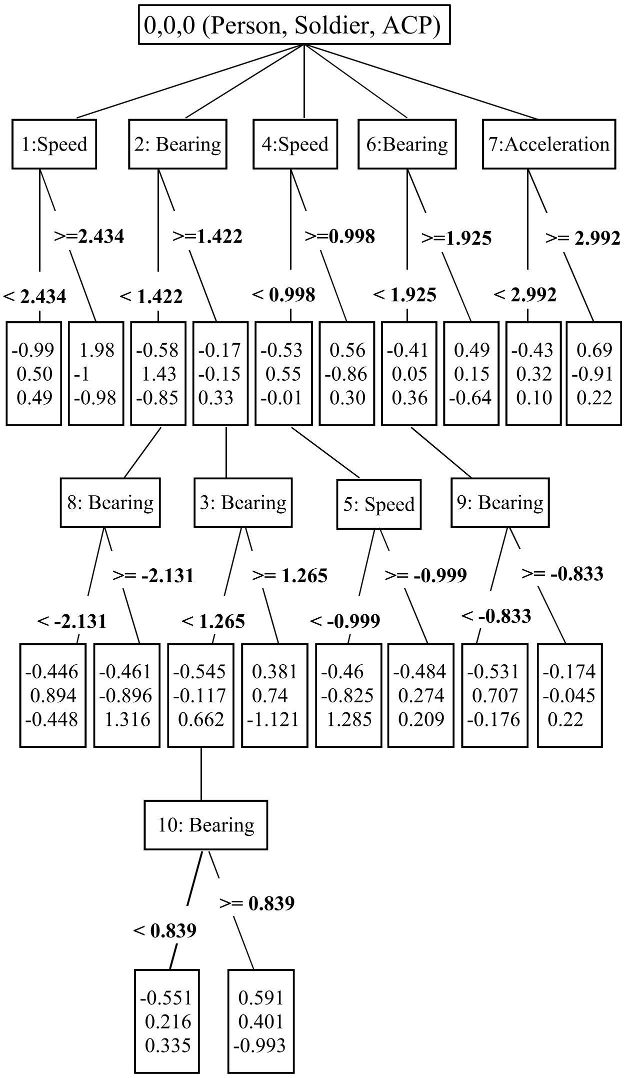


FIGURE 7. Logistic alternating decision tree.

By applying the hypothesis of independence between the variables, we obtain:

$$P\left(c_i \mid F\right) = P\left(c_i\right) \prod\nolimits_{f_i \in F} P(f_i|c_i) \tag{2}$$

The objective is to select the class with the highest probability, knowing the given feature vector. Therefore, the estimated class (c) is the one that maximizes the conditional probability, hence:

$$\mathbf{P}\left(\mathbf{c_i} \mid \mathbf{F}\right) = \underset{\mathbf{c_i}}{\mathbf{argmax}}\, \mathbf{P}\left(\mathbf{c_i}\right) \prod\nolimits_{\mathbf{f_i} \in \mathbf{F}} \mathbf{P}(\mathbf{f_i}|\mathbf{c_i}) \tag{3}$$

B. SUPPORT VECTOR MACHINE (SVM)

A Support Vector Machine (SVM) is a supervised machine learning algorithm, proposed early in the 1990s [22] (Hsu, 2008), that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems. The decision function to solve the classification of two-class SVM problem is:

$$f\left(R\right) = w^T R + b \tag{4}$$

where R and b are the parameters to estimate.

SVM attracts exceptional attention for its ability to easily adapt to multi-class environments, in which data is not linearly separable. Indeed, the need for a multi-label classification has given rise to multi-classes SVM. The attractive aspect of SVMs is introducing the kernel functions which transform the nonlinearly separable case in low dimensional space into the linearly separable one in the higher dimensional space. There are several types of kernels in common use, but not limited to linear kernel, Gaussian kernel, radial base function (RBF) kernel, polynomial kernel, etc.

C. MULTILAYER PERCEPTRON

By default, the Weka environment uses the back-propagation to learn the multilayer perceptron neural network. The activation function for all the nodes are sigmoid with a parameter $k > 0$:

$$\sigma_k\left(x\right) = \frac{\exp(kx)}{\left(\exp\left(kx\right)+1\right)} = \frac{1}{1+\exp(-kx)} \tag{5}$$

Back-propagation is part of the error correction learning rule for multilayer networks. Discovered in the early 80s by Rumelhart and Mc Clelland, the purpose of back-propagation is to partition the responsibility for the error on a multilayer network. The literature shows that neural networks and in particular back-propagation have been remarkably successful in several fields of application. As a mention, but not limited to, the authors [23] proposed a new method for locating nodes with back-propagating neural networks (BP-NN) in order to reduce the localization accuracy problem of the Wireless Sensor Network (WSN). Also in the health sector, the authors [24] have used BP-NN as a classification tool for the segregation of mammography into abnormal and normal.

In a backward propagation network, activation moves forward through the network from one level to the next. Error propagation goes backward in a similar fashion. The network can calculate the error of the output units just like a simple single-layer network. The error of a neuron at level n is a function of the errors of all neurons at level n + 1 which use its output.

For retro propagation networks, a linear input threshold function is not used, but rather a sigmoidal activation function. So the main idea of the back-propagation algorithm is based on two steps: calculate the network error for the first time and then try to minimize it. This method has shown positive results in several problems, however, the major inconvenience is that the convergence takes a long time and that each step requires the calculation over the whole training set.

D. BEST-FIRST DECISION TREES (BFTree) & LOGISTIC ALTERNATING DECISION TREE (LADTree)

Among the most popular classification techniques, we can find decision tree learning: it is distinguished by its speed and acceptable performance. The reading of a decision tree is done on two main axes: the internal nodes represent the choice between the alternatives, while the leaf nodes mark a classification.

The standard decision tree classification process is triggered by placing a root node (in our case: person, soldier, ACP) then we divide the learning instances into as many subsets as branches extending from the root node. This step is repeated for a chosen branch. Generally, we use a fixed order, from left to right, to develop the nodes. In return, to do this, the BFTree method adopts the best order (The ''best'' in the sense of the node which minimizes the impurities among all the nodes available for the division).

Decision trees are simple classifiers to implement and to understand. On the other hand, they are distinguished by a lower accuracy performance than other classifiers. To overcome this drawback, a reinforcement strategy was adopted. Hence the birth of the alternate decision tree (ADTree): a classifier that combines a boosting algorithm and a decision tree. To increase the applicability of ADTree, the authors propose in [25] a new algorithm - multivariate ADTree. It presents and discusses its different variants (Fisher's ADTree, Sparse ADTree, and Regularized Logistic ADTree). To induce ADTree, we apply a logistic boosting algorithm to it, which gives birth to a new variety: LADTree. The LADTree classifier generates a multi-class alternating decision tree [26]. At each iteration, a single attribute is chosen as a separator node. The results of each learning instance are stored by class. Then, the final model is the result of the fusion of all the trees relating to the different classes.

