Mach Learn (2017) 106:1547–1567
DOI 10.1007/s10994-017-5646-4

Weightless neural networks for open set recognition

Douglas O. Cardoso[1] · **João Gama**[2] · **Felipe M. G. França**[3]

Received: 28 March 2016 / Accepted: 20 May 2017 / Published online: 12 July 2017

Abstract Open set recognition is a classification-like task. It is accomplished not only by the identification of observations which belong to targeted classes (i.e., the classes among those represented in the training sample which should be later recognized) but also by the rejection of inputs from other classes in the problem domain. The need for proper handling of elements of classes beyond those of interest is frequently ignored, even in works found in the literature. This leads to the improper development of learning systems, which may obtain misleading results when evaluated in their test beds, consequently failing to keep the performance level while facing some real challenge. The adaptation of a classifier for open set recognition is not always possible: the probabilistic premises most of them are built upon are not valid in a open-set setting. Still, this paper details how this was realized for WiSARD a weightless artificial neural network model. Such achievement was based on an elaborate distance-like computation this model provides and the definition of rejection thresholds during training. The pro-

Editors: Thomas Gärtner, Mirco Nanni, Andrea Passerini, and Celine Robardet.

Douglas O. Cardoso thanks CAPES (process 99999.005992/2014-01) and CNPq for financial support. João Gama thanks the support of the European Commission through the project MAESTRA (Grant Number ICT-750 2013-612944). Felipe M. G. França thanks the support of FAPERJ, FINEP and INOVAX.

✉ Douglas O. Cardoso
douglas.cardoso@cefet-rj.br
http://docardoso.github.io

João Gama
jgama@fep.up.pt
http://www.liaad.up.pt/area/jgama/

Felipe M. G. França
felipe@cos.ufrj.br
http://www.cos.ufrj.br/∼felipe

1 Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, GCOMPET, Petrópolis, RJ, Brazil

2 Universidade do Porto, LIAAD-INESC TEC, Oporto, Portugal

3 Universidade Federal do Rio de Janeiro, PESC-COPPE, Rio de Janeiro, RJ, Brazil

posed methodology was tested through a collection of experiments, with distinct backgrounds and goals. The results obtained confirm the usefulness of this tool for open set recognition.

Keywords Open set recognition · Classification · Reject option · Anomaly detection · WiSARD

1 Introduction

Classification is an activity which models numerous everyday situations. The fundamental classification problem regards two classes, and assumes the prior availability of a data sample which reflects the characteristics of the population under consideration. Its most natural variant is multi-class classification, in which the number of classes is greater than two. Another popular related task is the identification of elements of a single, well-known class, what is called one-class classification (Khan and Madden 2009), or anomaly detection. As it can be perceived, all these alternatives differ by the number of classes to be modeled. A third task based on classification is open set recognition (Scheirer et al. 2013). For its accomplishment, observations of some classes should be recognized accordingly while inputs which do not belong to any of these classes should be rejected. In this context, to reject a data point means to consider it unrelated to all classes learned from the training sample.

Hypothetically speaking, using a classifier for open set recognition would require to make it capable of identifying extraneous data. Discriminative classifiers, which work based on the conditional probability $P(y|\mathbf{x})$, can only provide the distance between a given observation $\mathbf{x}$ and the decision margin defined during training. This information is somewhat poor for the purpose of rejection. Generative classifiers seem to be naturally more appropriate for this matter: the joint probability $P(\mathbf{x}, y)$ they model could be readily used evaluate the pertinence of $\mathbf{x}$ to y. However, the probabilistic foundation of these models does not comprise the reduced notion of the prior probabilities of the classes, a inherent characteristic of open set tasks.

A Wilkes, Stonham and Aleksander Recognition Device (WiSARD) classifier (Aleksander et al. 1984) is composed by a collection of discriminators, which are used to evaluate how well an observation fits the classes they represent. Despite the name of such structures (discriminators), WiSARD exhibits some generative capabilities: for example, it is possible to obtain prototypes of the known classes through a procedure called DRASiW (Grieco et al. 2010), a reference to the reversal of the ordinary system operation. Producing such prototypes is possible thanks to how learning is realized by this model, explicitly collecting pieces of information from training data for later use. Such generative trait of WiSARD motivated the analysis of its use for open set recognition. After some exploratory results (Cardoso et al. 2015), now a fully developed methodology is detailed here.

The remainder of this paper is organized as follows: Sect. 2 presents the theoretical basis used for the development of this work; Sect. 3 explains the computation of rejection thresholds; Sect. 4 presents experiments for practical evaluation of the proposed methodology; at last, some concluding remarks are provided in Sect. 5.

2 Research background

2.1 Open set recognition

Classification requires that all classes in the problem domain are well-represented in the training sample. Such condition is called the *closed set assumption*. As the name implies,

Table 1 Differences between open set recognition and related problems

Task	Goal	Training data	Predictions
Classification	Discrimination between classes	Abundant data of all classes	Label of a known class
Anomaly detection	Recall of abnormal data	Abundant normal data; few or none outliers	Outlier: yes or no
Open set recognition	Identification of data from targeted classes	Abundant targeted data; few or none non-targeted	Label of a targeted class or 'unknown'

this is not necessary for open set recognition: beyond known classes, there could be an even larger collection of unknown classes whose observations should be identified as so. A fundamental difference between classification and recognition tasks is in the set of possible outcomes of inferences: for regular classification, the best guess for the true class of an input observation is always provided; for recognition, if none of the known classes appears to be the true class, the response is to consider the observation foreign to all known classes. The action of ruling an observation as extraneous, which occurs in detection and recognition tasks, is referred to as *rejection*. Table 1 summarizes the differences between open set recognition and its closest relatives.

Unfortunately, a great number of works which ignore the necessity of rejection can be found in the literature. These works proposed solutions to problems which are mistaken as regular classification tasks, although dealing with data from non-targeted classes is not only hypothetically possible but expected in practice. This could lead to poor results when one of these solutions is employed out of its test bed. Such questionable approaches can be found in various contexts: fault detection (Mirowski and LeCun 2012) and human activity recognition (Anguita et al. 2013) are some examples.

As a simple and clear-cut description, open set recognition can be seen as an automated learning task in which:

- any data point $\mathbf{x} \in \mathbb{R}^n$ is related to a single class, or label, $y = f(\mathbf{x}) \in \mathbb{N}$;
- a training set, i.e., a collection of data points $X = \{\mathbf{x}_i\}$ and respective labels $Y = \{y_i\}$, is available a priori;
- if $f(\mathbf{x}_i)$ is a *targeted* class, then $y_i = f(\mathbf{x}_i)$, else $y_i = -1$ (i.e., 'unknown');
- $\hat{y} = \hat{f}(\mathbf{x})$ denotes a prediction of the true class of $\mathbf{x}$, based on training data;
- as a task goal, if $f(\mathbf{x})$ is targeted, $\hat{f}(\mathbf{x}) = f(\mathbf{x})$, else $\hat{f}(\mathbf{x}) = -1$;
- the possibility of predicting $\hat{y} = -1$ is referred to as *reject option*, an alternative to regular class prediction;
- elements of non-targeted classes in $\{f(\mathbf{x}) : \mathbf{x} \in X\}$ as well as those of classes not represented in the training sample should be rejected;
- the use of reject option should be adjusted, as part of the learning process.

An interesting aspect of a task which requires rejection is how important this action is for its accomplishment. This comes from the fact that for different problems, the amount of data which should be rejected may differ. For example, rejection is less useful for the recognition of chickens and ducks among farm animals than among birds in general, as the last group is broader than the first. From this intuition, the *openness* of a given problem is an estimate of the indispensability of rejection for its proper solution. Scheirer et al. (2013) defined this

measure as shown in Eq. (1), using three quantities: C_e, the number of all existing classes, which could have to be handled while performing predictions; C_t, the number of classes with observations in the training sample; and C_r, the number of targeted classes. The following relation holds: $C_r \leq C_t \leq C_e$.

$$\text{Openness} = 1 - \sqrt{\frac{2C_t}{C_r + C_e}}. \tag{1}$$

Open set recognition requires learning not only the differences between targeted classes but also what distinguishes data of these classes to extraneous data. This first requirement is already covered by existing classifiers functioning. Therefore, the adaptation of these models to this second requirement can be considered reasonable. A straightforward idea in this regard is to attach to each class prediction some sort of confidence rate of such inference.

