

Détection des fautes dans les réseaux des capteurs sans fil à base de la méthode du Perceptron

Mira Sarkis^{1,2}, Dima Hamdan^{1,2}, Oum-El-KheirAktouf², IoannisParissis²

1. Université Libanaise, Laboratoire LASTRE, Tripoli, Liban

2. Grenoble INP, Laboratoire LCIS, 26000 Valence, France

Contact : gne.mirasarkis@gmail.com , dima.hamdan@lcis.grenoble-inp.fr

Résumé

Les capteurs sans fil par leur sensibilité à l'environnement où ils sont déployés, leur limitation en terme d'énergie, mémoire et puissance de calcul risquent de nombreuses défaillances qui peuvent paralyser le réseau et mettre fin au service de l'application. Les capteurs peuvent communiquer des données inadéquates, erronées et dans certains cas ne plus communiquer. Cet article présente une approche en ligne pour la détection des fautes, basée sur la classification des données en se référant à une méthode des réseaux de neurones formels, le Perceptron. Les capteurs échangent localement leurs lectures et décident ensemble de l'exactitude des données. Les simulations appliquées sur une base de données réelle montrent que 35.6% des données risquent d'être erronées.

Abstract

Sensors with their limitations in terms of energy, memory and processor, are prone to many failures that may threaten each part of the network, including the application as a whole. In this case, sensors can communicate inadequate or faulty data and sometimes fail to communicate anything. In this paper we present an online approach for fault detection based on data classification and artificial neural networks. Sensors exchange locally among themselves their knowledge concerning the environment and decide all together the state of their sensed data. Simulation results applied on real datasets show that up to 35.6% of data may be faulty.

Mots-clés : réseau de capteurs sans fil, fautes de données, détection en ligne, réseau de neurones formel, Perceptron, corrélation spatiale.

Keywords: wireless sensor networks, data faults, online fault detection, formal neural networks, Perceptron, spatial correlation

1. Introduction

Les réseaux de capteurs sans fil (RCSF) consistent en des dispositifs autonomes déployés aléatoirement ou avec déterminisme pour la surveillance et la supervision des phénomènes dans différents domaines de la vie courante tels que la surveillance des personnes âgées, la télésurveillance des malades [6], le contrôle des systèmes industriels, les maisons intelligentes, l'agriculture, etc.

L'évolution de la technologie de la miniaturisation des dispositifs électroniques a poussé à la formation de réseaux denses et hétérogènes. Cela a permis un potentiel de diversification et enrichissement des applications d'un côté. Mais de l'autre côté, les petites dimensions sont contraignantes puisqu'elles posent des limitations au niveau logiciel et matériel : les capteurs sont équipés de batteries non échangeables, de mémoires de petite taille (quelques Kbytes) et d'un processeur simplifié.

Les réseaux de capteurs sont déployés pour qu'ils servent l'application pour une durée s'étendant de quelques jours jusqu'à des années. Cela rend fort probable la confrontation à des défaillances, des fautes et des échecs matériels et logiciels aux différents niveaux du modèle OSI : physique (nœuds), réseaux (communication), application (service).

Dans ce travail, nous nous intéressons aux fautes au niveau des données : les données peuvent être incohérentes, désynchronisées, incomplètes, erronées et même perdues. Pour toutes ces

D'inspiration biologique, les neurones artificiels sont des petites unités de traitement composées d'une ou plusieurs entrées, d'une sortie et d'un corps cellulaire qui effectue des calculs à partir des données de l'entrée pour produire la sortie (figure 2). En pratique, un neurone est capable de discriminer deux classes, c'est-à-dire il effectue une analyse de ce qu'il perçoit du monde extérieur, effectue des calculs linéaires et découpe les entrées en deux zones linéairement séparables.

Le modèle biologique fait intervenir une notion temporelle souvent difficile à intégrer dans le modèle mathématique simple ; de ce fait nous remplaçons l'intégration temporelle par une sommation des signaux arrivant au neurone. La somme est ensuite comparée à un seuil. Le résultat de cette comparaison est la sortie du neurone qui peut être soit binaire (0,1) soit sous forme de $(1,-1)$.