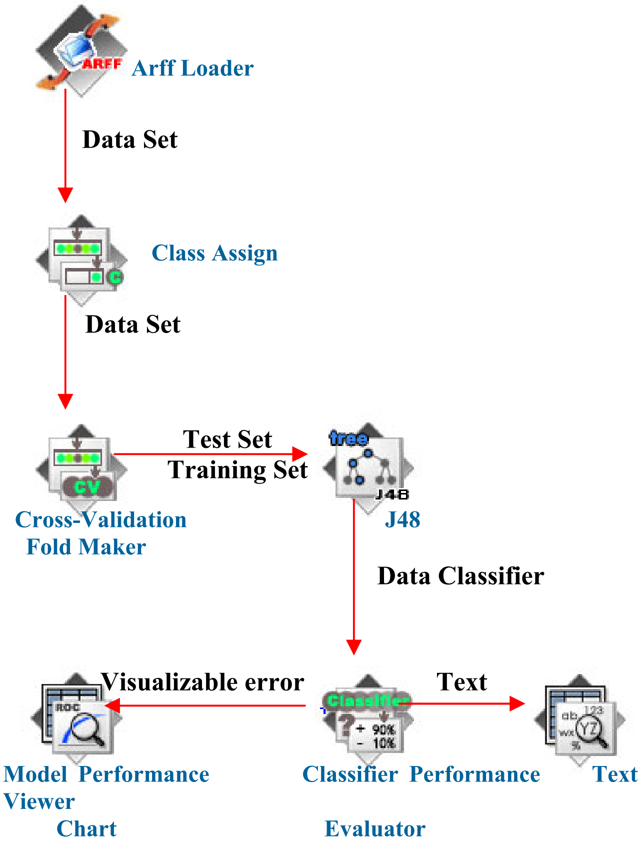


FIGURE 8. Knowledge Flow for J48.

E. J48

J48 is a univariate decision tree, it is an extension of the ID3 algorithm and creates a small tree used for classification [27]. It uses the divide and conquer approach to grow decision trees [28]. Besides and to reduce classification errors, the J48 algorithm is characterized by the use of an improved tree pruning technique. In fact, because of outliers, and possible over-adjustment, pruning represents an essential recourse in the creation of trees. Figure 8 presents the knowledge Flow for the J48 Algorithm: it offers a data-flow inspired interface with some features as process data in batches or incrementally, process multiple batches or streams in parallel, view models produced by classifiers for each fold in cross-validation, etc…

The basic algorithm to construct the J48 decision tree takes the following steps [29].

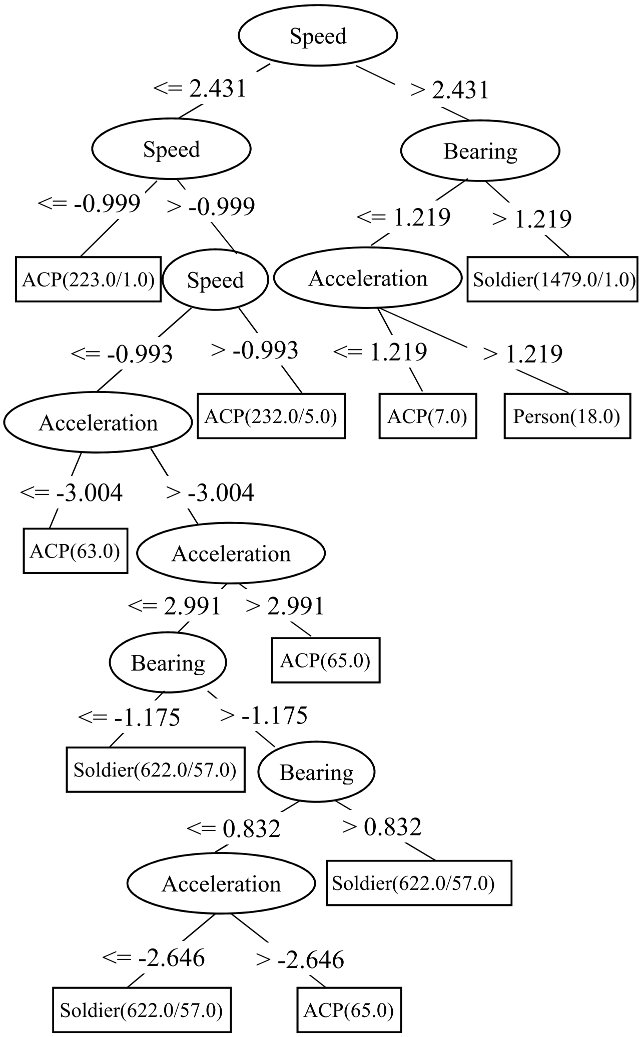


FIGURE 9. J48 Tree.

TABLE 3. Classifiers evaluation summary: 5 fold cross-validation.

Scheme	Naïve Bayes	SVM	MLP	BF Tree	LAD Tree	J48
Correctly Classified Instances %	87,244	87,400	86,911	86,867	87,289	87,444
Incorrectly Classified Instances %	12,756	12,600	13,089	13,133	12,711	12,556
Mean absolute error	0,122	0,084	0,125	0,118	0,118	0,119
Root mean squared error	0,243	0,290	0,251	0,253	0,239	0,248
Relative absolute error %	27,363	18,900	28,187	26,545	26,612	26,744
Root relative squared error	51,567	61,482	53,241	53,699	50,744	52,513
Time taken to build model (s)	0,010	1,340	2,540	0,880	0,440	0,050

Step 1: Calculate the Entropy of the training set S:

$$E\left(S\right)=-\sum\nolimits_{i=1}^{C}P_i\log_2 P_i \tag{6}$$

where C is the number of classes, and Pi is the probability of randomly picking an element of class i. In general, entropy can be considered as the degree of variance of the data.

Step 2: Now, to quantify the quality of a split, we calculate the Information Gain X(S) for test attribute X to partition.

$$\text{Information GainX (S)} = \text{E (S)} - \sum\nolimits_{i=1}^{M}\frac{|S_i|}{|S|}E(S_i) \tag{7}$$

where Si is a subset of S corresponding to i$^{\text{th}}$ output. M is the number of branches (subsets). $\sum_{i=1}^{M}P_iE\left(S_i\right)$ determine the quality of the split by weighting the entropy of each branch by how many elements it has. Authors can interpret that Information Gain reflects how much entropy they removed. So, the Information Gain is calculated for a split by subtracting the weighted entropies of each branch from the original entropy. The best split is chosen by maximizing Information Gain and then will be selected as the threshold.

In case the instances belong to the same class, the tree represents a leaf.

Step 3: In this step, we calculate the partition information value Split Info(X):

$$\text{SplitInfo}\left(X\right)=-\sum\nolimits_{i=1}^{M}\left[\frac{|S_i|}{|S|}\log_2\frac{|S_i|}{|S|}+(1-\frac{|S_i|}{|S|})\times\log_2\left(1-\frac{|S_i|}{|S|}\right)\right] \tag{8}$$

Step 4: Calculate the Gain Ratio(X):

$$\text{Gain Ratio (X)} = \frac{\text{Information Gain(S)}}{\text{Split Info(X)}} \tag{9}$$

Step 5: The election of the root node is won by the attribute with the highest gain ratio. For each intermediate node, step 1 to step 4 will be repeated until all instances are exhausted and reach the leaf node according to step 2.