A margin classifier, as a multilayer perceptron or a support vector machine (SVM) works by the definition of a function $f : \mathbb{R}^d \to \mathbb{R}$, $\mathbf{x} \mapsto y$ which provides class predictions $\hat{y} = \text{sgn}(f(\mathbf{x}))$. For any $\mathbf{x} \in \mathbb{R}^d$, $f(\mathbf{x})$ is nothing but the signed distance between $\mathbf{x}$ and a decision margin. This naturally induces the idea of using this value to identify extraneous elements: the farther $\mathbf{x}$ is from the margin, the stronger is the evidence that it does not belong to the known classes. However, the distance to the margin of two hypothetical observations can be the same, while their distance to train data is arbitrarily different. In the end, the only information any margin classifier can provide is this observation-to-margin distance. Consequently, a confidence rate to be used for rejection is hard to compute for a class prediction realized by a classifier of this kind.

As a matter of fact, this limitation can be related to the kind of probabilistic model a margin classifier is, trying to approximate $\text{argmax}_y P(y|\mathbf{x})$ using the learned decision function $f(\mathbf{x})$. Alternatively, generative classifiers estimate the joint probability $P(\mathbf{x}, y)$, from which the conditional probability can be computed. Although it may seem acceptable to use the probability $P(\mathbf{x}, y)$ as the desired confidence rate for the association of $\mathbf{x}$ to class y, this is not true. The fact that prior probability of the classes is generally unknown in open set problems disallows inference based on probabilistic principles as the Law of Total Probability and Bayes' theorem (Scheirer et al. 2014). Besides this, a good estimation of the probability density targeting rejection would require a large, noise-free data set (Tax and Duin 2008), richer in the informative aspect than a data set to be used just for classification.

There is a rich variety of works in the literature regarding classification with reject option (Fischer et al. 2016; Herbei and Wegkamp 2006; Bartlett and Wegkamp 2008; Yuan and Wegkamp 2010; Fumera and Roli 2002; Zhang and Metaxas 2006; Grandvalet et al. 2008). Although related in some sense, this task should not be mistaken by open set recognition. Indeed, both allow to reject an observation instead of classifying it. However, their difference lies in the purpose of such action: for classification with reject option, such alternative action targets avoid ruling an observation of one class as a member of another one; for open set recognition, rejection is primarily intended to observations which do not belong to any known class. Thus, their association to any class represents a wrong prediction, while rejection is the only right decision. Therefore, methods and techniques for classification with reject option should not be carelessly used for open set recognition.

There exist approaches for open set recognition in the literature. Many of these are based on discriminative principles: rejection-adapted support vector classifiers (Scheirer et al. 2013, 2014; Jain et al. 2014) and ensembles of one-class classifiers based on support vectors (Chen et al. 2009; Hanczar and Sebag 2014; Homenda et al. 2014) are possibly the most common descriptions of methods recently proposed for this task. This can be considered a natural

consequence of the popularity of these techniques, previously used in huge variety of closed set tasks. However, for open set recognition, a solution with a generative background could fit in more naturally thanks to its embedded confidence estimation. That is, the adaptation of a solution of this kind looks less painful than the same for a discriminative solution. A promising alternative is the development of a distance-based (Tax and Duin 2008) or prototype-based (Fischer et al. 2015) method. Such solution would have some capabilities similar to generative methods, while avoiding the probabilistic premises which are not valid in open set tasks.

2.2 Weightless artificial neural networks and WiSARD

Most mainstream artificial neural network (ANN) models (McCulloch and Pitts 1943) accomplish learning modifying weights associated to edges which interconnect network nodes. Weightless ANNs (Aleksander et al. 2009) are memory-based alternatives to weights-based ones. All links of these networks have no weight, acting as the simplest communication channels, exercising no effect on data traffic. Therefore, their nodes are responsible for the learning capability these networks exhibit. These nodes operate as memory units, keeping small portions of information. Such parts are combined when a query regarding the knowledge the system possess needs to be answered. These information pieces are the outcome of mapping the data used as knowledge source.

The biological inspiration of these nodes is the influence of dendritic trees on neuron functioning. In the first abstraction described, such trees were modeled as a weighted edges, which multiply the neuron inputs before the application of the activation function on their summation. Although practical, this is a rough simplification of how these trees operate. As a matter of fact, the input signals of biological neurons, which can be of two types (excitatory or inhibitory), are combined by the dendritic tree before reaching the neuron soma, where they prompt the generation of a new signal. This action can be naturally compared to the definition of a boolean key used to access a index of boolean values. In fact, this is how the most basic neurons of weightless ANN models work.

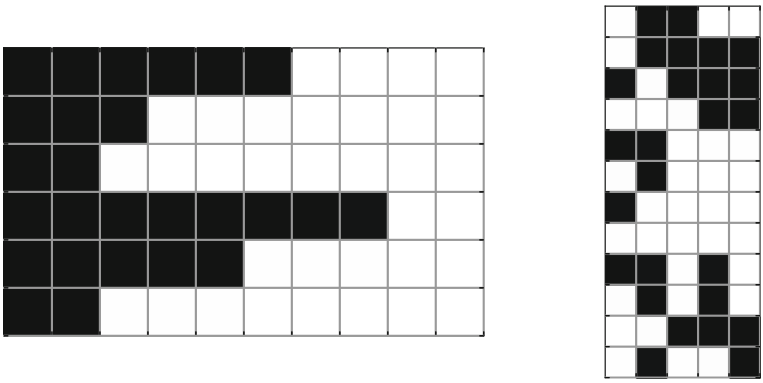
The WiSARD (Aleksander et al. 1984) is a member of the family of weightless ANN models. Such model was originally designed for classification. To realize a class prediction, it provides for each class a value in the interval [0, 1]. The value concerning a class represents how well the provided observation matches the acquired knowledge regarding that class. The values which compose an answer given by WiSARD are computed from structures called discriminators. Each discriminator is responsible for storing the knowledge regarding a class, as well as assessing the matching between the class it represents and any observation whose class has to be predicted. Because its functioning comes down to explicitly managing information divided into tuples of bits, this model is also known as n-tuple classifier.

How a discriminator learns about its respective class is described in Algorithm 1. In a sentence, it records in its nodes the values resulting from mapping the observations in the training sample. Mind some notation introduced next. The discriminator of class $\dot{y}$ is represented by $\Delta_{\dot{y}}$. The jth node of $\Delta_{\dot{y}}$ is represented by $\Delta_{\dot{y},j}$. The number of nodes which compose each discriminator is represented by δ, a model parameter. Vector addressing$(\mathbf{x}) = (a_1\ a_2\ \ldots\ a_\delta)$ has δ entries, and its jth entry addressing$_j(\mathbf{x}) = a_j$ is a binary string with β bits. At last, β is also a model parameter.

After training, a WiSARD instance can rate the matching between any known class $\dot{y}$ and an observation $\mathbf{x}$ as shown in Eq. (2a). Consider that $X_{\dot{y}}$ denotes the subset of observations

1: **for all** $\Delta_{\dot{y},j}$, the network nodes **do**
2: $\quad \Delta_{\dot{y},j} \leftarrow \varnothing$ ▷ Initially, nodes are empty sets
3: **for all** pairs $(\mathbf{x}_i, y_i)$, the training sample **do**
4: $\quad$ Let addressing$(\mathbf{x}_i) = (a_1\ a_2\ \ldots\ a_\delta)$ be a δ-dimensional vector mapped from $\mathbf{x}_i$
5: $\quad$ **for all** addresses a_j in addressing$(\mathbf{x}_i)$ **do**
6: $\quad\quad \Delta_{y_i,j} \leftarrow \Delta_{y_i,j} \cup \{a_j\}$

Algorithm 1: A description of WiSARD training procedure



(a) **(b)**

Fig. 1 An illustration of an addressing procedure, considering: $n = 6, \gamma = 10, \delta = 12, \beta = 5$ and $\mathbf{x} = (0.64, 0.27, 0.24, 0.76, 0.46, 0.22) \in [0, 1]^n$. **a** Binary matrix resulting from the application of Eq. (3a) on $\mathbf{x}$. Each row of **b** is an address, to be used as a RAM node key during WiSARD training or its matching computation. **a** $e(\mathbf{x})$. **b** $m \circ e(\mathbf{x})$

in the training set X which belong to class $\dot{y}$. At last, classification happens according to Eq. (2b).

$$\text{matching}(\mathbf{x}, X_{\dot{y}}) = \frac{1}{\delta} \sum_j \left[\text{addressing}_j(\mathbf{x}) \in \Delta_{\dot{y},j}\right]; \tag{2a}$$

$$\hat{y} = \text{argmax}_{\dot{y}}\, \text{matching}(\mathbf{x}, X_{\dot{y}}). \tag{2b}$$

Mathematically, WiSARD addressing procedure can be described as a composite function $m \circ e : \mathbb{R}^n \to \{0, 1\}^{\delta \times \beta}$, such that: $e : \mathbb{R}^n \to \{0, 1\}^{n \times \gamma}$ is any encoding function (Kolcz and Allinson 1994; Linneberg and Jorgensen 1999) which provides binary representations of the observations; and $m : \{0, 1\}^{n \times \gamma} \to \{0, 1\}^{\delta \times \beta}$ is a random mapping defined prior to training, described as $\mathbf{A} \mapsto \mathbf{B}$, $B_{i,j} = A_{i',j'}$, for arbitrary i, j, i', j'. Variable γ, which controls encoding resolution, is another model parameter. If data is originally binary, an identity-like function can be used for encoding: that is the case for black-and-white images, the kind of data for which WiSARD was originally developed. Otherwise, for example, if all data features are scaled to interval [0, 1], the zero-padded-unary encoding function, Eq. (3a), can be used. Still in this regard, Fig. 1 illustrates an hypothetical addressing operation.

$$e(\mathbf{x}) = (h(x_0), h(x_1), \ldots, h(x_n)), \tag{3a}$$

$$h(y) = \left(\left[\lfloor \gamma y \rceil \geq 1\right], \left[\lfloor \gamma y \rceil \geq 2\right], \ldots, \left[\lfloor \gamma y \rceil \geq \gamma\right]\right). \tag{3b}$$

[1] Iverson bracket: $[L] = 1$ if the logical expression L is true; otherwise, $[L] = 0$.