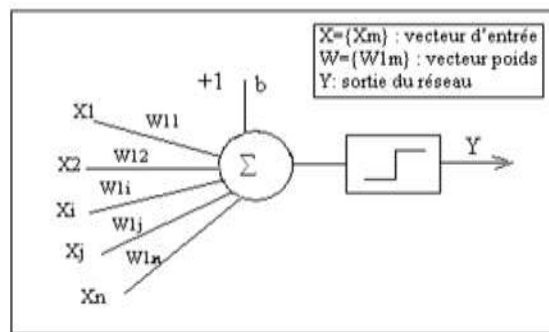


FIG. 2 – Architecture d'un réseau de neurone formel : sommation des entrées pondérée par des poids, suivie par un seuillage.

Toutes les méthodes de réseaux de neurones consistent en deux étapes : l'apprentissage et le test.

La première consiste à apprendre le réseau à partir des entrées d'apprentissage. A la fin de cette première étape, le critère de séparation est déterminé. Alors, la deuxième étape est une exploitation de ces critères pour la classification des données.

Pour cela, nous séparons les données en deux parties : Les données d'apprentissage et les données du test. L'ensemble des données de test est beaucoup plus grand que l'ensemble d'apprentissage. Plus l'espace d'apprentissage est large, plus le modèle obtenu s'approche de la réalité.

Notre approche se base sur une méthode de réseaux de neurones qui s'appelle le Perceptron. Perceptron est une méthode de classification supervisée basée sur un apprentissage supervisé où chaque entrée est étiquetée par la classe à laquelle elle appartient.

La figure 2 illustre le Perceptron dont le vecteur d'entrée de dimension n est connecté au neurone par des connexions caractérisées par des poids W_{ki} pour $k=1 : n$ et par un biais de poids b . La sortie du sommateur est alors $\sum X_k.W_{ki}+b$ qui est comparé théoriquement au 0.

$\sum X_k.W_{ki}+b=0$ exprime un hyperplan de séparation de dimension n . Le but de l'apprentissage est de déterminer cet hyperplan en altérant les poids $\{W_{ki}\}$ ainsi que b pour chaque itération de telle façon que la règle de classification associée donne une probabilité d'erreur minimale.

Algorithme 1 : Apprentissage du Perceptron

Entrées : données X à classifier, les sorties prédites ($Y_{\text{prédit}}$),

Sortie : poids des neurones $\{W_k\}$

Début

Initialiser les poids

tant que $iteration < 10000$ **faire**

si classe (X) \neq décision(X, W) **alors**

 1 $W_{k+1} \rightarrow W_k + T_a \cdot Y_i \cdot X_i$

$K \rightarrow k+1$

 // T_a : taux d'apprentissage

sinon

$$W_{k+1} \rightarrow W_k$$

fin

Le taux d'apprentissage T_a détermine la vitesse de convergence ainsi que la précision de la méthode. Souvent T_a est pris égal à 0.1.

La convergence de l'algorithme a souvent lieu après 10000 itérations.

Algorithme 2 : Décision du Perceptron

Entrées : données X à classifier, l'hyperplan de séparation

Sortie : la classe Y de chaque entrée

```

1  Début
2      si  $W.X+b>0$  alors
3           $Y = +1$ 
4      sinon
5           $Y = -1$ 
6  fin
  
```

2.c. Application du Perceptron pour la détection des fautes de données de température dans le RCSF

Les capteurs sont assemblés à trois en se basant sur leur position géographique : ceux qui sont proches physiquement devraient avoir leurs données corrélées.

Pour un couple $(S1, S2)$ quelconque de chaque ensemble, si nous tracons les données $D2$ de l'un en fonction des données $D1$ de l'autre, nous devons trouver $D2=1*D1$.

Nous avons considéré 13 différentes associations de capteurs. Pour chaque ensemble $(S1, S2, S3)$, nous déterminons les 3 distances d_{ij} avec $i, j=1:3$ et $i \neq j$ existantes entre les températures des trois capteurs, qui seront les entrées du réseau.

