Un ensemble classificateur pour la classification de données dynamiques. Application à un problème de qualité d'air intérieur

Philippe Thomas, William Derigent, Marie-Christine Suhner

UN ENSEMBLE CLASSIFICATEUR POUR LA CLASSIFICATION DE DONNEES DYNAMIQUES. APPLICATION A UN PROBLEME DE QUALITE D'AIR INTERIEUR

[1,2]Philippe THOMAS, [1,2]William DERIGENT, [1,2]Marie-Christine SUHNER

[1]Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, 54506 Vandœuvre-lès-Nancy cedex, France
[2]CNRS, CRAN, UMR7039, France
Philippe.thomas@univ-lorraine.fr, William.derigent@univ-lorraine.fr, marie-christine.suhner@univ-lorraine.fr

RESUME : *La qualité de l'air intérieur a un impact déterminant sur l'exposition des personnes à des polluants dans le monde moderne du fait que les populations passent beaucoup de leur temps dans différents environnements intérieurs. La société Anaximen développe un objet connecté, appelé Alima, qui mesure toutes les minutes plusieurs paramètres relatifs à la qualité de l'air : température, humidité, concentrations de COV, CO_2, formaldéhyde et particules fines (pm). Au-delà de cet aspect de collecte de mesures, Alima inclut des outils d'analyse de données ayant pour but de déterminer les situations d'usages de l'habitation (présence, cuisine, ménage…) afin d'être capable de fournir des conseils à l'utilisateur dans le but d'améliorer cette qualité de l'air. Ce problème est un problème de classification de données dynamiques, et dans cet article, deux outils (réseaux de neurones et arbres de décision) sont testés et comparés pour réaliser cette tâche. Afin d'améliorer les performances du classificateur, les ensembles classificateurs sont également étudiés.*

MOTS-CLES : *Qualité de l'air intérieur, réseaux de neurones, arbres de décision, classification, ensemble classificateur.*

1 INTRODUCTION

La pollution de l'air est maintenant identifiée comme un problème majeur. Cependant, si dans l'opinion publique, ce problème est principalement en relation avec la qualité de l'air extérieur, dans la réalité, c'est bien la qualité de l'air intérieur qui possède le plus grand impact. En effet, la qualité de l'air intérieur a un impact déterminant sur l'exposition des populations à différents polluants dans le monde moderne du fait que ces populations passent la majorité de leurs temps dans différents environnements intérieurs (Walsh *et al*. 1987).

Les vingt dernières années ont vu un accroissement de l'intérêt de la communauté scientifique concernant les effets de la qualité de l'air intérieur sur la santé. L'évolution du design et des méthodes et matériaux de construction de bâtiments dans le but d'améliorer l'efficacité énergétique de ces mêmes bâtiments ont rendu leur espace intérieur plus confiné que dans les bâtiments de construction plus ancienne. De plus, l'évolution des matériaux de construction et d'isolation a induit un accroissement de l'utilisation de matériaux synthétiques qui conduisent à une pollution de l'air intérieur (Jones 1999).

Les impacts avérés sur la santé de différents polluants sont nombreux. Le Tableau 1 est une extraction de (Spengler et Sexton, 1983) permettant d'illustrer certains des polluants de l'air intérieur majeurs. Les sources de pollutions peuvent être localisées à l'intérieur des bâtiments (matériaux de construction, mobiliers, cuisinières…) ou à l'extérieur (air provenant d'une fenêtre ouverte ou transitant à travers le système de ventilation).

Polluant	**Principales sources d'émission**
Allergènes	poussière, animaux domestiques…
Amiante	Matériaux retardant d'incendie, isolation…
Dioxyde de carbone	Activité métabolique, combustion…
Monoxyde de carbone	chaudières, cuisinières, gaz…
Formaldéhyde	Panneaux de particules, isolation, ameublement…
Micro-organismes	Population, animaux, plantes, air conditionné…
Dioxyde d'azote	Air extérieur, chaudière, moteurs thermiques…
Substances organiques	Adhésifs, solvants, matériaux de construction…
Ozone	Réactions photochimiques
Particules	Tabac, produits de combustion...
hydrocarbures aromatiques polycycliques	Combustion, Tabac…
Pollens	Air extérieur, arbres, herbe, plantes…
Radon	Sol, matériaux de construction (béton, pierre)
Spores	Sol, plantes, denrées alimentaires, surfaces intérieures…
dioxyde de soufre	Air extérieur, combustion…

Tableau 1 : listes de polluants et sources de pollution

Les symptômes et conséquences liés à l'exposition aux polluants varient en fonction du type et de la concentration du polluant considéré. Par exemple, le dioxyde de carbone (dont la concentration intérieure peut varier de 700 à 3000 ppm), est un gaz asphyxiant et peut conduire