[2] $\lfloor x \rceil$ represents the nearest integer of real number x.

As previously stated, in Sect. 2.1, open set recognition implies working with significantly poorer prior knowledge compared to regular classification. The same can be said about the variant of such task in which rejection is allowed, but the probabilistic premises of the task remain unaltered (Scheirer et al. 2014). This motivates using WiSARD in this condition, as it does not rely on an estimation or assumptions regarding data distribution, what opposes various classifiers. Instead, it works rating how well an observation to be classified fits to stored knowledge based on counting corresponding features. One of the goals of this research was to verify the utility of such fitting level for rejection, despite the apparent simplicity of its calculation.

From a certain perspective, a discriminator works as a complex "distance" meter. That is, during training it stores numerous binary features extracted from observations of the class it represents. Then, the proximity between an observation and the knowledge maintained by the discriminator is measured according to the number of binary features extracted from this observation which match those features previously stored. Still in this regard, model parameters δ and β control the granularity of such measure. Such interpretation of WiSARD matching is aligned with previously established ideas about distance-based rejection (Tax and Duin 2008). However, its characterization in this regard is important to confirm the validity of such point of view for the intended application.

Additionally, WiSARD quasi-generative trait also inspired the examination of its functioning in an open-set context. For this purpose, an alternative setup of this model was conceived. In such setup, instead of simply storing features obtained during training, the absolute frequency of each feature would be computed. These counts would be used for the definition of prototypes of the modeled classes (Grieco et al. 2010), similarly to a generative model. The embedded rejection capability of generative classifiers and the previous use of prototype-based methods in this regard (Fischer et al. 2015) complete this idea.

Having in mind the aforementioned characterization of WiSARD matching computation, Figs. 2, 3 and 4 depict a comparison of it to some well-known data analysis tools. This comparison concerns proximity assessment based on toy data samples. This aims to provide some intuitive notion of how WiSARD differs from alternatives with some similar capabilities. Each test case follows the same idea: given a base data sample of 100 two-dimensional observations and a delimited area in the space, estimate the distance between each point in this area and the sample. The measurements were scaled in order to indicate the proximity to the sample, as values from 0 to 1, the farthest to closest, respectively. Using these proximity rates, a contour plot was drawn to highlight subareas in which the assessed proximity is similar. A dotted line was used to delimit where proximity rate is above zero.

3 Computation of rejection thresholds

The starting point of the proposed development is the view of matching computation as an observation-to-data proximity meter. From this, it is possible to move on to the next step in the conception of a rejection-capable WiSARD. As originally defined, classification comes down to the identification of the best matching class, based on the knowledge kept by its respective discriminator. Therefore, the matching rates of the classes were used just to separate each other in the feature space. Now, considering the general proximity information these measurements provide, it is acknowledged that their use can be extended, for example, to the identification of extraneous data.

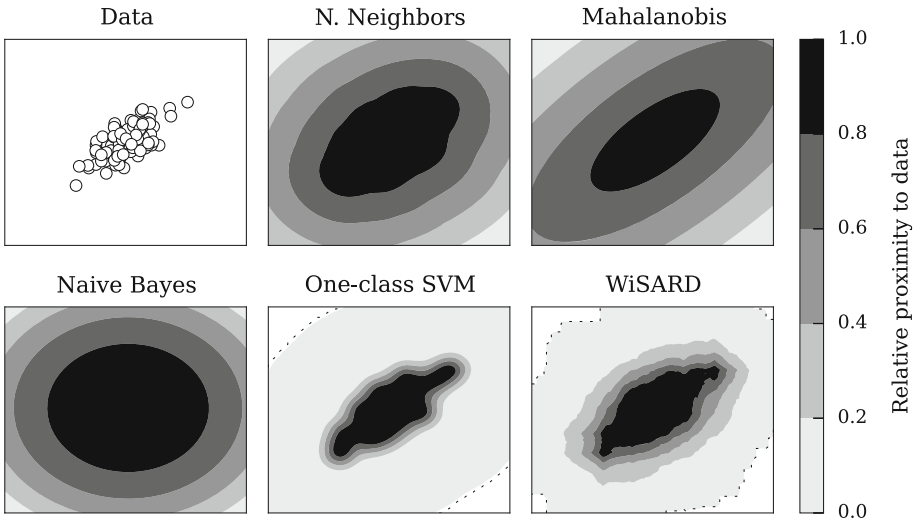


Fig. 2 The 'gaussian blob' toy example. The most evident difference between WiSARD matching and its alternatives is the irregularity of the provided contour levels. This can be related to WiSARD lower granularity compared to its rivals. However, WiSARD best reflects data idiosyncrasies, thanks to its distinct feature matching principle. Such mechanism is inherently discontinuous, contrasting with the smoothness of other methods

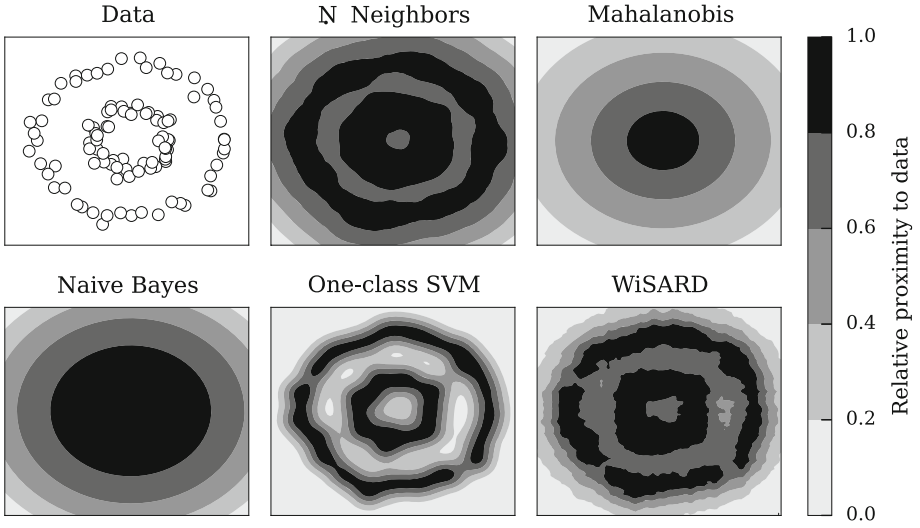


Fig. 3 The 'two circles' toy example. Again, WiSARD roughness is clear, but also its overall proper data representation, clearly superior to those of Mahalanobis and Naive Bayes options. An adequate approach for this test should have an improved sense of locality and enough precision to separate both circles, what was successfully accomplished by WiSARD. The nearest neighbors method concentrated most of its measurements in the interval [0.6, 1.0], while measurements of the one-class SVM are mostly under 0.2, while the range [0.2, 0.8] is underused. WiSARD seems to distribute the measurements more evenly, providing an alternative, possibly more meaningful, proximity assessment