Nous avons limité la proportion des jours d'apprentissage à $1/6$ des 32 jours, soient 5 jours dont les données sont étiquetées. Pour les étiqueter, nous calculons la distance euclidienne entre le couple de données (D_i, D_j) et la première bissectrice et nous déterminons un seuil de classification basé sur le mode des distances et la valeur moyenne. Si la distance est supérieure à ce seuil, nous lui associons le tag -1, et sinon le tag sera +1.

La figure 3 montre les droites de séparation trouvées pendant la phase d'apprentissage du couple de capteurs d'ID 1 et 2. Nous remarquons que les droites sont toutes proches à l'exception de celle correspondant au jour 1. En moyennant les pentes de ces droites, nous déterminons la droite de séparation.

Distance	Dist(S1, S2)	Dist(S1, S3)	Dist(S2, S3)
Pente	1.00843	1.00988	1.00756
Taux d'erreur(%)	0.84	0.9	0.75

TAB. 1 – Les pentes des droites de séparation des trois distances d_{12} , d_{13} , d_{23} .

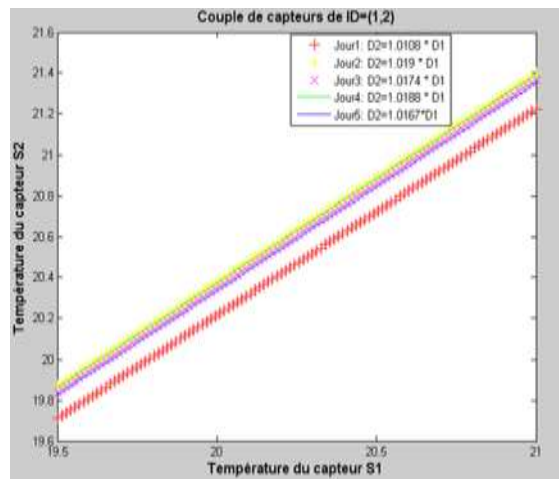


FIG. 3 -Droites de séparation trouvées pendant l'apprentissage du couple de capteurs d'ID 1 et 2.

Le tableau 1 présente les moyennes des pentes des droites de séparation des 3 distances. Les valeurs obtenues valident notre hypothèse puisque les pentes s'approchent de la valeur théorique avec un taux d'erreur de 0.84%, 0.9% et 0.75% respectivement pour les trois distances. Pendant la deuxième phase, nous calculons la distance entre chaque couple de points et la droite de séparation trouvée et puis nous la comparons à un seuil ; 3 seuils sont considérés : 0.5, 0.7 et 1.

Le Perceptron appliqué sur le couple des capteurs d'IDs 1 et 2 nous a permis de valider que le meilleur seuil de classification est 0.7. Pendant le jour numéro2 les données de ces deux capteurs sont fortement corrélées : le coefficient de corrélation est de l'ordre de 0.990226 et tend forcément vers 1.

Le tableau 2 montre qu'une unique faute est détectée avec le seuil 0.7.

Seuils	0.5	0.7	1
Taux de fautes	62/562	1/562	0/562

TAB. 2 – Résultats du Perceptron, avec changement du seuil, appliquée sur les données du jour 2 du couple de capteurs d'ID 1 et 2.

3. Résultats et discussion

Dans le but de détecter les fautes en ligne, le Perceptron est appliqué sur une base de données réelle de façon hors ligne pour déterminer le modèle de la corrélation spatiale entre l'ensemble des capteurs. Nous avons essayé de vérifier ce modèle sur les 27 jours de test. Les résultats montrent que les droites de la figure 2 expriment avec un certain degré de fidélité la relation linéaire existante entre les données des capteurs corrélés spatialement.

