

# Generalization Capacity of Handwritten Outlier Symbols Rejection with Neural Network

Harold Mouchère, Eric Anquetil

► **To cite this version:**

Harold Mouchère, Eric Anquetil. Generalization Capacity of Handwritten Outlier Symbols Rejection with Neural Network. Tenth International Workshop on Frontiers in Handwriting Recognition, Université de Rennes 1, Oct 2006, La Baule (France). inria-00104310

**HAL Id: inria-00104310**

**<https://hal.inria.fr/inria-00104310>**

Submitted on 6 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Generalization Capacity of Handwritten Outlier Symbols Rejection with Neural Network

Harold Mouchère\*

Éric Anquetil

IRISA / INSA / CNRS  
Campus Universitaire de Beaulieu  
Avenue du Général Leclerc  
35042 Rennes, France  
{Harold.Mouchere, Eric.Anquetil}@irisa.fr

## Abstract

*Different problems of generalization of outlier rejection exist depending of the context. In this study we firstly define three different problems depending of the outlier availability during the learning phase of the classifier. Then we propose different solutions to reject outliers with two main strategies: add a rejection class to the classifier or delimit its knowledge to better reject what it has not learned. These solutions are compared with ROC curves to recognize handwritten digits and reject handwritten characters. We show that delimiting knowledge of the classifier is important and that using only a partial subset of outliers do not perform a good reject option.*

**Keywords:** Reject options, distance rejection, handwritten symbol recognition.

## 1. Introduction

In handwriting recognition problems the distance rejection of outliers allows to not recognize shapes which have not been learned. Many contexts of classification use it or can be improved by a reject option which allows to identify outliers.

For example, in context free applications like pen-based musical score editor [1, 12] where the user can write a high number of symbols (digits, letters, musical symbols) a specialized classifier is used for each symbol type. But the context does not permit the application to choose the correct specialized classifier. Thus the recognition system uses a cascade of dedicated classifiers. These classifiers must then have the capacity to reject the shapes they must not recognize. For example the digit recognizer have to reject letters and musical symbols. In such problems the shapes that each classifier must reject are well-defined and samples are available.

In the context of numerical field extraction in handwritten mail [3] the classifier must recognize digits and reject the rest of the text. But this reject class can not be sampled and learned as many things can appear in the rest

of the text. The same problem can appear in the collaboration between segmentation task and classification [1, 11]: the classifiers must be able to reject badly segmented patterns to ask another segmentation. In applications where handwritten characters, digits, schemes, symbols, ... can be inputted, the badly segmented patterns can not be sampled as any situations of overlap can appear. In the general field of pen-based human computer interface, if the user writes an unexpected shape (a scrawl) it would be more comfortable if nothing is recognized. In these three contexts the reject class is ill-defined because of the great variety kind of outlier patterns.

The outliers rejection is a very complex task not solved yet [4, 6, 9, 10, 11, 13]. The aim of this paper is to study the capacity of different rejection strategies to deal with the generalization of a learned reject option. Three different cases can be distinguished depending on outliers to reject during the use (generalization phase) compared to those available during the learning phase:

- the reject option is learned with a set of classes A and then the classifier will have to reject these same classes A, it is called the  $A \rightarrow A$  problem;
- the reject option is learned with a set of classes A and the classifier will have to reject another set of classes B, it is called the  $A \rightarrow B$  problem, in a limit case A can be empty;
- the reject option is learned with a set of classes A and the classifier will have to reject both classes from A and B, it is called the  $A \rightarrow A \& B$  problem, this is an intermediate problem between the  $A \rightarrow A$  problem and the  $A \rightarrow B$  problem.

In these three problems the aim is twofold: to maximize the rejection of the outliers and to minimize the rejection of target classes called examples which must be accepted and recognized. This trade-off is measured using two rates: the True Acceptance Rate (TAR) is the rate of target examples accepted and the False Acceptance Rate (FAR) is the acceptance rate computed on outliers. The section 5 uses TAR and FAR in ROC curves [5] to compare the different reject options. Furthermore, while the

---

\*This work is supported by the Brittany Region.

perfect solution is not found, the rejection of outliers will involve the rejection of examples even if they were well recognized by the classifier. So the Performance Rate on target examples must be kept as higher as possible.