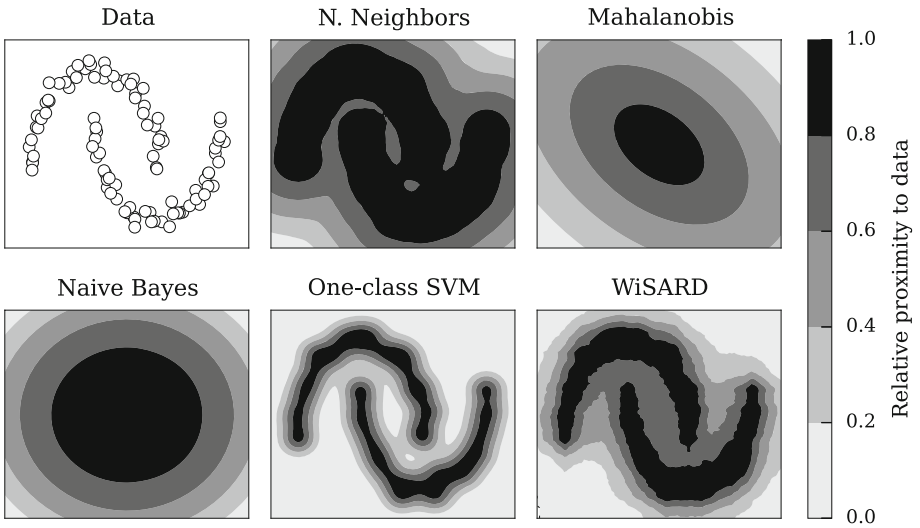


Fig. 4 The 'two moons' toy example. Assuming the independence between input attributes, the Naive Bayes method is unable to measure proximity to data. Mahalanobis distance (and presumably any method based on global average distance or centroid) lacks locality, what leads to poor results when handling complex, detail-rich data sets, and multi-modal classes. The nearest neighbors method provides adequate results, but is unable to highlight minutiae of the sample. The one-class SVM yields smooth contours and great deal of detail, but concentrates its measurements on the extremes of the scale, either close to 0 or 1. WiSARD's most patent characteristics are its meaningful proximity assessment and precise reproduction of data peculiarities, but also its irregularity. This way, this test confirms what the previous ones show

The most straightforward mechanism to label possible foreign examples is to consider as so any observation $\mathbf{x}$ for which matching$(\mathbf{x}, X_{\hat{y}}) < t$. It could be considered that $t \in [0, 1]$ is an additional parameter which controls how prone to rejection is the system. This would work if the distributions of matching scores of all classes were the same. However, these distributions generally differ, according to characteristics of training data respective to each class, as sample size, density and homogeneity. Thus, a scheme using individual thresholds $t_{\dot{y}}$ for each targeted class $\dot{y}$ is preferred, allowing to handle unbalanced and noisy data sets properly (Fumera et al. 2000). Equation (4) is the rejection-capable alternative to Eq. (2b) which represents such scheme. The ultimate target is to learn these thresholds from data, making their definition as flexible as possible.

$$\hat{y} = \begin{cases} y' & \text{if } y' = \operatorname{argmax}_{\dot{y}} \ \text{matching}(\mathbf{x}, X_{\dot{y}}) \wedge \text{matching}(\mathbf{x}, X_{y'}) \geq t_{y'} \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

The multiple-threshold rejection scheme proposed here was developed in two steps. First, it was analyzed how to efficiently infer some knowledge about the matching of a class and its own elements, according to available training data. From such analysis, it was derived one rejection mechanism which resembles the aforementioned naive alternative, but provides thresholds adapted to each class. The next step comes down to identifying, for each class, the threshold which maximizes a measure of classification effectiveness defined according to a model parameter. These optimal thresholds, whose definition is based on the information obtained in the first step, are employed by a second rejection method also introduced here. Sections 3.1 and 3.2 are dedicated to each of these parts.

3.1 Manual thresholding

Consider, for a certain class $\dot{y}$ and some training data, that matching$(\mathbf{x}, X_{\dot{y}})$ is a random variable, since it depends on $\mathbf{x}$, another random variable. Suppose that although the distribution of matching$(\mathbf{x}, X_{\dot{y}})$ is not fully determined, the minimum value this variable can assume for observations whose true class is $\dot{y}$, $\{\mathbf{x} : f(\mathbf{x}) = \dot{y}\}$, is known. The intuition behind the rejection method presented next is to use such value for thresholding:

$$t_{\dot{y}} = \min_{\mathbf{x}: f(\mathbf{x})=\dot{y}} \text{matching}(\mathbf{x}, X_{\dot{y}}).$$

In practice, this minimum is indeterminable: the set $\{\mathbf{x} : f(\mathbf{x}) = \dot{y}\}$ is impossible to realize without complete knowledge of data from class $\dot{y}$. However, it could be estimated from the training sample:

$$t_{\dot{y}} = \min_{\mathbf{x}\in X_{\dot{y}}} \text{matching}(\mathbf{x}, X_{\dot{y}} \backslash \{\mathbf{x}\}). \tag{5}$$

Naively, this calculation requires performing regular WiSARD training $|X_{\dot{y}}|$ times, as a leave-one-out rotation of the data sample. This means a $O(|X_{\dot{y}}|^2 \delta\beta)$ time complexity. This quadratic relation to the size of the data set would reduce WiSARD usual applicability for larger data sets. Therefore, it would be interesting to avoid its establishment. This was possible through the exploration of some properties of this model.

In order to reduce the computational cost of $t_{\dot{y}}$ calculation, it is proposed a modification of WiSARD training procedure to embed such calculation, avoiding to perform it separately. Equation (5) hints to compute the matching of each observation in $X_{\dot{y}}$, one at a time. As a matter of fact, this can be realized collectively, keeping track of addresses obtained from observations in $X_{\dot{y}}$ but not shared between them. This enables to compute Eq. (6a) efficiently, and subsequently to provide a specialized redefinition of matching: Eq. (6b).

$$\text{exclusive}(\mathbf{x}, X_{\dot{y}}) = \{i : \nexists_{\mathbf{x}'\in X_{\dot{y}}\backslash\{\mathbf{x}\}} \text{addressing}_i(\mathbf{x}) = \text{addressing}_i(\mathbf{x}')\}; \tag{6a}$$

$$\text{matching}(\mathbf{x}, X_{\dot{y}} \backslash \{\mathbf{x}\}) = 1 - \frac{1}{\delta} |\text{exclusive}(\mathbf{x}, X_{\dot{y}})|. \tag{6b}$$

Algorithm 2 describes the modified training procedure of WiSARD. In a comparison to its original version (Algorithm 1), there are basically two changes. First, every time an address is to be written, its 'ownership' status is updated (Sects. 3.1, 3.2). Second, after all addresses are written, a loop over all exclusive addresses (i.e., those related to a single observation in the training set) is used to compute incrementally $|\text{exclusive}(\mathbf{x}_i, X_{y_i})|$ for all observations (Sect. 10). The additional operations represent an increase of the computational cost of WiSARD training, but its time complexity remains $O(|X|\delta\beta)$. That is as good as possible in this case. After this procedure is concluded, $\text{EXCLUSIVE}_i = |\text{exclusive}(\mathbf{x}_i, X_{y_i})|$.

Equation (6b) and, consequently, Eq. (5) can be easily calculated based on array EXCLUSIVE. This leads to a definition of thresholds strongly oriented to avoid mistaken rejections. This way, no element of the training sample would be incorrectly ruled as extraneous if it had not been considered during training. Such setting is useful, but in some situations mistaken rejections may be preferred to wrong associations of extraneous data to targeted classes. For example, to reject few observations of a targeted class in order to correctly identify a large amount of outliers is generally interesting. Furthermore, training data may be contaminated with incorrectly labeled observations, whose influence on threshold definition should be as small as possible. Figure 5 contrasts these positions.

Thus, for a more flexible rejection criterion, Eq. (7) was used as an alternative to Eq. (5). P_α denotes the α-th percentile of the considered values. Variable $\alpha \in (0, 100)$ is a model

Let OWNER be an empty dictionary
for all pairs $(\mathbf{x}_i, y_i)$, the train sample **do**
  for all addresses a_j in addressing($\mathbf{x}_i$) **do**
    if $a_j \notin \Delta_{y_i,j}$ **then**
      $\Delta_{y_i,j} \leftarrow \Delta_{y_i,j} \cup \{a_j\}$
      $\text{OWNER}_{y_i,j,a_j} \leftarrow i$ ▷ Adding a new dictionary entry
    else
      Remove entry OWNER_{y_i,j,a_j} ▷ Address is not exclusive
Let EXCLUSIVE = $0_{|X|}$ be an array of $|X|$ zeros
for all $\langle (y_i, j, a_j), i \rangle$, entries in OWNER **do**
  $\text{EXCLUSIVE}_i \leftarrow \text{EXCLUSIVE}_i + 1$