L'algorithme est appliqué 2 fois au niveau de chaque nœud puisque le capteur doit comparer ses lectures à celles des voisins. La décision concernant le statut des lectures se fait d'une façon locale au niveau d'un des trois capteurs. Soient les trois distances d12, d13 et d23, plusieurs cas existent:

1. Si d12, d13 et d23 sont jugés par le Perceptron comme des distances correctes alors il n'y a pas d'erreur au niveau des lectures.
2. Si d12 et d13 (respectivement (d12, d23) et (d13, d23)) sont jugés comme des distances erronées alors que d23 (respectivement d13 et d12) ne présente pas de problème, donc il est fortement probable que la lecture du capteur 1 (respectivement 2 et 3) est erronée.
3. Si d12 et d13 (respectivement (d12, d23) et (d13, d23)) sont jugés comme des distances saines alors que d23 (respectivement d13 et d12) est erronée, donc la lecture du capteur 1 (respectivement 2 et 3) est la plus fiable.
4. Si d12, d13 et d23 sont jugées par le Perceptron comme des distances erronées, 2 possibilités ont lieu : soit toutes les lectures sont erronées et il y a un problème au niveau des trois capteurs, soit un événement extérieur a lieu et dans ce deuxième cas nous devons consulter les ensembles voisins pour décider.

Les figures 4 présentent les taux d'erreurs obtenus en appliquant le Perceptron sur les 3 distances d12, d13 et d23 des 13 ensembles et cela pour différentes valeurs du seuil. Nous pouvons remarquer que l'ensemble numéro 3 formé par les capteurs d'ID 8, 9 et 10 est le plus homogène, le plus corrélé et le plus fiable parmi les autres ensembles. Les figures 5, 6 et 7 présentent la classification des données de cet ensemble pour un jour aléatoire. Le tableau 3 présente les moyennes des taux d'erreur trouvées pour différentes valeurs de seuils, Nous remarquons que pour le seuil 0.7

Seuil	0.5	0.7	1
Taux des fautes(%)	41.9	35.6	30.14

TAB. 3 – taux de fautes obtenus en appliquant Perceptron avec différents seuils

le taux d'erreur est de 35.6%. Cette valeur reflète le pourcentage des données qui risquent d'être erronées sans préciser le capteur défectueux, ni la cause de ces défaillances. [2] traite la corrélation temporelle au niveau des différents types (température, humidité et lumière) de capteurs en se basant sur 5 règles différentes et trouve que 19% des données sont erronées. La réalité est que notre approche ne prend pas en considération le dépassement des valeurs limites, ni les événements extérieurs qui peuvent avoir lieu, mais uniquement la similitude entre les résultats des capteurs. En plus, les paramètres de seuillage sont en relation avec les 5 jours d'apprentissage et en fait, il est avantageux d'élargir le nombre de ces jours à condition que le nombre total des jours soit grand pour mieux calibrer les valeurs critiques.

4. Conclusion et perspective

Dans ce papier, une méthode de détection des fautes a été proposée en exploitant les types de données envoyées par chaque nœud dans un RCSF : la température, le temps et la position géographique du dispositif et en prenant en considération les limitations du réseau des capteurs. Perceptron consiste tout simplement en un calcul de distance suivi d'une comparaison, ces fonctions ne nécessitent pas beaucoup d'espace mémoire et consomment peu d'énergie. Mais le Perceptron présente l'inconvénient d'être une méthode supervisée et nécessite l'étiquetage des données avant tout apprentissage. De même, il peut conduire à des maldétection et à des fausses alarmes. Notre futur travail est d'analyser une autre méthode des réseaux de neurones non supervisée, Self Organizing Map ou SOM, et de l'appliquer sur la même base de données pour enfin comparer l'efficacité des deux techniques.

Bibliographie

1. P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, R. Thibaux. <http://db.csail.mit.edu/labdata/labdata.html>. Intel Lab Data, 2004.
2. D. Hamdan, O. Aktouf, I. Parissis, B. Hassan, A. Hijazi. *Online fault diagnosis for wireless sensor networks-case study*. The Third International Conference on Wireless Communications in Unusual and Confined Areas, Clermont-Ferrand, France, 2012.
3. A. Moustafa, R. Selmic. *Wireless Sensor Network Modeling using Modified Recurrent Neural Network: Application to Fault Detection*, in Proc. IEEE Transactions and Measurement, Vol. 57, No. 5, mai 2008.
4. S. Siripanadorn, W.Hattagam, N. Teaumroog. *Anomaly Detection for Wireless Sensor Networks using Self Organizing Map and Wavelets*, International Journal of Communication, 2010.
5. Y. Yao, A.Sharma, L. Golubchik, R. Govinda. *Online Anomaly Detection for Sensor Systems: A Simple and Efficient Approach*, ScienceDirect-Performance Evaluation, 2010.
6. J. Yick, B. Mukherjee, D.Ghosal. *Wireless Sensor Network Survey*, Science Direct, 2008.
7. R. Zhur. *Efficient Fault-Tolerant Event Query Algorithm in Distributed Wireless Sensor Networks*, International Journal of Distributed Sensor Networks, volume 2010.