F. METRICS

Any classifier could have an error rate and it may fail to categorize correctly. Classification accuracy is calculated as correctly classified instances divided by the Total number of instances multiplied by 100.

In statistics, **Mean Absolute Error (MAE)** is a measure of the difference between two continuous variables. If we

TABLE 4. Classifiers evaluation summary: 10 fold cross-validation.

Scheme	Naïve Bayes	SVM	MLP	BF Tree	LAD Tree	J48
Correctly Classified Instances %	87,3111	87,422	87,022	87,155	87,355	87,444
Incorrectly Classified Instances %	12,6889	12,577	12,977	12,844	12,644	12,556
Mean absolute error	0,1216	0,083	0,122	0,115	0,118	0,119
Root mean squared error	0,243	0,289	0,249	0,255	0,239	0,248
Relative absolute error %	27,36	18,87	27,66	25,92	26,57	26,744
Root relative squared error	51,54	61,43	53,01	54,18	50,87	52,513
Time taken to build model (s)	0,03	0,16	2,71	0,93	0,53	0,05

TABLE 5. Classifiers evaluation summary: 15 fold cross-validation.

Scheme	Naïve Bayes	SVM	MLP	BF Tree	LAD Tree	J48
Correctly Classified Instances %	87,289	87,400	87,044	86,844	87,689	88,022
Incorrectly Classified Instances %	12,711	12,600	12,956	13,156	12,311	11,978
Mean absolute error	0,122	0,084	0,125	0,120	0,118	0,117
Root mean squared error	0,243	0,290	0,249	0,255	0,238	0,244
Relative absolute error %	27,365	18,900	28,133	26,954	26,522	26,406
Root relative squared error	51,556	61,482	52,796	54,182	50,578	51,725
Time taken to build model (s)	0,010	0,920	2,380	0,640	0,250	0,060

TABLE 6. Classifiers evaluation summary: 20 fold cross-validation.

Scheme	Naïve Bayes	SVM	MLP	BF Tree	LAD Tree	J48
Correctly Classified Instances %	87,267	87,556	87,000	87,133	87,444	87,778
Incorrectly Classified Instances %	12,733	12,444	13,000	12,867	12,556	12,222
Mean absolute error	0,122	0,083	0,125	0,117	0,119	0,118
Root mean squared error	0,243	0,288	0,249	0,252	0,240	0,246
Relative absolute error %	27,350	18,667	28,083	26,415	26,731	26,581
Root relative squared error	51,519	61,101	52,886	53,532	50,966	52,094
Time taken to build model (s)	0,010	0,920	2,360	0,550	0,250	0,060

assume that X and Y are variables that express the same phenomenon. Mean Absolute Error (MAE) is the measure of the average vertical (or horizontal) distance between each point and the identity line.

The Mean Absolute Error is given by:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \tag{10}$$

The **Root Mean Square Error (RMSE)**: is just the square root of MSE. The square root is introduced to make a scale of the errors to be the same as the scale of targets.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - x_i)^2 = MSE} \tag{11}$$

The measurement of RMSE is frequently used to determine the difference between the values predicted by a model and the observed values.

The **Relative Absolute Error (RAE)** is used in machine learning and data mining to determine the performance of a given predictive model. Thus, the relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor.

Mathematically, the Relative Absolute Error E_i of an individual program i is evaluated by the equation:

$$E_i = \frac{\sum_{j=1}^{n} \left|P_{(ij)} - T_j\right|}{\sum_{j=1}^{n} \left|T_j - \bar{T}\right|} \tag{12}$$

where $P_{(ij)}$ is the value predicted by the individual program i for sample case j (out of n sample cases); T_j is the target value

TABLE 7. Evaluation summary on training set.

Scheme	Naïve Bayes	SVM	MLP	BF Tree	LAD Tree	J48
Correctly Classified Instances %	87,289	88,022	87,467	90,844	87,756	88,422
Incorrectly Classified Instances %	12,711	11,978	12,533	9,156	12,244	11,578
Mean absolute error	0,121	0,080	0,123	0,086	0,116	0,116
Root mean squared error	0,243	0,283	0,245	0,207	0,235	0,240
Relative absolute error %	27,315	17,967	27,557	19,343	25,989	25,998
Root relative squared error	51,459	59,944	51,985	43,980	49,848	50,988
Time taken to build model (s)	0,010	0,900	2,400	0,530	0,260	0,060
Time taken to test model (s)	0,020	0,790	0,010	0,000	0,000	0,010

TABLE 8. Cost matrix.

		Predicted Class	
Actual Class	C(i|j)	Class = Yes	Class = No
	Class = Yes	C(Yes | Yes)	C(No | Yes)
	Class = No	C(Yes | No)	C(No | No)

for sample case j; and $\bar{T}$ is given by the formula:

$$\bar{T} = \frac{1}{n}\sum\nolimits_{j=1}^{n} T_j \tag{13}$$

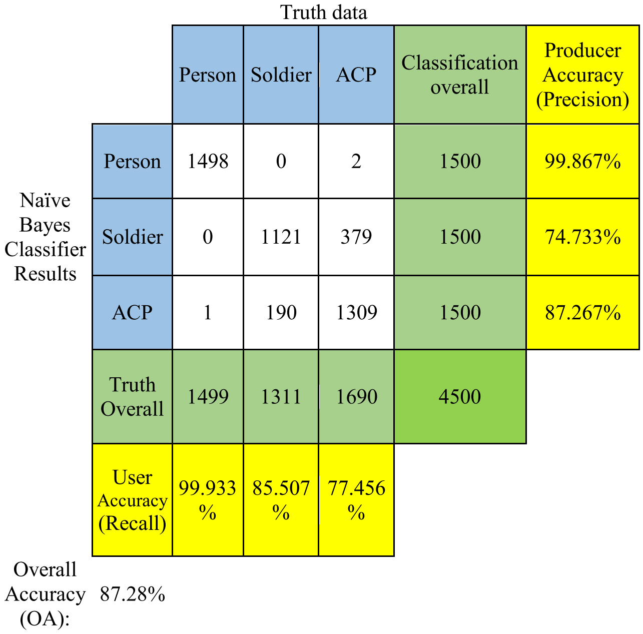
A **Confusion Matrix**, Also known as an error matrix, recaps the performance of a classification model by given the count of instances in a predicted class for each instance in an actual class.

G. RESULTS AND DISCUSSION

Based on all the training dataset and the different cross-validation folds (validation 5, 10, 15, and 20), authors can carry out a global analysis of the evaluation of the different classifiers (Naïve Bayes, SVM, Multilayer Perceptron, BF Tree, LAD Tree, and J48): Tables 3-7.