Algorithm 2: WiSARD training procedure, modified to track exclusive addresses

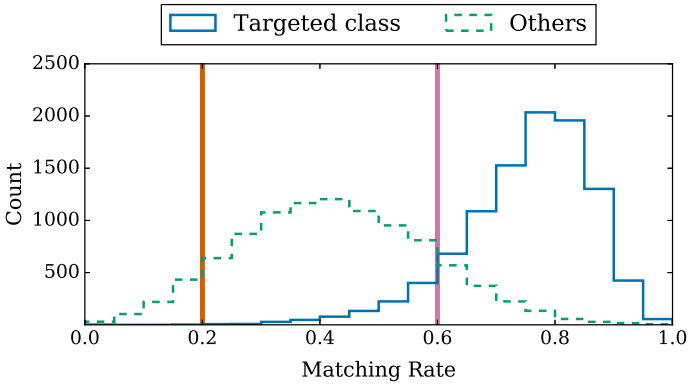


Fig. 5 Two histograms depicting the distribution of matching rates for hypothetical data. The one with solid contour regards data from targeted class, $\{\text{matching}(\mathbf{x}, X_{\dot{y}} \backslash \{\mathbf{x}\}) : \mathbf{x} \in X_{\dot{y}}\}$, while the other one regards the remaining data, $\{\text{matching}(\mathbf{x}, X_{\dot{y}}) : \mathbf{x} \in X \backslash X_{\dot{y}}\}$. The *vertical lines* represent threshold values, which would lead to the rejection of observations to their *left*. Using Eq. (5), the 0.2 threshold would be chosen. This assures that no $\mathbf{x} \in X_{\dot{y}}$ would be considered extraneous to its class. However, the 0.6 threshold could be preferred despite some bad rejections it would make, because of the much greater portion of observations extraneous to $\dot{y}$ it would rightfully reject

parameter. The definition of rejection thresholds based on percentiles, which are robust statistics, is interesting in the light of the bias-variance trade-off. As an alternative approach, a combination of mean and standard deviation (e.g., the three sigma rule) could be used here. However, this could enhance the influence of noise and outliers on thresholds definition.

$$t_{\dot{y}} = \underset{\mathbf{x} \in X_{\dot{y}}}{P_\alpha} \text{matching}(\mathbf{x}, X_{\dot{y}} \backslash \{\mathbf{x}\}) \tag{7}$$

The combination of Algorithm 2 and Eqs. (6b), (7) provides a rejection criterion based on what can be inferred about a class from its own observations only. This is particularly interesting for situations in which all training data concerns a single, targeted class, as in various unary classification tasks. Even in this scenario it is still possible to use α to control rejection tendency.

3.2 Optimal thresholding

The manual thresholding scheme which was just described defines $t_{\dot{y}}$ using no observation besides those from $X_{\dot{y}}$. However, there is no reason to avoid employing observations from $X \setminus X_{\dot{y}}$ to establish a rejection criterion if those are available. Moreover, to use data from other classes looks reasonable considering that such data is extraneous to class $\dot{y}$ and should be rejected accordingly. In other words, to reject observations of the targeted classes which would be otherwise misclassified is just another perspective of the same original goal.

Ideally, $t_{\dot{y}}$ would be set so that

$$\forall \mathbf{x}\ \text{matching}(\mathbf{x}, X_{\dot{y}}) \geq t_{\dot{y}} \iff f(\mathbf{x}) = \dot{y}. \tag{8}$$

Such condition wherein the rejection threshold establishes a perfect dichotomy of the observations possibly related to class $\dot{y}$ is generally infeasible. That is because it is quite common to have some observations truly related to $\dot{y}$ but with a low matching value, while the opposite happens for some elements of other classes. Therefore, instead of looking for such unrealistic threshold, finding the best value for $t_{\dot{y}}$ according to some measure of classification effectiveness was the alternative used in this regard. This can be enunciated similarly to an optimization problem:

$$\begin{aligned} \underset{t_{\dot{y}}}{\text{maximize}} \quad & \alpha'(\text{LABELS}, \text{PREDICTIONS}); \\ \text{subject to} \quad & \text{LABELS}_i = [f(\mathbf{x}_i) = \dot{y}], \\ & \text{PREDICTIONS}_i = [\text{matching}(\mathbf{x}_i, X_{\dot{y}}) \geq t_{\dot{y}}]. \end{aligned} \tag{9}$$

Equation (9) is defined according to the binary classification task of ruling if observations as $\mathbf{x}_i$ are related to class $\dot{y}$ or not. LABELS is an array which represents the ground truth of such task. PREDICTIONS indicates the labels inferred according to matching computation and a given $t_{\dot{y}}$. Here α' represents the aforementioned measure of classification effectiveness. Previously (Fumera et al. 2000) only accuracy was considered to guide thresholds adjustment. However, any method to rate prediction quality can be employed for this: for example, F-measure (Goutte and Gaussier 2005). This way, α' would represent a model parameter which plays the same role of parameter α introduced in the previous subsection. However, α'-based thresholds computation uses training data classification [Eq. (8)], instead of relying just on matching rates of these observations [Eq. (7)].

Still in the same regard, consider $\alpha'_{\dot{y}}(\text{LABELS}, \text{PREDICTIONS})$ the objective function of Eq. (9) with respect to class $\dot{y}$. A single objective function regarding the optimization of all thresholds, respective to each known class, is described by Eq. (10). This way, all observations in the training set can be used to define the rejection threshold of a class, instead of its observations only (Fischer et al. 2016). Such scheme provided better results in the performed experiments, what could be possibly related to the difference between open set recognition and classification with rejection-option: the first requires a global notion of the uncertainty with respect to ruling an observation as an element of a known class, while the second is focused on minimizing the cost resulting from misclassifications.

$$\sum_{\dot{y}} \alpha'_{\dot{y}}(\text{LABELS}, \text{PREDICTIONS}) \tag{10}$$

The idea here is to obtain a reasonable $t_{\dot{y}}$ by solving Eq. (9) just for the training sample. That is, each of the mentioned $\mathbf{x}_i$ is an observation of X which would be classified with respect to $\dot{y}$. Then, the search for the optimal value of $t_{\dot{y}}$ can be limited to all matching$(\mathbf{x}_i, X_{\dot{y}})$ values. Again, Algorithm 2 is used for training in order to avoid performing explicitly the

leave-one-out rotation of the data set. Subsequently, Algorithm 3 is carried out to tackle the aforementioned optimization problem. At last, considering that number of targeted classes is denoted by $|\dot{Y}|$, the time complexity of training becomes $O(|X||\dot{Y}|\delta\beta)$. This related to the fact that the loop starting at Sect. 3.2 of Algorithm 3, which dominates the computation of $t_{\dot{y}}$, can be performed in $O(|X|\delta\beta)$ steps.

1: Let $\dot{y}$ be the targeted class whose optimal threshold $t_{\dot{y}}$ is to be computed
2: **for all** $\mathbf{x}_i \in X$ **do**
3: $\quad$ LABELS$_i \leftarrow [f(\mathbf{x}_i) = \dot{y}]$
4: **for all** $t : \exists_{\mathbf{x} \in X}$ matching$(\mathbf{x}, X_{\dot{y}}) = t$ **do**
5: $\quad$ **for all** $\mathbf{x}_i \in X$ **do**
6: $\quad\quad$ PREDICTIONS$_i \leftarrow [\text{matching}(\mathbf{x}_i, X_{\dot{y}}) > t]$
7: $\quad$ SCORE$_t \leftarrow \alpha'$(LABELS, PREDICTIONS)
8: $t_{\dot{y}} \leftarrow \text{argmax}_t$ SCORE$_t$

Algorithm 3: Threshold optimization procedure

As already mentioned, each class-related rejection threshold is defined according to the best solution of a binary classification subtask. Such solution may vary according to which measure α' is picked to evaluate classification effectiveness. The choice of α' should consider that, for any of these subtasks, class '1' is the targeted class, while class '0' just gathers misclassified observations (i.e., $f(\mathbf{x}_i) \neq \dot{y}$): comparing extreme scenarios, it is better to reject no observation, as the original WiSARD does, than to reject them all, including elements of the targeted classes.

Measures as accuracy are indifferent to distinct roles the classes may have, while others like F-measure are calculated based on a positive (in other words, targeted) class. Consequently, measures of the last kind should be preferred for this use. Still with respect to F-measure, its parameter β can be used to control how prone to rejection is the system: if precision is prioritized, by setting $\beta < 1$, there is a stronger rejective tendency; otherwise, if recall is favored, rejections should occur less frequently. This is similar to setting the cost of a single rejection, as commonly seen in the literature (Herbei and Wegkamp 2006; Fischer et al. 2016). The F_1 score (i.e., $\beta = 1$), which considers precision and recall equally important, was the default standard for threshold optimization used in this research. In this case, a mistaken rejection is considered half as bad as a wrong classification.

4 Experimental evaluation

In this section a collection of learning tasks with open-set premises are presented. These are accompanied by the results obtained when they were approached with rejection-capable WiSARD-based systems which follow the ideas just detailed. Alternative approaches to these tasks, some which can be found in the literature, are used to provide baseline results for comparison. Through these experiments it can be noticed how harmful it is to tackle recognition problems with regular classifiers, ignoring the existence of extraneous data. Indeed, some data sets used here were, before this work, only considered for classification. Therefore, the introduction of each data set is followed by an exposition of its open-set nature.