To illustrate this work, we study the capacity of two neural networks often used in rejection problems: the Multi-Layers Perceptron (MLP) and the Radial Basis Function Network (RBFN). These two classifiers have different knowledge modeling and so have different rejecting behaviors as explained in section 2.1 and 4.

In previous work [13] we have presented a unified strategy for rejection based on reliability functions with multi-thresholds and a new iterative algorithm to learn these thresholds. This strategy allows to deal with different natures of reject and with different kinds of classifiers. In this work we show that thresholds based strategies permit a better generalization for the distance outliers rejection.

This paper is organized as follows. The next section present a brief state of the art about the used classifiers and the possible rejection strategies. After that the section 3 presents how these different strategies are enrolled. The section 4 discusses the possible generalization of outliers rejection for the different presented reject options. Finally the section 5 presents experimental results with the recognition of on-line handwriting digits rejecting on-line handwriting characters.

## 2. State of the art

We present in this section the MLP and RBFN classifiers and their learning process to highlight their differences. Then, among the two main reject natures [13], the distance reject option choice is justified for the outliers rejection. After that, we present the possible ways to define a distance reject option with these neural networks.

### 2.1. Classifiers

Feedforward neural networks [2] are composed of three or more layers. The first one is the features input and the last one gives a score  $s_c$  for each class  $c$ . These outputs are a linear combination of the activations  $\mu_i$  of the previous layer with weights  $w_{ic}$ :

$$s_c = \sum_i w_{ic} \mu_i. \quad (1)$$

The classification decision is taken by choosing the class  $C_1$  with the higher score  $s_{C_1}$ .

Multi-Layer Perceptrons (MLP) use a sigmoidal activation function in the hidden layers depending on the activation  $\mu_i$  of neurones for the previous layer. So MLP define linear decision boundaries which are opened with one hidden layer and can be closed with two hidden layers. The weights are learned with the gradient descent algorithm [2] using a learning database and a validation database to stop the enrollment process.

Radial Basis Function Networks (RBFN) use Radial Basis Functions (RBF) in their unique hidden layer. They

use the Mahalanobis distance  $d_{\Sigma_i}(\vec{X}, \vec{V}_i)$  where  $\Sigma_i$  is a covariance matrix and  $V_i$  the center of the RBF. There are many ways to learn the RBF and weights of RBFN [2]. We present here one method. One or more prototype of the hidden layer are learned on each classes separately using Possibilistic C-Means [8]. Thus RBF define intrinsic properties of each class. The activation of each RBF is noted  $\mu_i$ . The output layer gives the class scores  $s_c$  which are discriminant properties defining the decision boundaries. The learning process of the RBF and of the output weights need only a learning database.

### 2.2. Rejection natures

We have shown in previous work [13] that there are two mainly reject natures: the confusion reject and the distance reject. Furthermore, the choice of the used reject nature is important depending on the needs of applications.

The aim of the confusion reject is to improve the accuracy of the recognizer by rejecting pattern on which the classifier can strongly make a misclassification. These errors are near the decision boundaries because the two better class scores are nearly equal.

The distance reject allows to delimit the knowledge of the used classifier. In this way, it can reject shapes which do not belong to learned classes. Hence, if a shape is too far from the knowledge it must be rejected.

For the outliers rejection it is clear that the distance reject is more appropriate than the confusion rejection indeed the aim is not to increase the accuracy of the classifier but to delimit its knowledge to reject patterns which have not been learned.

### 2.3. Reject option solutions

A reject option can be done by many different ways. There are two main strategies for outliers rejection. The first uses a rejection class ( $RC$ ) and the second uses reliability functions ( $RF$ ). This section presents how they work and the next section 3 will present how learn them.

#### 2.3.1. Rejection class solution

For this solution called  $RC$ , a rejection class  $c_r$  can be added to the recognition problem as in [3]. Doing so, the reject decision is taken if the reject score  $s_{c_r}$  is higher than other class scores. In some applications there already exists a classifier for target classes, but this  $RC$  solution need to re-learn the recognizer to integrate the rejection class.

#### 2.3.2. Reliability functions solution

This approach do not modify the original classifier i.e. the reject option not need the the re-learning of the target classes. A reliability function  $\psi$  in  $\mathfrak{R}$  depends on the used classifier and of the nature of the wanted reject option. This function allows to determine the reliability you must have in the result of the classifier. The more a pattern must be rejected, the less is the reliability function. In [4, 10]

only one reliability function is used. As explained in [13] in our approach we permit to define a set of  $N$  reliability functions which allow more precision in the reject.