In light of the results of these tables, researchers can deduce the performance comparisons of the classifiers in terms of correctly classified instances versus incorrectly classified instances, mean absolute error versus mean squared error, relative absolute error versus relative squared error root, and the time required to build the model.

Naïve Bayes is the fastest approach in terms of model building time: the values collected are 0.01 s, 0.03 s, 0.01 s, and 0.01 s for 5, 10, 15, and 20 cross-validations respectively, compared to times ranging from 0.05 s to 0.06s

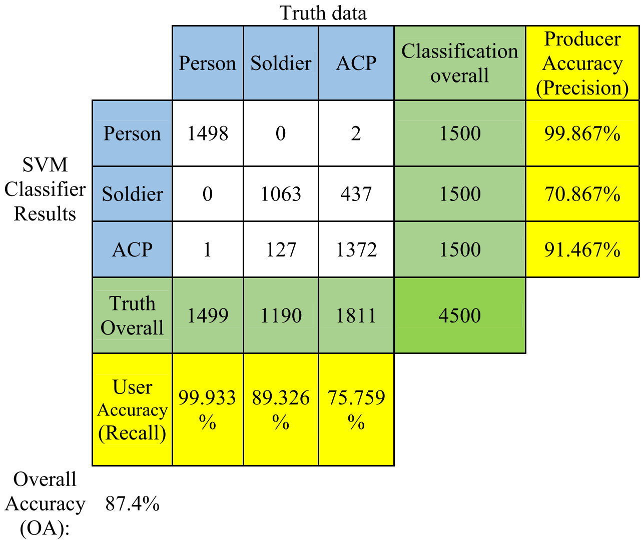


Truth data

		Person	Soldier	ACP	Classification overall	Producer Accuracy (Precision)
Naïve Bayes Classifier Results	Person	1498	0	2	1500	99.867%
	Soldier	0	1121	379	1500	74.733%
	ACP	1	190	1309	1500	87.267%
	Truth Overall	1499	1311	1690	4500	
	User Accuracy (Recall)	99.933 %	85.507 %	77.456 %		

Overall Accuracy (OA): 87.28%

FIGURE 10. Naïve Bayesian Confusion matrix with 15 fold cross-validation.

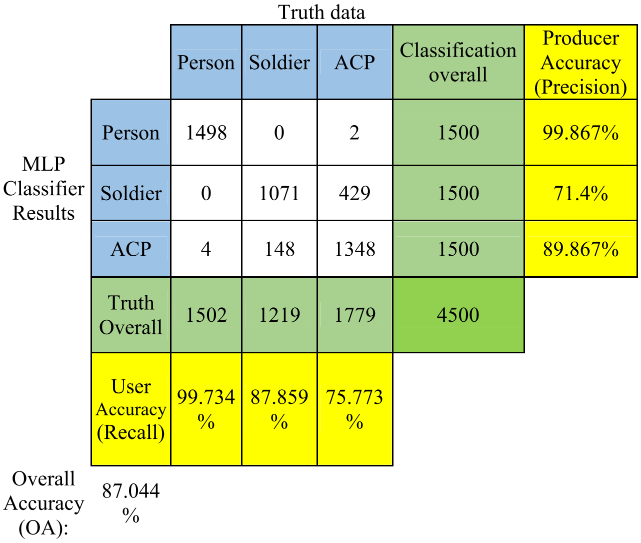


Truth data

		Person	Soldier	ACP	Classification overall	Producer Accuracy (Precision)
SVM Classifier Results	Person	1498	0	2	1500	99.867%
	Soldier	0	1063	437	1500	70.867%
	ACP	1	127	1372	1500	91.467%
	Truth Overall	1499	1190	1811	4500	
	User Accuracy (Recall)	99.933 %	89.326 %	75.759 %		

Overall Accuracy (OA): 87.4%

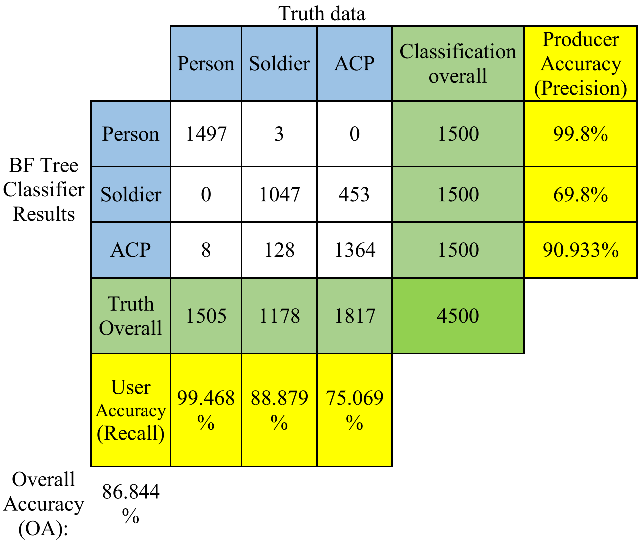
FIGURE 11. SVM Confusion matrix with 15 fold cross-validation.



Truth data

		Person	Soldier	ACP	Classification overall	Producer Accuracy (Precision)
MLP Classifier Results	Person	1498	0	2	1500	99.867%
	Soldier	0	1071	429	1500	71.4%
	ACP	4	148	1348	1500	89.867%
	Truth Overall	1502	1219	1779	4500	
	User Accuracy (Recall)	99.734 %	87.859 %	75.773 %		
Overall Accuracy (OA):	87.044 %					

FIGURE 12. **MLP Confusion matrix with 15 fold cross-validation.**



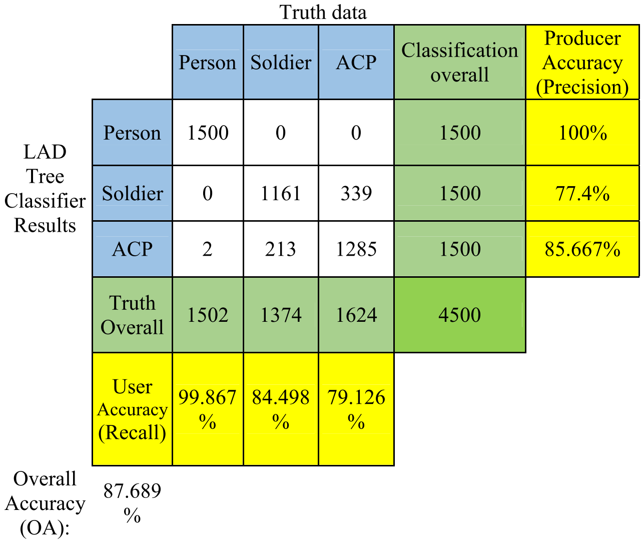
Truth data

		Person	Soldier	ACP	Classification overall	Producer Accuracy (Precision)
BF Tree Classifier Results	Person	1497	3	0	1500	99.8%
	Soldier	0	1047	453	1500	69.8%
	ACP	8	128	1364	1500	90.933%
	Truth Overall	1505	1178	1817	4500	
	User Accuracy (Recall)	99.468 %	88.879 %	75.069 %		
Overall Accuracy (OA):	86.844 %					

FIGURE 13. **BF-Tree Confusion matrix with 15 fold cross-validation.**

for J-48. However, over the entire cross-validation (5, 10, 15, 20), the J48 classifier always had the highest percentage of correctly classified instances (87.444%, 87.444%, 88.022%, 87.778%). J48 is the best algorithm in terms of accuracy. Also, we observe, that 15 fold cross-validation is the adequate way to get these best performances. In the training set, BF Tree has the best correctly classified instance rate, but it takes more time (0.53 s) to build a model than Naïve Bayes or J48 which have respectively (0.01 s, 0.06 s).