Aiming to provide a rich description of each task, a measure of the coverage of all classes by the training samples is indicated together with other relevant information. Class coverage, proposed here as shown in Eq. (11), is a measure in the same spirit of openness. However, the first can be seen as an improvement over the last one, considering the following reasons: by definition, it is assured that coverage $\in [0, 1]$; and it is reasonable to relate a greater number of targeted classes to a smaller need for rejection. This second point is consistent with the fact that classes to be recognized are expected to be comprehensively detailed in the training sample. This way, they help to portrait the task domain more precisely than available data from other classes.

$$\text{Coverage} = \sqrt{\frac{C_r + C_t}{2C_e}}. \tag{11}$$

4.1 Closed-set versus open-set anomaly detection

The 'DGA' data set (Mirowski and LeCun 2012) regards power transformers in one of two possible states: operating regularly, as desired, or in the imminence of failure. The challenge here is to rule if a transformer is faulty or normal, according to the concentration of 7 gases dissolved in its insulation oil. This is a small data set, composed of 50 'normal' and 117 'faulty' observations. Originally this data set was used for classification, so that previously reported results were obtained considering random train-test data splits.

However, it makes sense to consider the existence of a single normal state, opposed to various abnormal, faulty ones: power transformers can deviate from their standard functioning in many ways. In practice, it is impossible to guarantee that all possible abnormal conditions are known a priori. An accurate reproduction of the concrete task related to the DGA data set should feature such incompleteness of the training sample. Since plain random partitions of the data set do not ensure such condition, a suitable alternative to those was employed: Algorithm 4 describes how train-test splits in the aforementioned mold were generated; in short, instead of single faulty observations, clusters of them were split into the training and test samples.

Let KMEANS$(X, n) = \{C_1, \ldots, C_n\}$ be a partition of X in n clusters
function MAKESPLITSDGA(data set X, $s \in \{1, \ldots, 9\}$)
  SPLITS $= \varnothing$ ▷ SPLITS is a set of train-test splits of X
  $C \leftarrow$ KMEANS$(X_{\text{faulty}}, 10)$ ▷ C is a partition of all faulty observations in clusters
  Let $SC = \{C \text{ choose } s\}$ be the set of all s-combinations of clusters in C
  for all $SC_i \in SC$ **do** ▷ SC_i is a collection of clusters of faulty observations
    $T_{\text{faulty}} \leftarrow \cup_j SC_{ij}$ ▷ SC_{ij} is the jth cluster in SC_i
    Let T_{normal} be a random 80% excerpt of X_{normal}
    $T \leftarrow T_{\text{normal}} \cup T_{\text{faulty}}$
    SPLITS $\leftarrow$ SPLITS $\cup \{(T, X \backslash T)\}$
  return SPLITS

Algorithm 4: Generator of train-test splits of the DGA data set

The class coverage of the sample partitions provided by function MAKESPLITSDGA varies according to its parameter s: if each cluster of faulty observations C_i is considered a class, a lower s means a smaller number of classes in each training set T. Consequently, it also means more classes in its testing counterpart $X \backslash T$. To assess the influence of coverage

Table 2 Characteristics of tasks based on the DGA data set

Characteristics	Tasks			
	$s = 2$	$s = 5$	$s = 8$	Fivefold CV
# Train-test splits	4500	25,200	4500	5000
Targeted classes (C_r)	1	1	1	2
Known classes (C_t)	3	6	9	2
Existing classes (C_e)	11	11	11	2
Coverage (%)	43	56	64	100

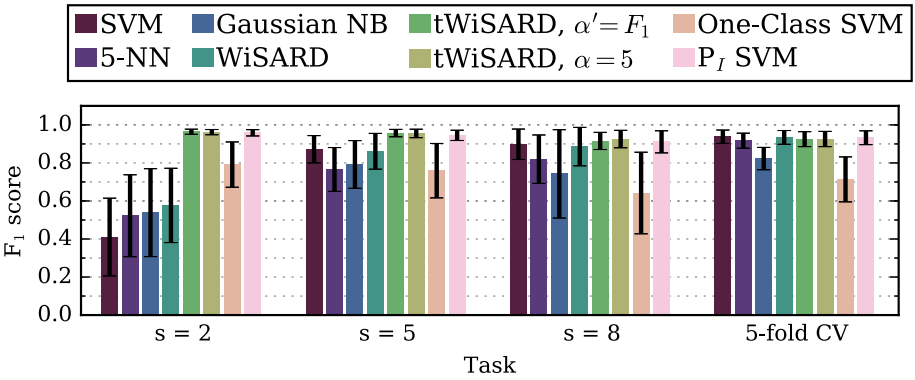


Fig. 6 Results for the tasks based on the DGA data set

in this task, different values of s were used: 2, 5 and 8. For each of these three values, MAKESPLITSDGA was called 100 times, generating a mass of partitions of the original data set. Additionally, 5000 splits from random fivefold cross validation settings were also used, for the sake of comparison to a closed-set classification scenario. The reported results regard each train-test split in the 4 groups just described. Table 2 summarizes the information about these groups.

Two tWiSARD ('t' stands for threshold) versions were tested: one using the manual thresholding scheme, with $\alpha = 5$; and another whose thresholds were optimized according to $\alpha' = F_1$ score; other parameters of both were set as $\beta = \delta = \gamma = 100$. It is also reported the performance of the following alternatives, with respective parameter setups: a 5 nearest neighbors classifier; a Gaussian Naive Bayes classifier; a SVM and a 1-vs-all P_I SVM, both with $C = \gamma = 10$; a one-class SVM, with $\nu = 0.005$ and $\gamma = 0.025$; a WiSARD with $\beta = \gamma = 100$ and $\delta = 10$. These settings were obtained in a best-effort search and provided optimal results. P_I SVM (Jain et al. 2014) represents the state of art regarding open set recognition.

Figure 6 illustrates the results of this first experiment. It shows four bar groups, related to each task based on the DGA data set. From left to right, the tasks are ordered from the lowest to the highest coverage. This way, it is possible to observe some patterns related to such variation. For example, the overall performance grows with coverage, what is expected using richer training data. All regular classifiers (the first four alternatives) obey this trade-off. On the other hand, the one-class SVM, a rejection-oriented method, best performed in the lowest coverage scenario. The three rejection-based methods stand out among the rest,

producing top results regardless of the coverage level. This is an interesting evidence in favor of the unrestricted use of methods for open set recognition, even when coverage could be considered high, or for any classification-like task.

Statistically, both tWiSARD versions excel: according to Wilcoxon signed-rank tests with a significance level of 0.01, they were superior to any other tested alternative in all three open-set scenarios. However, in the fivefold CV setting, SVM, WiSARD and P_I SVM were, by a thin margin, the top performers. Despite this fact, it would be reasonable to choose any of the two tWiSARD alternatives to be used for a recognition task based on the DGA data set wherein the coverage level was unknown: on average, they produced the best results of this experiment. At last, in three of the four tasks tWiSARD with $\alpha' = F_1$ score performed as well or better than tWiSARD with $\alpha = 5$ for most of the train-test splits.

4.2 Open set recognition with multiple targeted classes

It was just shown how a two-class classification task may be better interpreted as an open set recognition problem, with a single targeted class. This is also possible in scenarios with more than two classes, what requires the discrimination between classes of interest as well as the identification of data extraneous to all of them. These two goals are conflicting in some way: observations which would be correctly classified can be mistaken as foreign data. Therefore, it is necessary to find an equilibrium to avoid spoiling good class predictions while still rejecting accurately. An interesting question in this regard is: can such balance be found using data from the targeted classes only, without using extraneous data during training? This was analyzed through the experiment described next.

For such purpose, the 'UCI-HAR' data set (Anguita et al. 2013) was employed. It is, quoting its authors, "an Activity Recognition database, built from the recordings of 30 subjects doing Activities of Daily Living (ADLs) while carrying a waist-mounted smartphone with embedded inertial sensors". Each observation is a collection of 561 statistics of the sensor readings. However, in this work just a subset of 46 attributes was used: those related to the mean of the readings. This data set is composed of over ten thousand elements, each of them related to one of six activities (i.e., the classes): 'Walking', 'Upstairs', 'Downstairs', 'Sitting', 'Standing' and 'Laying'.

As the DGA data set, the UCI-HAR data set was first used for classification. This way, each of the six classes was represented in both training and test samples. However, in practice, activities beside those known a priori can be realized in an unprecedented way (Hu et al. 2013), and they should be recognized as so. In order to mimic a realistic human activity recognition task, in which not all possible activities are known and modelled, each of the six classes was omitted at a time from training: the train-test splits of the data set were defined by a total of 40 fivefold cross-validation runs; each of the 200 test sets was processed six times, considering the same respective train sets, except for the class left out. Thus, in each train-test round, $C_r = C_t = 5$, $C_e = 6$ and, consequently, coverage $\approx 91\%$.