Thus the reject option is defined with a set of  $N$  thresholds  $\{\sigma_i\}$  each one associated to a reliability function  $\psi_i$ . Then to have a reject, all functions must be lower than their respective threshold:

$$\forall i = 1..N, \psi_i \leq \sigma_i. \quad (2)$$

These functions are defined using class scores or internal information as RBF activations. We will use in this paper three different kinds of functions coherent with a distance reject option. The first one is a commonly used function, the second one was made to have a good representation of known outliers and the last one permits a good representation of the recognizer knowledge:

- $\psi_{C1}^{Dist}$  : only one function is define using the class score of the winner class, it is the commonly used function for distance rejection with MLP [10, 4]:

$$\psi_{C1}^{Dist} = s_{C1}, \quad (3)$$

- $\psi_c^{Dist}$  : one function is define per output using RBFN class score  $s_c$  thus examples can be accepted by any classes, these scores contain distance information (cf. equation 1),

$$\psi_c^{Dist} = s_c, \quad (4)$$

- $\psi_i^{Dist}$  : used only with RBFN, one function is define per RBF using its activation  $\mu_i$  if it is the most activated so each RBF has to accept examples form which it is the nearest RBF:

$$\psi_i^{Dist} = \begin{cases} \mu_i & \text{if } i = \text{argmax}_k(\mu_k), \\ 0 & \text{else.} \end{cases} \quad (5)$$

### 2.3.3. Hybrid solution

These two approaches are not incompatible: reliability functions can be used with a classifier including a rejection class. It is the hybrid (*Hyb*) solution. A pattern is then rejected if the rejection class  $c_r$  has the better score or if all reliability functions are lower than their thresholds. It is the solution used in [9] as one threshold decides the classification between target class and known outliers and another threshold is used to reject unknown outliers.

## 3. Learning reject options

As explained in introduction, the distance reject is learned on available outliers of type  $A$ . The database of outliers  $A$  is called  $D_A$  and the database of the well-defined target classes is called  $D_T$ . The database  $D_A$  can be representative of outliers in the  $A \rightarrow A$  problem, non representative or empty for the  $A \rightarrow B$  problem and incomplete for the  $A \rightarrow A \& B$  problem.

### 3.1. Rejection class learning

For this reject option  $RC$ , outliers are a class considered as other target classes. Thus  $D_A$  and  $D_T$  are mixed in a database  $D_{T+A}$ . This database is then used to learn MLP and RBFN classifiers with a rejection class even if there already exists a target classes recognizer. The enrollment of these classifiers are done as explained in section 2.1. This solution need to have outliers example to learn the rejection class, so when outlier samples are not available during the learning phase ( $D_A$  is empty) the  $RC$  solution is impossible.

### 3.2. Thresholds learning

This reject option  $RF$  is used on an already learned recognizer of target classes. So the classifier is first learned with the database  $D_T$  as explain in 2.1. Then the learning of the distance rejection using the reliability functions defined in section 2.3.2 consists of choosing an appropriate set of thresholds  $\{\sigma_i\}$ .

We present in [13] a unified automatic algorithm which is a generic framework for threshold learning for the different natures of reject. Different variants of this algorithm have been developed but in this paper we present specialized variants, the most efficient for outliers rejection.

It is based on reliability functions using both the database  $D_T$  of examples to accept and the outliers database  $D_A$  to reject. This algorithm has one parameter  $\theta$  which is the minimum True Acceptance Rate (TAR) on  $D_T$  wanted. It is compound of four steps:

1. At the initialization, the reliability values  $\psi_i$  of examples and outliers are computed and the thresholds  $\sigma_i$  are sets to reject all examples and outliers;
2. Next steps are repeated while the TAR on  $D_T$  is below the parameter  $\theta$ ;
3. The function `choose` decides the next threshold  $\sigma_i$  to be decreased;
4. The chosen threshold  $\sigma_i$  is decreased by the function `decrease` to accept more examples (and so some outliers).

We define two variants of this Automatic Multi-Thresholds Learning algorithm (AMTL) called AMTL1 and AMTL2. They differ on their aim and strategy, hence on what the functions `choose` and `decrease` do.