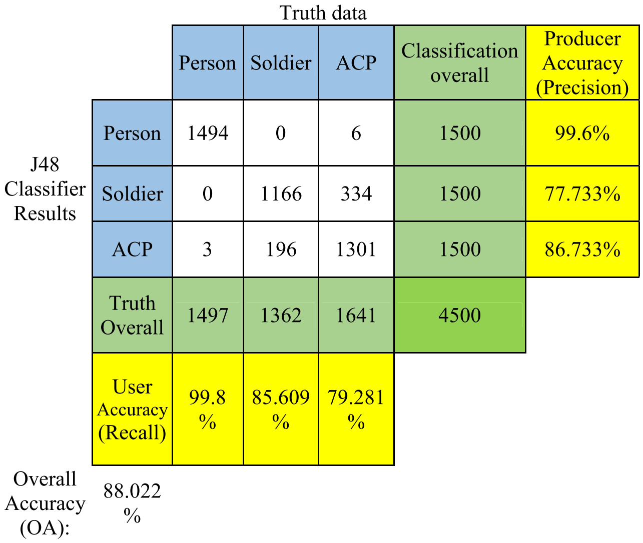
Another way to evaluate the quality of the output of a classifier is the confusion matrix: A confusion matrix is a technique for summarizing the performance of a classification algorithm. It can show a better idea of what our classification model is getting right and what types of errors it is making.



Truth data

		Person	Soldier	ACP	Classification overall	Producer Accuracy (Precision)
LAD Tree Classifier Results	Person	1500	0	0	1500	100%
	Soldier	0	1161	339	1500	77.4%
	ACP	2	213	1285	1500	85.667%
	Truth Overall	1502	1374	1624	4500	
	User Accuracy (Recall)	99.867 %	84.498 %	79.126 %		
Overall Accuracy (OA):	87.689 %					

FIGURE 14. **LAD-Tree Confusion matrix with 15 fold cross-validation.**



Truth data

		Person	Soldier	ACP	Classification overall	Producer Accuracy (Precision)
J48 Classifier Results	Person	1494	0	6	1500	99.6%
	Soldier	0	1166	334	1500	77.733%
	ACP	3	196	1301	1500	86.733%
	Truth Overall	1497	1362	1641	4500	
	User Accuracy (Recall)	99.8 %	85.609 %	79.281 %		
Overall Accuracy (OA):	88.022 %					

FIGURE 15. **J48 Confusion matrix with 15 fold cross-validation.**

The comparison of the confusion matrices for different test modes is given in the following figures: Fig. 10-15.

Based on:

- True Positive (TP) = event values are correctly predicted.
- True Negative (TN) = no-event values are correctly predicted.
- False Positive (FP) = event values are incorrectly predicted.
- False Negative (FN) = no-event values are incorrectly predicted.

We can deduce the following criteria:

- The Recall (True Positive Rate (TPR)):

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

TABLE 9. Cost / Benefit Analysis For class Person (Minimize Cost / Benefit).

TABLE I
COST MATRIX

Predicted (a)	Predicted (b)	
0	1.0	Actual (a)
1.0	0	Actual (b)
Total Population		4500

TABLE II
CONFUSION MATRIX

Predicted (a)	Predicted (b)	
1449 33.2%	6 0.13%	Actual (a) : Person
2 0.04%	2998 66.62%	Actual (b) : Soldier ; ACP
Classification Accuracy		99.8222%

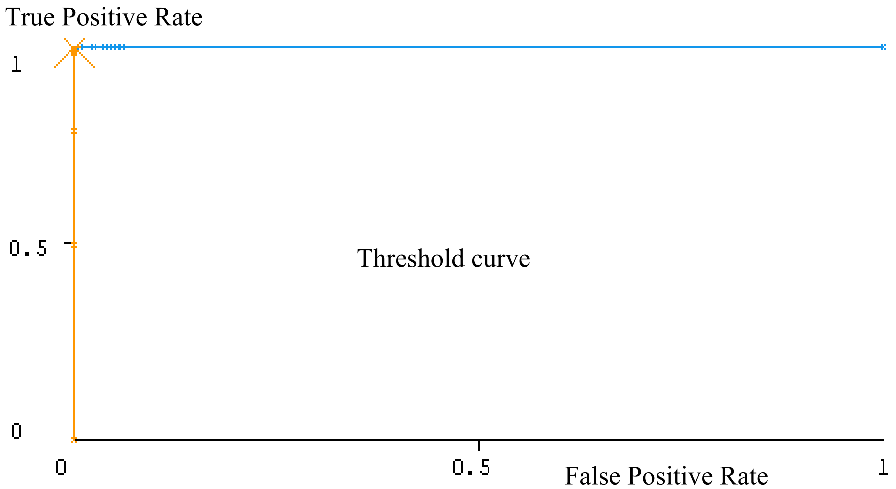


FIGURE 1. Threshold curve

Cost = 8

Random = 1998.67

Gain = 1990.67

% Population = 33.2444

% Target (recall) = 99.6

Score Threshold = 0.9804

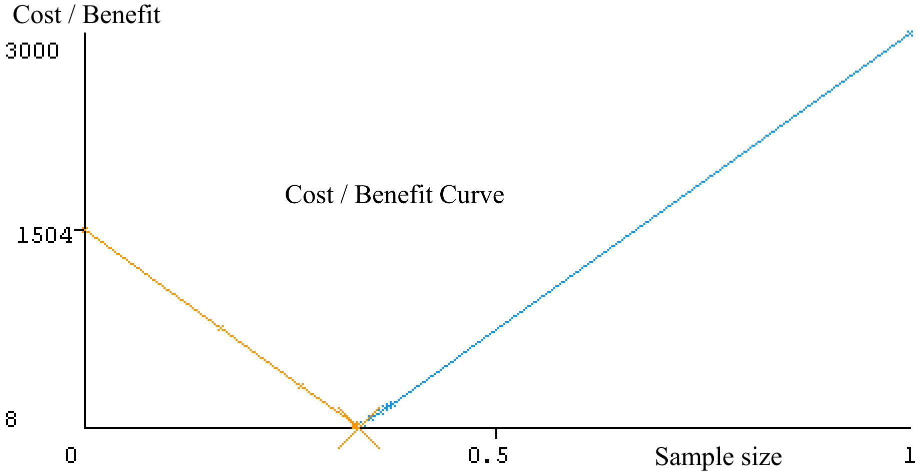


FIGURE 2. Cost / Benefit Curve

- The precision (positive predictive value (PPV)):

$$PPV = \frac{TP}{TP + FP} \tag{15}$$

- The accuracy (ACC):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

The analysis of the confusion matrices relating to the different classification approaches (Naïve Bayes, SVM, Multilayer Perceptron, BF Tree, LAD Tree, and J48), shows that the best overall accuracy with a rate equal to 88.022% is on the side of the J48 algorithm.