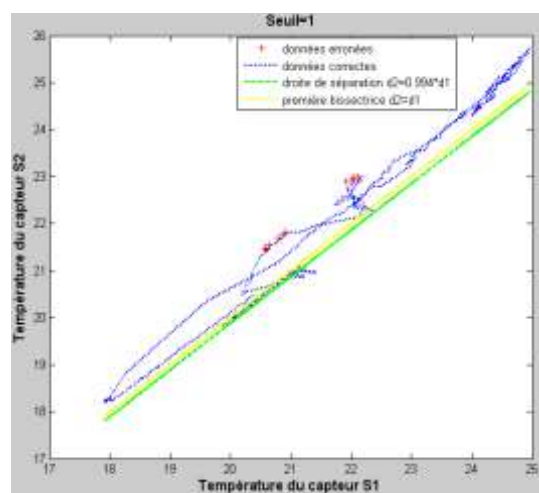
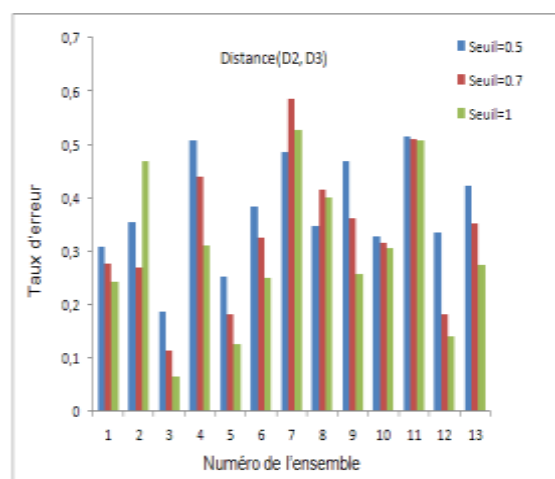
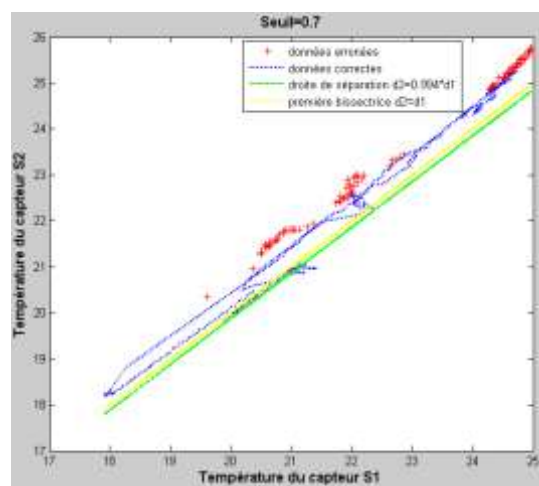
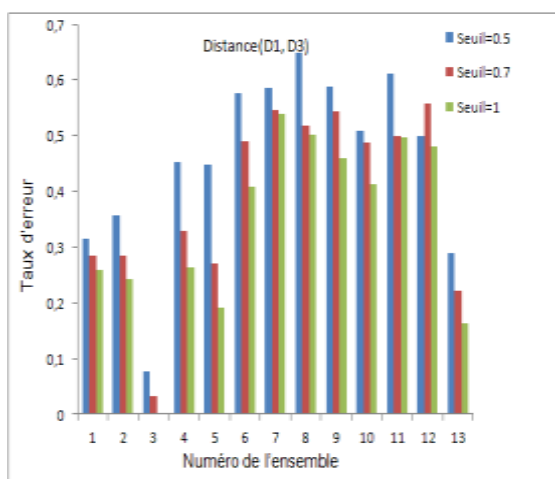
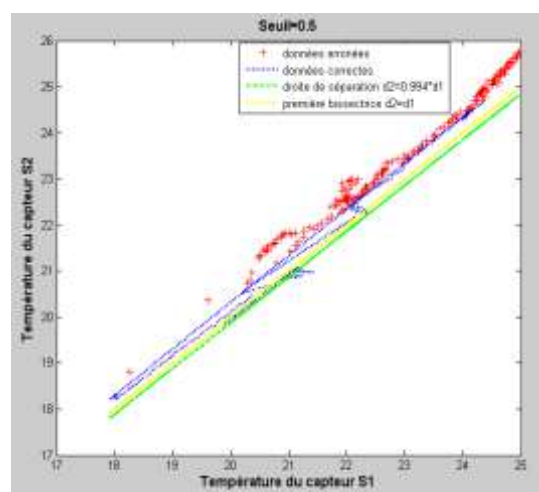
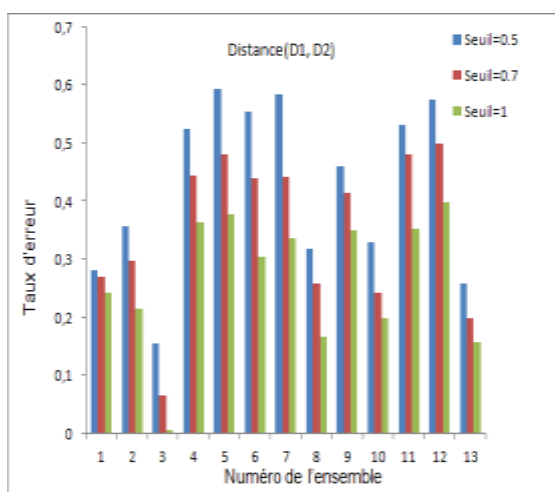