à une simple irritation des voies respiratoires (Maroni *et al*. 1995), quand une exposition à un formaldéhyde avec une concentration de 100 ppm peut conduire à la mort.
Ceci explique l'attention nouvelle du public au problème de la qualité de l'air intérieur, et sa volonté de pouvoir la mesurer dans leur propre logement. Pour répondre à ce besoin naissant, la société Anaximen développe un objet connecté appelé Alima (Alima, 2003).
Alima est capable de mesuré toutes les minutes plusieurs paramètres physiques : température, humidité, concentration de COV, CO_2, formaldéhyde et particules fines (pm). Les données sont stockées dans l'objet et peuvent être collectées par une base de données distante. Elles sont consultables par l'utilisateur via une application mobile ou un site web. Au-delà des seuls aspects de mesure de la qualité de l'air, la société Anaximen projette d'embarquer dans Alima des capacités d'analyse de données. L'objectif considéré ici est de déterminer, connaissant l'évolution des différents paramètres collectés (température, humidité, CO_2, COV, pm) l'état d'usage du logement considéré (présence, ouverture de fenêtre, cuisine, ménage…) de manière à ce qu'Alima soit en capacité de conseiller l'utilisateur quant à ses actions. Le problème considéré est donc un problème de classification de données dynamiques. Anaximen et le CRAN sont associés pour développer un outil permettant à Alima d'acquérir cette capacité.
Dans cet article, deux outils sont plus particulièrement testés, les arbres de décision et les réseaux de neurones. Afin d'améliorer les performances du classificateur retenu, les ensembles classificateurs sont également étudiés.
La section 2 présente un court état de l'art concernant les problèmes de classification. La section 3 présente les trois outils testés et comparés (réseaux de neurones, arbres de décision, ensemble classificateur) quand la section 4 présente l'application industrielle avant de conclure.

2 BREVE REVUE DES PROBLEMES DE CLASSIFICATION

Il existe deux grandes classes de problèmes de classification : la classification supervisée et la classification non-supervisée. Pour les problèmes de classification non-supervisée (clustering) l'objectif est de grouper les différents patterns en un ensemble de clusters significatifs. Une review de ces problèmes a été proposée par Jain et al. (1999).
Pour la classification supervisée, une collection de patterns auxquels sont associées des étiquettes (pré classification) existe. L'objectif est d'être capable d'étiqueter un nouveau pattern inconnu. Les patterns étiquetés sont utilisés pour apprendre la description des classes qui sera utilisées pour classer les nouveaux patterns inconnus.
Ce papier se focalise sur les problèmes supervisés qui correspondent à un processus de fouille de données (Knowledge Discovery in Data KDD). Un processus KDD se décompose en plusieurs étapes (Patel et Panchal, 2012) :

- Sélection : collecte des données à partir de diverses sources (a),
- Preprocessing : nettoyage des données (b),
- Transformation : conversion dans un format commun (c),
- Data mining : recherche du résultat (d),
- Interprétation/Evaluation/Présentation : présentation des résultats sous une forme compréhensible (e).

Les deux principales étapes sont la sélection (a) et le data mining (b). Pour la phase de collecte de de données (a), une connaissance experte permet de déterminer quels champs (attributs, caractéristiques) sont les plus informatifs. Si une telle connaissance experte n'est pas disponible, alors la « force brute » doit être utilisée, c'est-à-dire mesurer tout ce qui est disponible et espérer que les caractéristiques et attributs les plus informatifs pourront être isolés (Kotsiantis 2007). Cette approche nécessite d'exploiter des outils capables de déterminer si une caractéristique est utile ou non. Dans de nombreux cas, une collecte manuelle au moins partielle des données est incontournable ce qui est alors une source importante d'erreurs et de valeurs aberrantes. Un pre-processing (b) des données est souvent nécessaire, afin par exemple, de synchroniser les différentes bases de données, supprimer les valeurs aberrantes évidentes, ou encore digitaliser les données qualitatives comme la couleur par exemple (c).
Le data mining (d) qui est le cœur du processus de KDD consiste à analyser les données, et à en extraire l'information utile. Le choix de l'outil d'apprentissage est une phase critique du processus. Différentes approches peuvent être utilisées. On peut citer, l'intelligence artificielle, les machines d'apprentissage ou les algorithmes statistiques.
Les « support vector machines » (SVM) et les réseaux de neurones (NN) utilisent des concepts très proches et donnent des résultats souvent comparables. Certaines fois, les SVM donnent de meilleures résultats (Meyer et al., 2003), d'autre fois, ce sont les NN (Paliwal et Kumar, 2009; Hajek et Olej, 2010). Nous testerons ici des NN.
Les NN effectuent une recherche locale de minimum. Ce fait induit que différents jeux de paramètres initiaux doivent être utilisés pour l'apprentissage afin d'éviter le risque de tomber dans un minimum local très éloigné du minimum global recherché. Ces différents apprentissages sur des jeux de poids initiaux différent conduisent à des classificateurs différents plus ou moins performants.
Les arbres de décision sont une autre approche pour extraire un classificateur d'un ensemble de données. Ils permettent d'obtenir un ensemble de règles présentant une structure hiérarchique et séquentielle pour partitionner les données (Murthy 1998). Une telle approche permet d'obtenir un classificateur plus compréhensible comparativement aux boites noires obtenues avec des NN ou des SVM (Kotsiantis 2013). Cette approche sera également testée et comparée avec l'approche NN.
Quel que soit l'outil utilisé, le choix du meilleur classificateur peut être effectué selon deux approches :