For AMTL1, the aim is to find the better trade-off between the rejection of the target classes and the rejection of outliers and thus it uses the reliability function  $\psi_c^{Dist}$ . Then the function `choose` selects the threshold which minimizes the number of outlier accepted to accept one new example and the function `decrease` decreases the threshold to accept all necessary outliers in order to accept one more example.

For AMTL2, the aim is to have the better description of the knowledge of the RBFN. The reliability function

$\psi_i^{Dist}$  allows this fine detail. Furthermore it do not use any information about outliers to better describe target classes. Thus the function `choose` selects the threshold which maximize the density  $d_i$  of examples activating function  $\psi_i$ . The density  $d_i$  of example is defined by considering the necessary relative variation  $\Delta\sigma_i$  of the threshold  $\sigma_i$  to accept half of the  $M_i$  resting examples activating  $\psi_i$ :

$$d_i = \frac{2\Delta\sigma_i}{\sigma_i M_i}. \quad (6)$$

The function `decrease` decreases the threshold to accept one more example.

It must be noticed firstly that AMTL2 do not used the outliers database  $D_A$  and so do not need it. Secondly, to learn the unique threshold of the reliability function  $\psi_{C1}^{Dist}$ , the both variants AMTL1 and AMTL2 will give the same results.

#### 4. Rejection and generalization

The generalization capacity depends on the used classifier and of the reject option. If the *RC* solution is chosen, the reject generalization depends only on the capacity of generalization of the classifiers. MLP look for the better discriminating boundaries and so, if the classes are enough well-defined, MLP with *RC* can have good results in the *A→A* problem. But these boundaries risk to be badly placed for the *A→B* problem. RBFN first find intrinsic properties of classes and then look for optimal boundaries. So RBFN with *RC* solution can expect to have as good results as the MLP with *RC* solution for the *A→A* problem. On the contrary RBFN with *RC* can use the intrinsic description of classes to obtain better results for the *A→B* problem. The common problem of the *RC* solutions is to obtain the wanted TAR. Indeed the obtain TAR depends on the ratio of outliers to target examples. Furthermore *RC* solution is not usable when  $D_A$  is empty.

If the *RF* solution is chosen, any reject rate can be achieved. Indeed the algorithm *AMTL* start with one hundred percent of reject and stop when the wanted reject rate is achieved, accepting more and more target examples. Furthermore, as the *RF* solution is based on a target class recognizer, the results on outlier rejection depend on reliability functions (see section 2.3.2) available for this classifier. On the one hand, with MLP the only available reliability function uses the output scores which are discriminant information. On the other hand, the RBFN use intrinsic knowledge on which reliability functions can be defined. Thus the *RF* solution is expected to be better if it is used with RBFN than used with MLP for both the *A→A* problem and the *A→B* problem. Furthermore this *RF* solution with AMTL2 can be without any outliers information and thus we expect it would have good results in the generalization *A→B* problem.

The hybrid solution *Hyb* allows to solve the problem of having the wanted TAR with *RC*. It can achieve a TAR from the one obtain with *RC* to 0, but can not accept more examples than those accepted by *RC*. By the same way,

the hybrid solution can not improve the performance of *RC* solutions.

#### 5. Experimental Results

The aim of these experimental results is helping the user to choose the appropriate reject option for his problem. To compare the results of the different reject options for the *A→A* problem, the *A→B* problem and the *A→A&B* problem we use Receiver Operating Characteristics (ROC) curves as defined and explained in [5]. Each point on these curves (called operating points) corresponds to a classifier with a TAR and FAR, the perfect classifier being on top left of TAR-FAR space with 100% of TAR and 0% of FAR. These operating points can be compared with the random reject option which is the line where TAR equals FAR. The comparison of two operating points where none of them are nearer the perfect point than the other depends on the user constraints.

We placed the experiments in the domain of handwritten on-line digits recognition in a context free application where digits and characters can be inputted. So the target classes database  $D_T$  is composed of the 10 digits. We use the UNIPEN [7] digits database split in a learning database and a test database. The outliers in these experiment are the 26 characters of the Ironoff [14] characters database. We need a database  $D_A$  of known outliers and a database  $D_B$  of unknown outliers, thus the classes 0 to 12 ('a' to 'm') are use for  $D_A$  and classes 13 to 25 ('n' to 'z') are used for  $D_B$ . Note that  $D_A$  is split in a learning database and a test test one but for  $D_B$  we only need a test database. The table 1 gives the sizes of all these databases.