The same group of methods compared in the anomaly detection experiments is employed here, except for the one-class SVM, which can not handle multiple classes. These methods are enumerated next, with respective parameter setups: a 5 nearest neighbors classifier; a Gaussian Naive Bayes classifier; a WiSARD classifier; two tWiSARD versions, one with $\alpha = 10$ and another with $\alpha' = F_{2.5}$ score; a SVM; and a 1-vs-all P_I SVM, with $P = 0.4$; Both SVM and P_I SVM were set with $C = 1000$. WiSARD and both tWiSARD were set with $\beta = 50$, $\delta = 200$ and $\gamma = 20$.

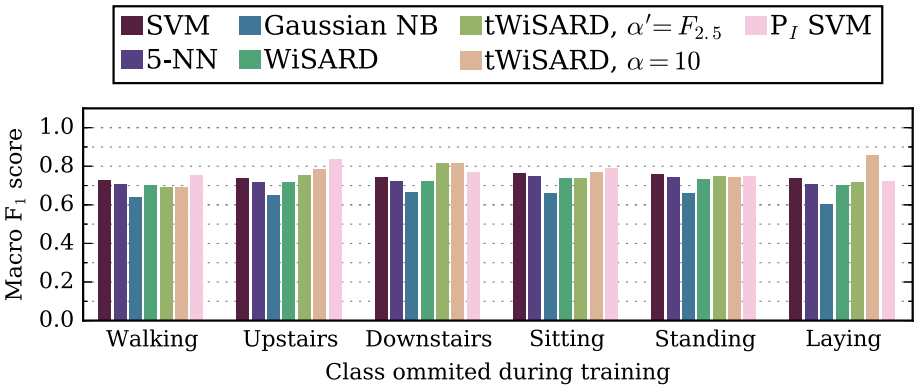


Fig. 7 Results for the tasks based on the UCI-HAR data set. *Error bars* were omitted because deviations were negligible

The UCI-HAR data set features some class imbalance: 18.8% of the data is related to the most frequent class, while 13.6% belongs to the least frequent one. Despite this difference, all six classes can be considered equally important in the task domain. In order to avoid taking this data set condition into account on the evaluation of the provided predictions, the Macro F_1 score (Sokolova and Lapalme 2009) was chosen as performance metric for this task. Such choice is explained by the fact this metric is insensitive to class imbalance: the assignment of elements to each class can be seen as a separate binary classification problem, with true and false positives, as well as negatives; the Macro F_1 score is the average of the F_1 scores of these sub-problems.

The results of the experiment with the UCI-HAR data set are portrayed in Fig. 7. Each bar group is associated to one collection of train-test rounds in which a class was left out of the training sample. On most cases, the rejection-capable methods had better performances than their regular counterparts: both tWiSARD versions edged the WiSARD classifier on 5 of the 6 tasks, while the same happened for P_I SVM and the regular SVM on the first 4 tasks. For all cases, except for that of class 'Standing', one of the last three alternatives was the best performer. These can be seen as evidences which support to take specific care of extraneous data in situations like the one represented by the UCI-HAR data set. The superiority of the methods for open set recognition was verified in this setting with relatively high coverage (91%). This performance difference could be expected to increase when dealing with lower coverage, as shown in the test with the DGA data set.

According to Wilcoxon signed-rank tests with a significance level of 0.01, tWiSARD with $\alpha = 10$ had the best results overall. This can be partially credited to its distinct performance when the 'Laying' class was considered extraneous. The explanation for such outcome is the following: when trying to reject elements of the 'Laying' class, which is the most dissimilar of all, each individual rejection is more likely to be correct; this way, a more rejection-prone criteria should perform better in this case. This is confirmed by Table 3: when rejecting the 'Laying' class, tWiSARD with $\alpha = 10$ was the uncontested best alternative regarding not only extraneous-data recall, which grows with rejection tendency, but also precision. This table also shows that on average both tWiSARD versions were superior to P_I SVM rejection-wise. Still in this regard, P_I SVM was almost entirely ineffective to reject classes 'Standing' and 'Laying', what opposes tWiSARD performance.

Table 3 Rejection performances for tasks based on the UCI-HAR data set

Omitted class	Precision			Recall		
	tWiSARD $\alpha' = F_{2.5}$	tWiSARD $\alpha = 10$	P_I SVM	tWiSARD $\alpha' = F_{2.5}$	tWiSARD $\alpha = 10$	P_I SVM
Walking	0.029	0.185	0.368	0.004	0.111	0.107
Upstairs	0.464	0.459	0.700	0.153	0.479	0.404
Downstairs	0.661	0.518	0.342	0.406	0.672	0.114
Sitting	0.201	0.373	0.388	0.030	0.279	0.125
Standing	0.341	0.288	0.021	0.094	0.173	0.004
Laying	0.464	0.706	0.000	0.062	0.999	0.000
Average	0.360	0.421	0.303	0.125	0.452	0.126

4.3 Open set recognition with very low coverage

The concept of coverage was defined to provide a quantitative degree of complexity of open-set problems. It looks reasonable to rate this according to the number of classes represented in the training sample compared to those to be handled during the effective use of the consolidated knowledge. The DGA and UCI-HAR data sets, originally considered for classification, were used to define tasks with coverage under 43 and 91% respectively. This last experiment is an interesting benchmark, designed specifically for open set recognition, with coverage under 20%.

The 'LBP88' data set[3] is composed by elements from two image sets, Caltech 256 (Griffin et al. 2007) and ImageNet (Deng et al. 2009). The first was used to provide train data, while the test sets were composed of positive observations of the first source and negative ones from the last. This cross-data set design requires the proper rejection of observations from classes not targeted, independently of its origin. In each of 5 rounds, 88 classes were randomly selected. Each of these 88 classes was used once as the one to be recognized, being represented in the training and test samples by 70 and 30 observations, respectively. The remainder of the training sets were 70 (5×14) observations of 5 classes randomly chosen from the 87 negative classes. In turn, the test sets also had 5 observations from each of the 87 classes not targeted. Adding up, the training and test samples had 140 and 465 observations, respectively. Each observation was described by 59 attributes.

The open-set nature of the LBP88 data set is quite similar to that of the DGA data set. That is, both are used to define tasks in which one class is well-known a priori and should base the decision criterion, while scarce information from other classes can be used in order to refine such criterion. From another point of view, their respective tasks differ with respect to the desired goal and, consequently, the performance evaluation: for anomaly detection, implied by the DGA data set, the goal is to identify elements extraneous to the base class as abundantly and precisely as possible; for the LBP88 data set, the goal is inverted in a certain way, as the identification of elements of the base class is desired.

The same methods compared through the tasks defined using the DGA data set were reused for the LBP88 data set, but with different parameters: a WiSARD classifier; two tWiSARD varieties, one with $\alpha = 50$ and another with $\alpha' = F_{0.4}$ score; a 5 nearest neighbors classifier; a Gaussian Naive Bayes classifier; a SVM, a one-class SVM and a 1-vs-all P_I SVM, all with

[3] http://www.metarecognition.com/openset/ (accessed 2016/03/06), LBP-like Features, Open Universe of 88 Classes.

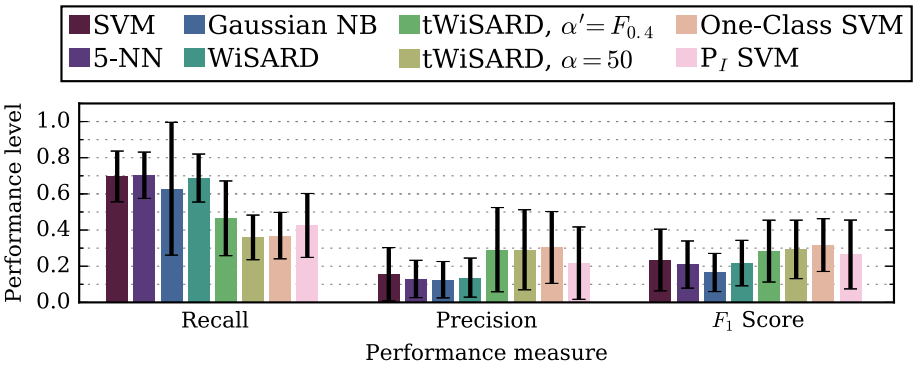


Fig. 8 Results for the experiment on the LBP88 data set

$\gamma = 35$. WiSARD and both tWiSARD were set with $\beta = 100, \delta = 590$ and $\gamma = 1000$. SVM and its variants were set with $\gamma = 35$. P_I SVM was also set with $P = 0.5$.