How can we interpret the result of false positive and false negative through the confusion matrices? Everyone agrees

TABLE 10. Cost / Benefit Analysis For class Soldier (Minimize Cost / Benefit).

TABLE III
COST MATRIX

Predicted (a)	Predicted (b)	
0	1.0	Actual (a)
1.0	0	Actual (b)

Total Population 4500

TABLE IV
CONFUSION MATRIX

Predicted (a)	Predicted (b)	
1164 25.87%	336 7.47%	Actual (a) : Soldier
217 4.82%	2783 61.84%	Actual (b) : Person; ACP

Classification Accuracy 87.7111%

Cost = 553

Random = 1960.33

Gain = 1407.33

% of Population = 30.6889

% of Target (recall) = 77.6

Score Threshold = 0.469

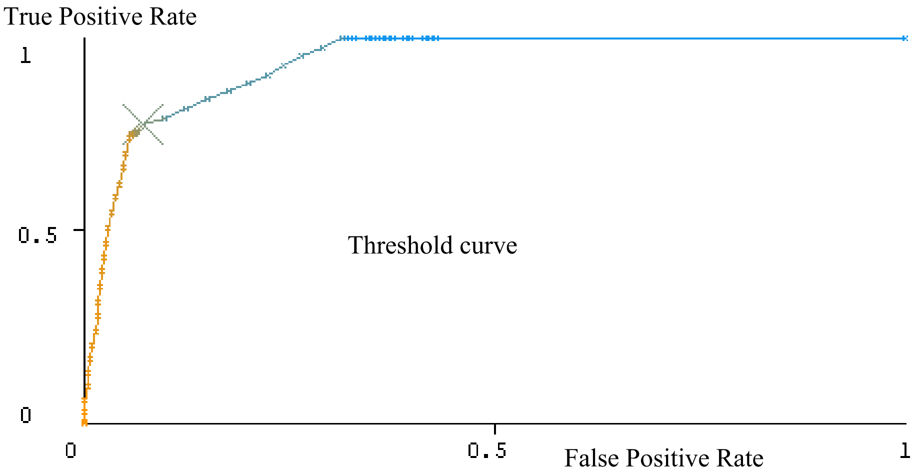


FIGURE 3. Threshold curve

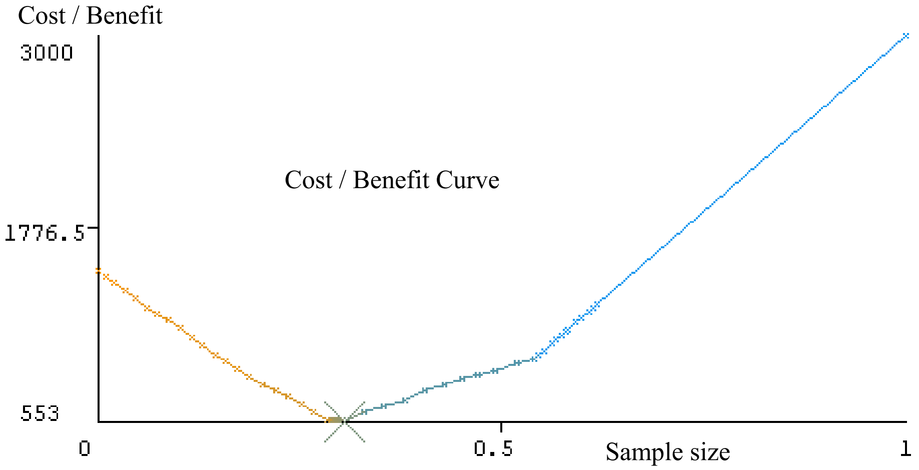


FIGURE 4. Cost / Benefit Curve

that the presence of imbalanced classification is an unavoidable aspect of data mining practice.

For example, in the medical diagnosis of certain disease such as coronavirus 19, if we consider a positive class for every infected person and a negative class for every other healthy person, then classifying a patient as healthy when he/she is really sick (False negative) is much more serious and dangerous than the reverse case, i.e. classifying a person as sick when he/she is healthy (false positive): In the first case, the person who is really sick will infect hundreds or even thousands of people, while in the second case the misclassification will be rectified after further tests or after the person's stable state. By analogy with this example and going back to the context of our application, it is better to identify the right

TABLE 11. Cost / Benefit Analysis For class ACP (Minimize Cost / Benefit).

TABLE V
COST MATRIX

Predicted (a)	Predicted (b)	
0	1.0	Actual (a)
1.0	0	Actual (b)
Total Population		4500

TABLE VI
CONFUSION MATRIX

Predicted (a)	Predicted (b)	
1281 28.47%	219 4.87%	Actual (a) : ACP
342 7.6%	2658 59.07%	Actual (b) : Person; Soldier
Classification Accuracy		87.7111%