FIG. 4 –Histogramme des taux d’erreurs obtenus en appliquant Perceptron sur les triplets de distances des 13 ensembles et pour les 3 valeurs de seuils

FIG. 5 –Classification des températures du couple de capteurs d’ID 8 et 9 pour les 3 seuils

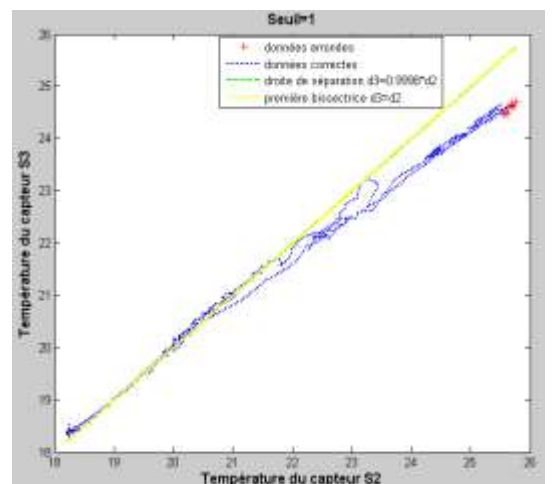
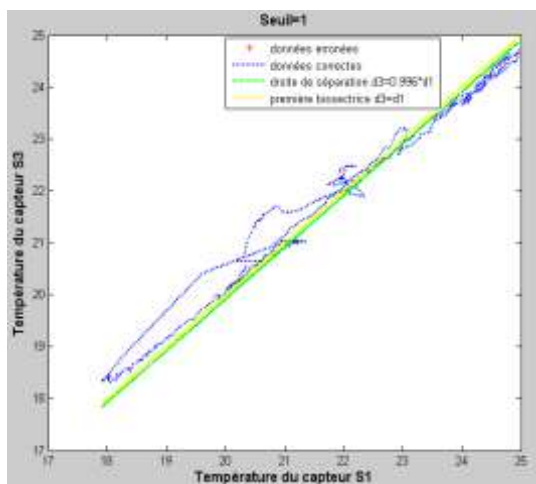
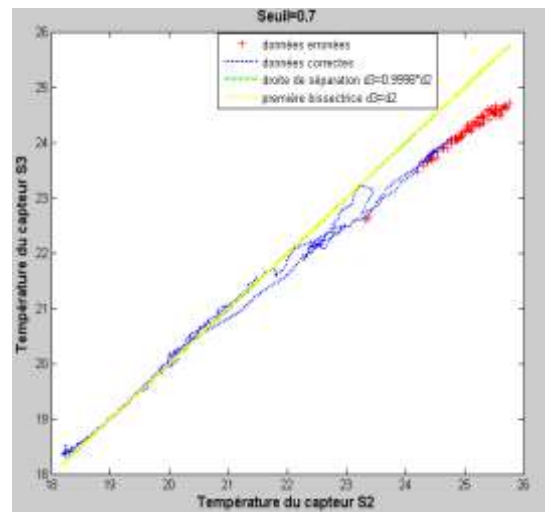