**Table 1.** Database sizes and contents.

Name	Classes	Learning size	Test size
$D_T$	'0' to '9'	6156	6714
$D_A$	'a' to 'm'	2649	1769
$D_B$	'n' to 'z'	-	1786

The RBFN and MLP recognizers use a set of 21 on-line features. The RBFN have two RBF per class and the MLP have one hidden layer with 20 neurones. Others structures have been testing without changing main results.

The different reject options are learned as explained in section 3. The exposed algorithms generate one operating point on ROC curves. To obtain different operating points with the *RC* solution (section 2.3.1) we use different proportion of the  $D_A$  database (1, 0.9, 0.75, 0.5 and 0.25). These different operating points are represented by crosses '×' for RBFN and '+' for MLP classifiers. The *RF* solution (section 2.3.2) allows to have a complete curve from 0% to 100% rejection by varying the  $\theta$  parameter of the AMTL algorithm, thus these operating points are represented by lines. Solutions using multi-threshold are denoted by 'AMTL1' (with functions  $\psi_c^{Dist}$ ) and 'AMTL2' (with functions  $\psi_i^{Dist}$ ) depending on the learning algorithm (section 3.2). The *RF* solution

using only one threshold (with function  $\psi_{C_1}^{Dis}$ ) are denoted '1Th'. The *Hyb* solutions (section 2.3.3) use a *RC* classifier and so is represented by a line from 100% of reject to the cross of the corresponding *RC* classifier.

### 5.1. $A \rightarrow A$ problem

The figure 1 presents the ROC curves for the  $A \rightarrow A$  problem. The FAR is so computed on the  $D_A$  test database. So these curves show how much the different reject options can learn to reject a specific class of outliers.

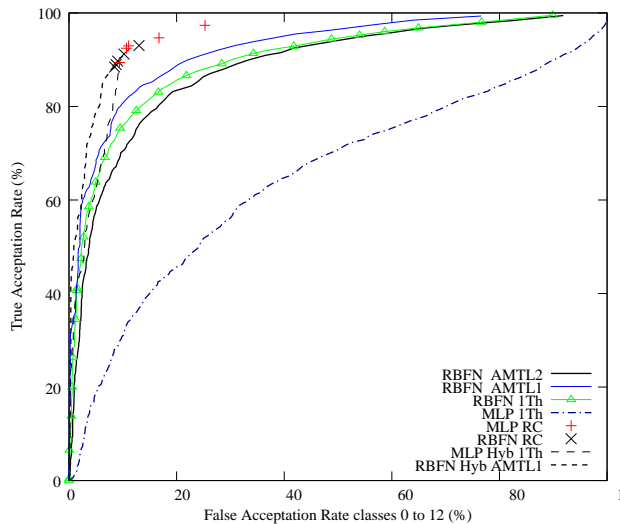


Figure 1. ROC curves for  $A \rightarrow A$  problem.

The results of *RC* solutions with MLP and RBFN are ones with the better operating points. They achieve FAR from 25% to 8%. The *Hyb* allows to improve this rate up to 0%. The *RF* solution achieves lower operating points but with a FAR from 0% to 100%. It must be noticed that the solution using AMTL1 achieves better operating points than AMTL2 and 1Th. Indeed the use of multi-threshold knowing the outliers (AMTL1) allows to better learn the rejection boundaries than not knowing outliers (AMTL2) or use only one threshold (1Th). The *RF* solution with MLP is by far the worst solution, it shows that MLP outputs do not include distance information.

So for the  $A \rightarrow A$  problem the *RC* solutions are the best ones but *RF* solution with AMTL1 could be a solution if special reject rate are expected or if a target classifier already exists.

### 5.2. $A \rightarrow B$ problem

For the  $A \rightarrow B$  problem, the reject options are enrolled by exactly the same way. So they are identical to previous ones but they are tested on the  $D_B$  database. The figure 2 presents these results.