Cost = 561

Random = 2041

Gain = 1480

% of Population = 30.0767

% of Target (recall) = 85.4

Score Threshold = 0.6791

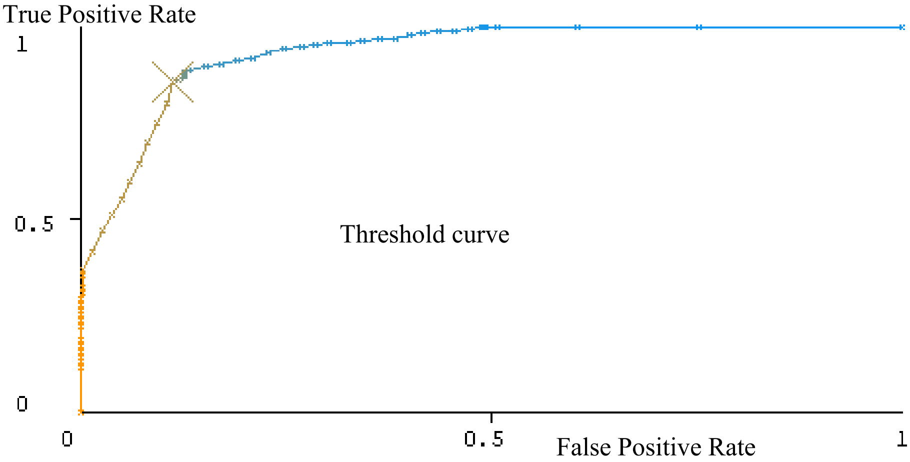


FIGURE 5. Threshold curve

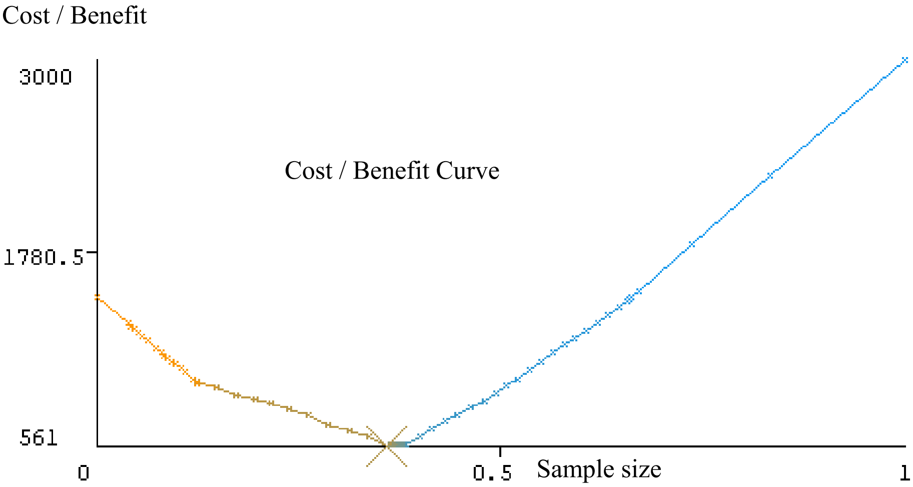


FIGURE 6. Cost / Benefit Curve

person (minority class) as intruders and follow up with them, than to honor the wrongly classified intruders.

The problem of imbalanced classification (misclassification) has been one of the factors in the emergence of a relatively new research topic in the field of machine learning under the name of "cost-sensitive learning" [30]. It consists of associating a "cost" penalty to an incorrect prediction and then trying to minimize the cost of a model on the learning dataset. This leads to cost-sensitive machine learning.

The misclassification cost can be described by a matrix called cost matrix $C = (C_{i|j})_{nn}$

Where $C_{i|j}$ indicates the cost due to misclassifying an instance I as class j. n represents the number of classes.

In the appendix, we take as an example the J48 algorithm and we present its cost matrix, the values of cost, gain, the percentage of target (recall), score threshold as well as the threshold curve and the Cost/Benefit Curve.

Independently of the classification approach adopted, researchers observed that the ambiguity (relatively different between the algorithms) of classification is essentially focused between a soldier and an ACP carrying a mass of iron. This leads us to reflect on an additional metric that plays the role of a decisive factor of discrimination between these two classes and minimizes the misclassification. This factor is the density (ρ) of the sensors to be deployed, otherwise what is the necessary number of sensors to detect an intruder. In [14], [31], the authors treat this idea. Indeed if we suppose that a target has a radius of influence that varies between rmin and rmax, then we can deduce the target influence area: Amin $= \pi$rmin2 and Amax $= \pi$rmax2 and by, therefore, the number (n) of sensors required to deploy in this area belongs to the interval [nmin $= \rho$Amin, nmax $= \rho$Amax]. On his part, Sami *et al.* [32] consider in their research, a distributed detection in a clustered wireless sensor network deployed randomly in a large field. Results obtained showed that as the number of clusters increases, the performance rapidly reaches the Chair-Varshney benchmark for fixed SNs deployment intensity. In other words, optimal detection can be achieved by forming more clusters in the network, in contrast to adding more sensor nodes to it.

V. CONCLUSION

In this paper, researchers have placed in the context of a monitoring application of the borderline between Tunisia and Algeria. The investigations were conducted by the border police on this subject, helped us to distinguish the nature of the intruders namely, a person, an armed person (soldier), and an ACP carrying a large mass of iron weapons. Then we carried out the kinetic study of these intruders, followed by a study on the nature of the sensors to be deployed for good detection. Finally, a comparative study was made between six classification approaches. The results obtained show the superiority of the J48 algorithm. It still remains that to find the best strategy to adopt for effective detection: Fix a density of deployment for our area of interest or choose the distributed detection in a clustered WSN.

APPENDIX

See Tables 9–11.

REFERENCES

[1] B. Aygün and V. C. Gungor, "Wireless sensor networks for structure health monitoring: Recent advances and future research directions," *Sensor Rev.*, vol. 31, no. 3, pp. 261–276, Jun. 2011, doi: 10.1108/02602281111140038.

[2] P. Sikora, L. Malina, M. Kiac, Z. Martinasek, K. Riha, J. Prinosil, L. Jirik, and G. Srivastava, "Artificial intelligence-based surveillance system for railway crossing traffic," *IEEE Sensors J.*, early access, Oct. 16, 2020, doi: 10.1109/JSEN.2020.3031861.

[3] G. Vallathan, A. John, C. Thirumalai, S. Mohan, G. Srivastava, and J. C.-W. Lin, "Suspicious activity detection using deep learning in secure assisted living IoT environments," *J. Supercomput.*, vol. 77, no. 4, pp. 3242–3260, Apr. 2021, doi: 10.1007/s11227-020-03387-8.

[4] K. Derr and M. Manic, "Wireless sensor networks—Node localization for various industry problems," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 752–762, Jun. 2015, doi: 10.1109/TII.2015.2396007.

[5] W. Chen and X. Wang, "Coal mine safety intelligent monitoring based on wireless sensor network," *IEEE Sensors J.*, early access, Dec. 21, 2020, doi: 10.1109/JSEN.2020.3046287.

[6] A. Altarazi, N. Al-Madi, and F. Awad, "Geometric-based localization for wireless sensor networks," in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Irbid, Jordan, Apr. 2020, pp. 108–112, doi: 10.1109/ICICS49469.2020.239558.

[7] W. Wen, X. Wen, L. Yuan, and H. Xu, "Range-free localization using expected hop progress in anisotropic wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 299, Dec. 2018, doi: 10.1186/s13638-018-1326-8.

[8] M. M. Abdellatif, J. M. Oliveira, and M. Ricardo, "The self-configuration of nodes using RSSI in a dense wireless sensor network," *Telecommun. Syst.*, vol. 62, no. 4, pp. 695–709, 2016, doi: 10.1007/s11235-015-0105-7.

[9] M. Shalaby, M. Shokair, and N. W. Messiha, "Performance enhancement of TOA localized wireless sensor networks," *Wireless Pers. Commun.*, vol. 95, no. 4, pp. 4667–4679, Aug. 2017, doi: 10.1007/s11277-017-4112-8.

[10] D. Xue, "Research of localization algorithm for wireless sensor network based on DV-hop," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 218, Dec. 2019, doi: 10.1186/s13638-019-1539-5.

[11] A. Paul and T. Sato, "Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications and challenges," *J. Sensor Actuator Netw.*, vol. 6, no. 4, p. 24, Oct. 2017, doi: 10.3390/jsan6040024.

[12] D. Chen, W. Wang, and Y. Zhou, "Notice of retraction: An improved DV-hop localization algorithm in wireless sensor networks," in *Proc. Int. Conf. Comput. Commun. Technol. Agricult. Eng.*, 2010, pp. 266–269, doi: 10.1109/CCTAE.2010.5544857.