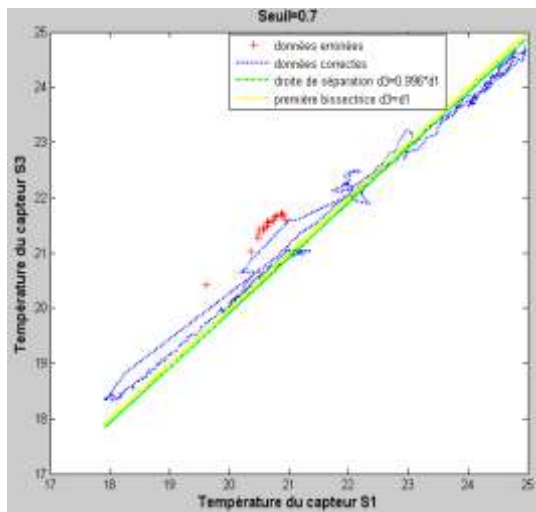
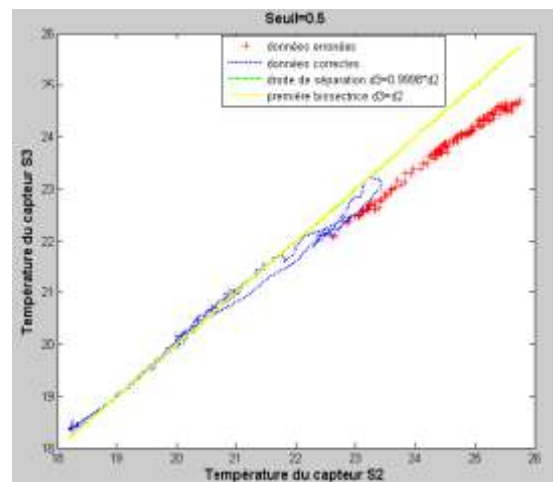
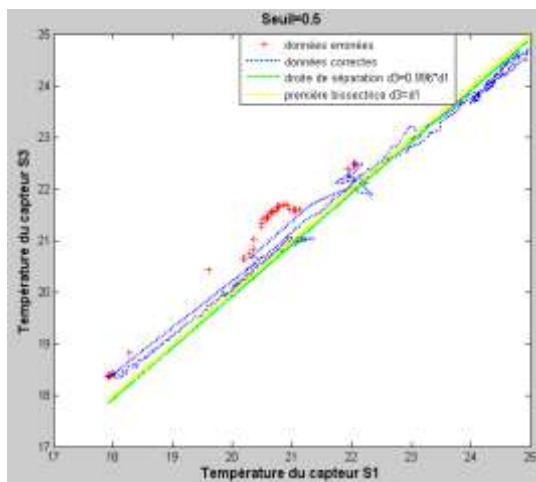


FIG. 6 -Classification des températures du couple de capteurs d'ID 8 et 10 pour les 3 valeurs de seuils

FIG. 7 -Classification des températures du couple de capteurs d'ID 9 et 10 pour les 3 valeurs de seuils