The *RF* solutions with RBFN perform the better trade-off for the  $A \rightarrow B$  problem. Furthermore the solutions without known outliers (RBFN AMTL2 and RBFN Th1) have better results than using known outliers in the learning phase. The *RC* solutions can not achieve FAR lower than

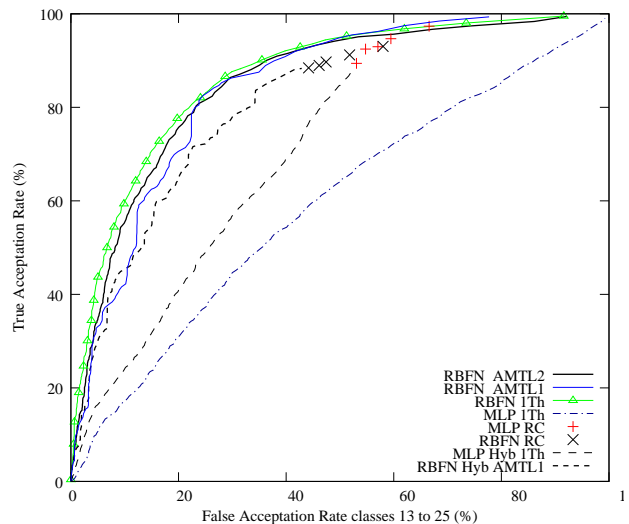


Figure 2. ROC curves for  $A \rightarrow B$  problem.

44% which can be improve with *Hyb* solutions.

For the  $A \rightarrow B$  problem it is better to not use any knowledge about outliers (*RF* AMTL2 or 1Th) than to use ill-defined knowledge (*RC* or AMTL1).

### 5.3. $A \rightarrow A \& B$ problem

The  $A \rightarrow A \& B$  problem is closer to some real situations of use than the two previous problems and is an intermediate problem. The results tested on the  $D_{A+B}$  database are presented in figure 3.

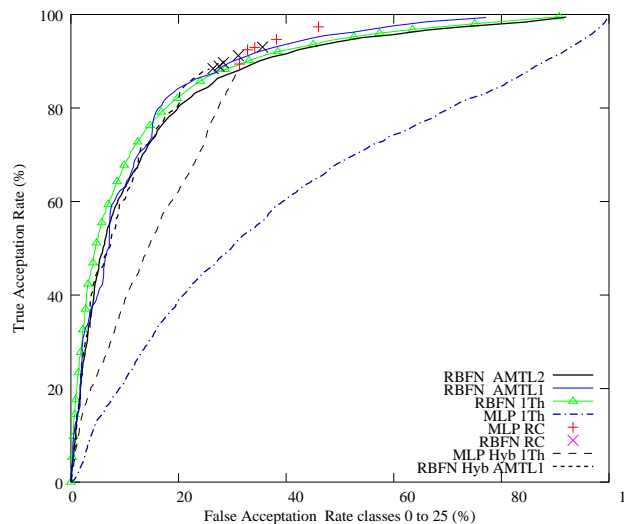


Figure 3. ROC curves for  $A \rightarrow A \& B$  problem.

The *RC* solutions have better results as in the previous problem but do not achieve less than 26% of FAR. The solution using RBFN and thresholds have close results even if RBFN ATML1 have better results for TAR higher than 80%. The solution MLP 1Th and MLP Hyb are again the worst.

Thus knowledges about outliers are not necessary to have good results for the  $A \rightarrow A \& B$  problem and delimiting

the recognizer knowledge improve the trade-off.

#### 5.4. Performance

The different reject options reject also target classes as the TAR is not at 100%. Thus the performance of the recognizer is going down if the reject option reject equally well recognized example and misclassified examples. The figure 4 show the performance on database  $D_T$  of the recognizer with reject option versus their FAR on database  $D_B$ . This FAR is chosen because all solutions have no information about these classes B (so the problem is of the same difficulty for every one) and FAR has no direct influence on performance (contrary to TAR).

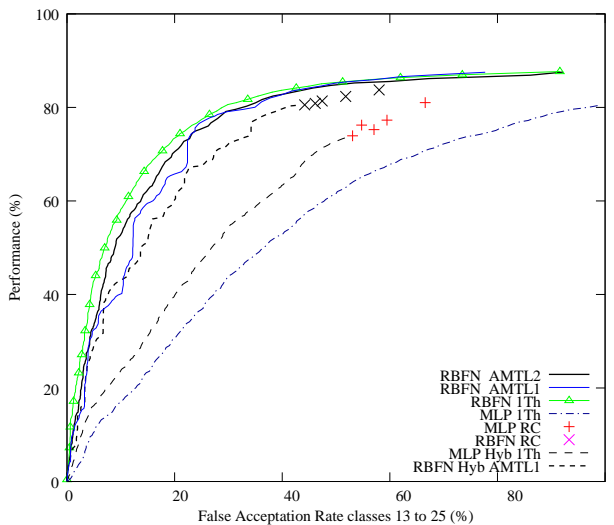


Figure 4. Performance.