[13] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Trans. Sensor Netw.*, vol. 1, no. 1, pp. 134–152, Aug. 2005.

[14] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, Dec. 2004.

[15] S.-H. Yang, *Wireless Sensor Networks Principles, Design and Applications*. London, U.K.: Springer-Verlag, 2014, doi: 10.1007/978-1-4471-5505-8.

[16] D. J. Cook and S. K. Das, *Smart Environments: Technology, Protocols and Applications*. London, U.K.: Wiley, Sep. 2004, doi: 10.1002/047168659X.

[17] Machine Learning Group at the University of Waikato. *WEKA The Workbench for Machine Learning*. Accessed: Oct. 19, 2020. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[18] W. Zhang and F. Gao, "An improvement to naive Bayes for text classification," *Procedia Eng.*, vol. 15, pp. 2160–2164, Dec. 2011, doi: 10.1016/j.proeng.2011.08.404.

[19] N. F. Rusland, N. Wahid, S. Kasim, and H. Hafit, "Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 226, Aug. 2017, Art. no. 012091, doi: 10.1088/1757-899X/226/1/012091.

[20] G. Rahangdale, M. Ahirwar, and M. Motwani, "Application of k-NN and Naïve Bayes algorithm in banking and insurance domain," *IJCSI Int. J. Comput. Sci. Issues*, vol. 13, no. 5, p. 69, Sep. 2016, doi: 10.20943/01201605.6975.

[21] S. Kharya and S. Soni, "Weighted naive bayes classifier: A predictive model for breast cancer detection," *Int. J. Comput. Appl.*, vol. 133, no. 9, pp. 32–37, Jan. 2016.

[22] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep. 2443126, Apr. 2010. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin

[23] C. Zhou, L. Wang, and L. Zhengqiu, "The study of WSN node localization method based on back propagation neural network," in *Advances in Intelligent Systems and Computing*, vol. 580, J. Abawajy, K. K. Choo, R. Islam, Eds. Cham, Switzerland: Edizioni della Normale, 2018, doi: 10.1007/978-3-319-67071-3_54.

[24] A. Gautam, V. Bhateja, A. Tiwari, and S. C. Satapathy, "An improved mammogram classification approach using back propagation neural network," in *Data Engineering and Intelligent Computing*, vol. 542, S. Satapathy, V. Bhateja, K. Raju, B. Janakiramaiah Eds. Singapore: Springer, 2018, doi: 10.1007/978-981-10-3223-3_35.

[25] H. K. Sok, M. P.-L. Ooi, Y. C. Kuang, and S. Demidenko, "Multivariate alternating decision trees," *Pattern Recognit.*, vol. 50, pp. 195–209, Feb. 2016, doi: 10.1016/j.patcog.2015.08.014.

[26] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall, "Multiclass alternating decision trees," in *Proc. ECML*, 2001, pp. 161–172.

[27] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuri, "Decision tree analysis on J48 algorithm for data mining," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 6, Jun. 2013, pp. 1114–1119.

[28] S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 decision tree," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2015, pp. 2023–2026.

[29] A. K. Srivastava, D. Singh, A. S. Pandey, and T. Maini, "A novel feature selection and short-term price forecasting based on a decision tree (J48) model," *Energies*, vol. 12, no. 19, p. 3665, Sep. 2019, doi: 10.3390/en12193665.

[30] F. Feng, K.-C. Li, J. Shen, Q. Zhou, and X. Yang, "Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification," *IEEE Access*, vol. 8, pp. 69979–69996, 2020, doi: 10.1109/ACCESS.2020.2987364.

[31] M. H. Jeridi, H. Khalaifi, A. Bouatay, and T. Ezzedine, "Targets classification based on multi-sensor data fusion and supervised learning for surveillance application," *Wireless Pers. Commun.*, vol. 105, no. 1, pp. 313–333, Mar. 2019.

[32] S. A. Aldalahmeh, S. O. Al-Jazzar, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Fusion rules for distributed detection in clustered wireless sensor networks with imperfect channels," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 5, no. 3, pp. 585–597, Sep. 2019, doi: 10.1109/TSIPN.2019.2901198.

MOHSEN OTHMANI was born in Tunis, Tunisia. He graduated in computer science engineering from the Faculty of Sciences of Tunis (FST), Tunisia. He received the master's degree in telecommunications from the National School of Engineers of Tunis (ENIT), Tunisia. He is currently the Ph.D. Student Researcher at SysCom Laboratory, ENIT, Tunisia. He is an Associate Professor with the College of Telecom and Electronics, Jeddah, Saudi Arabia. His research interests include WSN and neural networks.

MOHAMED HECHMI JERIDI graduated in telecommunication engineering from the Tunisian Aviation School, qualified in operation and maintenance of medium and high altitude aerial surveillance radars, on basic transmission, on HF and VHF radio, on Master T. Organizational/ Intermediate maintenance Level Course and on Radar Deployment Specialty Thales Company, France, certified of proficiency in arcade spectrum management tools and frequency management basic course Communications and Information Systems School Latina, Italy. He certified compliance with the R & TTE directive Central Laboratory of Electrical Industries, Certified for training for the successful completion of the Project Management Training 101 and the Ph.D. Student Researcher with the Laboratory of Communication Systems (Syscom), National Engineering School of Tunis, Tunisia.

QING-GUO WANG received the B.Eng. degree in chemical engineering, and the M. Eng. and Ph.D. degrees in industrial automation from Zhejiang University, China, in 1982, 1984, and 1987, respectively.

From 1992 to 2015, he was with the Department of Electrical and Computer Engineering, National University of Singapore, where he became a Full Professor, in 2004. From 2015 to 2020, he was a Distinguished Professor with the Institute for Intelligent Systems, University of Johannesburg, South Africa. Since 2020, he has been a Chair Professor with BNU-UIC Institute of Artificial Intelligence and Future Networks, Beijing Normal University (BNU Zhuhai), BNU-HKBU United International College, China. He holds A-rating from the National Research Foundation of South Africa. He is a member of Academy of Science of South Africa. He has published over 330 international journal articles and seven research monographs. He received about19000 citations with H-index of 75. His research interests include modeling, estimation, prediction, control, optimization, and automation for complex systems, including but not limited to, industrial and environmental processes, new energy devices, defense systems, medical engineering, and financial markets. He held Alexander-von-Humboldt Research Fellowship of Germany, from 1990 to 1992. He received the Award of the Most Cited Article of the Journal of *Automatica*, in 2006–2010, and in the Thomson Reuters list of the Highly-Cited Researchers 2013 in engineering. He is currently the Deputy Editor-in-Chief of the *ISA Transactions*, USA.

TAHAR EZZEDINE received the M.S., Ph.D., and H.D.R. degrees in telecommunications from the National School of Engineers of Tunis (ENIT), Tunisia. He is currently a Full Professor with the Department of Information and Communication Technology and the Founder/Leader of a Research and Development Group for research and development of wireless sensors networks in different fields enclosing GIS systems with SysCom Research Laboratory. His interests include smart objects and the IoT.