Figure 8 depicts the results for this experiment, described by three different performance measures: recall, precision and F_1 score. The last measure, which is the harmonic mean of the first two, is the quality standard which should be maximized. However, these three distinct points of view help to highlight some interesting details. Regular classifiers (the first four alternatives) exhibit a higher recall level but also a lower precision level than the rejection-capable methods (the last four alternatives). This happens because regular classifiers have no rejection option. Therefore, they can not mistakenly reject an observation which would be correctly classified despite its dissimilarity to training data. However, this has an expected negative effect on precision level. Consequently, the first quartet had the worse overall results, represented by the F_1 score. Among this last quartet, P_I SVM had the poorest performance. That is, despite achieving a good recall level, the effect of its relatively low precision on F_1 score is noticeable. This can be compared to tWiSARD with $\alpha' = F_{0.4}$, which had the best recall level inside the group just mentioned, but also top results regarding precision and F_1 score.

Considering Wilcoxon signed-rank tests with a significance level of 0.01 over the F_1 scores obtained by the tested methods, the one-class SVM was the single top performer. The proposed tWiSARD with $\alpha = 50$ and with $\alpha' = F_{0.4}$ score had the second and third best results, respectively. However, prioritizing recall over precision, tWiSARD with $\alpha' = F_{0.4}$ score could be considered the best alternative. Still concerning overall performance, it can be noticed that the two best methods (one-class SVM and tWiSARD with $\alpha = 50$) work using data from the targeted class only. This can be seen as an evidence that available information about extraneous classes may be misleading and produce negative effects on performance. In other words, depending on characteristics of the extraneous elements as variety, distribution and others, it may be wiser, safer, to avoid drawing conclusions based on scarce data about those.

5 Conclusion

Classification will always be one of the most difficult, ubiquitous and important machine learning tasks. Open set recognition is a classification-derived task, in which not all classes

are represented in the training set. After training, besides regular classification, examples of the classes not represented in the training set should be properly rejected.

Because of its proximity to classification, some approaches to open set recognition found in the literature were built on top of regular classifiers. While this is not wrong, it requires special attention to the differences between these tasks, which should guide the adaptation of those previously existing methods. The method introduced here, tWiSARD, was developed with such requisite in mind, based on the recognition-friendly WiSARD classifier. Such conception boosts the use of a well-established learning technique in situations where it is necessary to define more strictly the boundaries inside which it is possible to make conscious decisions.

The results of the experiments performed are insightful. They highlight some interesting characteristics of the data which did not emerge during the exclusive use of the classifiers to which the proposed approach was compared. An example of such fact is the variation of the performance of the tested methods in the proposed tasks of anomaly detection, compared to regular k-fold cross-validation. The distinct behavior of regular-classifiers compared to rejection-capable methods in the test scenario featuring low coverage data is another example in this sense.

In general, the proposed methodology was not only effective combining classification with precise identification of extraneous data. It also provided singular points of view of the context modeled from data. Even the comparison of the performances of its manual-thresholding and optimal-thresholding versions was informative: their behavior can be notably different, as evidenced in the multi-class recognition tests, for example. All these facts can be regarded as evidences in favor of the applicability of tWiSARD. Moreover, its superiority compared to other open-set- or rejection-oriented methods was statistically assessed in all three test scenarios considered. This credits such approach as a safe and versatile solution for open set recognition.

Acknowledgements Douglas O. Cardoso thanks Daniel Alves, Diego Souza and Kleber de Aguiar for the valuable suggestions.

References

Aleksander, I., Thomas, W., & Bowden, P. (1984). WiSARD, a radical step forward in image recognition. *Sensor Review*, *4*(3), 120–124.

Aleksander, I., Gregorio, M. D., França, F. M. G., Lima, P. M. V., & Morton, H. (2009). A brief introduction to weightless neural systems. In *ESANN 2009, proceedings of the 17th European symposium on artificial neural networks, Bruges*, April 22–24, 2009.

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *21st European symposium on artificial neural networks, ESANN 2013, Bruges*, April 24–26, 2013.

Bartlett, P. L., & Wegkamp, M. H. (2008). Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, *9*, 1823–1840.

Cardoso, D. O., França, F. M. G., & Gama, J. (2015). A bounded neural network for open set recognition. In *2015 International joint conference on neural networks, IJCNN 2015, Killarney*, July 12–17, 2015 (pp. 1–7). IEEE.

Chen, C., Zhan, Y., & Wen, C. (2009). Hierarchical face recognition based on SVDD and SVM. In *2009 International conference on environmental science and information application technology, ESIAT 2009, Wuhan*, 4–5 July 2009 (Vol. 3, pp. 692–695).

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), Miami, Florida*, 20–25 June 2009 (pp. 248–255).

Fischer, L., Hammer, B., & Wersing, H. (2015). Efficient rejection strategies for prototype-based classification. *Neurocomputing*, *169*, 334–342.

Fischer, L., Hammer, B., & Wersing, H. (2016). Optimal local rejection for classifiers. *Neurocomputing*, *214*, 445–457.

Fumera, G., & Roli F. (2002). Support vector machines with embedded reject option. In *Proceedings of the pattern recognition with support vector machines, first international workshop, SVM 2002, Niagara Falls*, August 10, 2002 (pp. 68–82).

Fumera, G., Roli, F., & Giacinto, G. (2000). Reject option with multiple thresholds. *Pattern Recognition*, *33*(12), 2099–2101.

Goutte, C., & Gaussier, É. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Advances in information retrieval, proceedings of the 27th European conference on IR research, ECIR 2005, Santiago de Compostela*, March 21–23, 2005 (pp. 345–359).

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., & Canu, S. (2008). Support vector machines with a reject option. In *Advances in neural information processing systems 21, proceedings of the twenty-second annual conference on neural information processing systems, Vancouver, British Columbia*, December 8–11, 2008 (pp. 537–544).

Grieco, B. P. A., Lima, P. M. V., Gregorio, M. D., & França, F. M. G. (2010). Producing pattern examples from "mental" images. *Neurocomputing*, *73*(7–9), 1057–1064.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset*. Technical reports 7694, California Institute of Technology.

Hanczar, B., & Sebag, M. (2014). Combination of one-class support vector machines for classification with reject option. In *Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2014, Nancy. Proceedings, part I*, September 15–19, 2014 (pp. 547–562).

Herbei, R., & Wegkamp, M. H. (2006). Classification with reject option. *Canadian Journal of Statistics*, *34*(4), 709–721.

Homenda, W., Luckner, M., & Pedrycz, W. (2014). Classification with rejection based on various SVM techniques. In *2014 International joint conference on neural networks, IJCNN 2014, Beijing*, July 6–11, 2014 (pp. 3480–3487).

Hu, B., Chen, Y., & Keogh, E. J. (2013). Time series classification under more realistic assumptions. In *Proceedings of the 13th SIAM international conference on data mining, Austin, Texas*, May 2–4, 2013 (pp. 578–586).

Jain, L. P., Scheirer, W. J., & Boult, T. E. (2014). Multi-class open set recognition using probability of inclusion. In: *Computer vision—ECCV 2014—13th European conference, Zurich. Proceedings, part III*, September 6–12, 2014 (pp. 393–409).

Khan, S. S., & Madden, M. G. (2009). A survey of recent trends in one class classification. In *Artificial intelligence and cognitive science—20th Irish conference, AICS 2009, Dublin. Revised selected papers*, August 19–21, 2009 (pp. 188–197).

Kolcz, A., & Allinson, N. (1994). Application of the CMAC input encoding scheme in the n-tuple approximation network. *IEE Proceedings-Computers and Digital Techniques*, *141*(3), 177–183.

Linneberg, C., & Jorgensen, T. (1999). Discretization methods for encoding of continuous input variables for boolean neural networks. In *International joint conference on neural networks, 1999. IJCNN '99* (Vol. 2, pp. 1219–1224).

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Mirowski, P., & LeCun, Y. (2012). Statistical machine learning and dissolved gas analysis: A review. *IEEE Transactions on Power Delivery*, *27*(4), 1791–1799.

Scheirer, W. J., de Rezende, R. A., Sapkota, A., & Boult, T. E. (2013). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(7), 1757–1772.

Scheirer, W. J., Jain, L. P., & Boult, T. E. (2014). Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(11), 2317–2324.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437.

Tax, D. M. J., & Duin, R. P. W. (2008). Growing a multi-class classifier with a reject option. *Pattern Recognition Letters*, *29*(10), 1565–1570.

Yuan, M., & Wegkamp, M. H. (2010). Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, *11*, 111–130.

Zhang, R., & Metaxas, D. N. (2006). RO-SVM: Support vector machine with reject option for image categorization. In *Proceedings of the British machine vision conference 2006, Edinburgh*, September 4–7, 2006 (pp. 1209–1218).