The *RF* solutions with RBFN permit to keep a higher performance than others solutions (the better being the RBFN 1th). It means that for the same FAR they reject more errors (misclassified examples) than other solutions.

#### 6. Discussions and Conclusion

The generalization capacity of different outliers rejection options are compared in this study. We define three different situations: the  $A \rightarrow A$  problem, the  $A \rightarrow B$  problem and the  $A \rightarrow A \& B$  problem depending of the available outliers during the learning phase. The different solution do not perform equally in these three situation. Furthermore they do not imply the same cost in term of performance reduction.

If the outliers are well defined the better solution is to introduce a rejection class in the recognizer (*RC* solution) but this solution imply also the biggest performance decrease. If there already exists a classifier of target classes, then the solution using RBFN with reliability function defined on its outputs scores (*RF* solution with RBFN AMTL1) is a good trade-off and allows to achieve more operating points.

If the outliers are unknown or very badly-defined, the better solution is to well define the knowledge of the clas-

sifier, hence to use reject options which do not need any outlier samples or which have a good generalization capacity. The solutions using RBFN and reliability functions are the best ones (RBFN 1th and RBFN AMTL2).

For more real problems where some outliers can be sampled but not completely, different solutions are usable but using reliability functions with RBFN allows more operating points and the AMTL1 delimiting the classifier knowledge using information about outliers is the best.

At the end of this study, a fact that stands forth very clearly is that the robustness of the outliers rejection depends on the capacity of the reject option to delimit the knowledge of the classifier.

#### 7. Acknowledgments

The authors would like to thank Guy Lorette, Professor at the University of Rennes1, for his precious advice.

#### References

- [1] E. Anquetil, B. Couasnon and F. Dambreville, "A Symbol Classifier able to Reject Wrong Shapes for Document Recognition Systems", *GREC'99*, 1999.
- [2] C. Bishop, *Neural Network for Pattern Recognition*, Oxford University Press, 1995.
- [3] C. Chatelain, L. Heutte and T. Paquet, "Segmentation-Driven Recognition Applied to Numerical Field Extraction from Handwritten Incoming Mail Documents.", *Document Analysis Systems*, 2006, pp 564–575.
- [4] C. De Stefano, C. Sansone and M. Vento, "To Reject or Not to Reject: That is the Question - An Answer in Case of Neural Classifiers", *IEEE Trans. on Systems, Man and Cybernetics*, 30(1):84–94, 2000.
- [5] T. Fawcett, "An introduction to ROC analysis", *Pattern Recognition Letters*, in Press, 2005.
- [6] G. Fumera, F. Roli and G. Giacinto, "Reject option with multiple thresholds", *Pattern Recognition*, 33(12):2099–2101, 2000.
- [7] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman and S. Janet, "UNIPEN project of on-line data exchange and recognizer benchmarks.", *Proc. of the 12th ICPR*, 1994, pp 29–33.
- [8] R. Krishnapuram and J. Keller, "A possibilistic approach to clustering", *IEEE Trans. on Fuzzy Systems*, 1(2):98–110, 1993.
- [9] T. Landgrebe, D. Tax, P. Paclik and R. Duin, "The interaction between classification and reject performance for distance-based reject-option classifiers", *Pattern Recognition Letters*, in Press, 2005.
- [10] C.-L. Liu, H. Sako and H. Fujisawa, "Performance evaluation of pattern classifiers for handwritten character recognition", *IJDAR*, 4(3):191–204, 2002.
- [11] J. Liu and P. Gader, "Neural networks with enhanced outlier rejection ability for on-line handwritten word recognition", *Pattern Recognition*, 35:2061–2071, 2002.
- [12] S. Macé, E. Anquetil, E. Garrivier and B. Bossiss, "A Pen-based Musical Score Editor", *Proceedings of ICMC*, September 2005, pp 415–418, Barcelona, Spain.
- [13] H. Mouchère and E. Anquetil, "A Unified Strategy to Deal with Different Natures of Reject", *ICPR (to be published)*, 2006.
- [14] C. Viard-Gaudin, P. Lallican, S. Knerr and P. Binter, "The IREST on/off dual handwriting database", *Proc. of the 5th ICDAR*, 1999, pp 455–458.