- Sélectionner le classificateur unique qui donne les meilleurs résultats parmi tous les classificateurs appris,
- Construire un ensemble de classificateurs et exploiter une stratégie de vote pour construire la décision.

Un ensemble de classificateur inclus différents classificateurs individuels. La sélection des classificateurs individuels peut être effectuée en exploitant les performances de chacun d'entre eux, ou en utilisant la diversité entre les différents classificateurs. Une bonne diversité entre les différents classificateurs permet d'améliorer les performances des ensembles classificateurs. Les quatre algorithmes les plus populaires permettant d'améliorer la diversité sont bagging (Breiman 1996), boosting (Freund et Schapire 1996), rotation forest (Rodriguez et al. 2006) et random subspace method (Ho 1998). Bagging et random subspace methods sont les deux approches les plus robustes lorsque les données sont bruitées (Dietterich 2000, Kotsiantis 2011). Nous utiliserons ici une approche bagging.

3 OUTILS POUR LA CLASSIFICATION

Comme expliqué précédemment, un grand nombre d'outils peuvent être utilisés pour résoudre un problème de classification. Nous allons nous focaliser sur deux d'entre eux appartenant à deux familles différentes et présentant des caractéristiques différentes et complémentaires : les arbres de décisions qui permettent d'obtenir des modèles de connaissance mais dont l'adaptabilité est limitée en présence d'un environnement évolutif, et les réseaux de neurones qui présentent de bonnes capacité d'adaptation par réapprentissage en présence d'environnement évolutif mais qui fournissent des modèles de type boite noire.

3.1 Arbres de décisions

Les arbres de décision sont des modèles séquentiels combinant une séquence de tests simples. Chaque nœud dans un arbre de décision représente un attribut d'une instance à classer et chaque branche sortant du nœud représente une valeur pour l'attribut testé dans le nœud.
Les instances sont classées à partir du nœud racine et triées en fonction de leurs valeurs d'attributs.
De nombreux algorithmes d'arbres de décision ont été développés comme C4.5 (Quinlan 1996), CART (Breiman et al. 1984) SPRINT (Shafer et al. 1996), SLIQ (Mehta et al. 1996). Nous utiliserons ici l'algorithme « Classification and Regression Tree » (CART) proposé par Breiman et al. (1984).
La première difficulté lors de la construction d'un arbre de décision consiste à exploiter le jeu de données afin de scinder l'espace d'entrée récursivement en zones de plus en plus petites de telles sorties que les jeux de données, dans chaque sous-espace soient de plus en plus pures que dans les niveaux supérieurs. A chaque niveau, l'algorithme recherche pour chaque attribut la meilleure subdivision qui accroit la pureté des descendants.
Le choix du critère de division est dérivé d'une fonction d'impureté, la plus classique étant la fonction d'impureté de Gini. Cependant, le choix de la fonction d'impureté n'a que peu d'impact sur le résultat final de l'arbre de décision (Breiman et al. 1984).
Le processus de croissance de l'arbre est récursif et la croissance des branches s'arrête lorsque (Lewis 2000) :

- Il n'y a qu'une seule observation dans le nœud terminal,
- Toutes les observations dans le nœud terminal possèdent une distribution identique rendant toute division impossible,
- Une limite externe sur le nombre de niveau de l'arbre ou sur un seuil de décroissance de l'impureté est atteinte.

Une fois l'arbre créé, une règle d'assignement de classe à chaque nœud doit être utilisée. La classe prédite associée à chaque nœud est celle correspondant à la classe la plus représentée parmi tous les cas associés au nœud considéré. Un coût de mauvaise classification peut également être utilisé (Breiman et al. 1984).
L'arbre ainsi obtenu présente généralement un problème de surapprentissage qui nécessite une phase de pruning. Cette phase de pruning est basée sur l'estimation du risque pour chaque nœud associé à un terme de pénalité correspondant au nombre de nœuds terminaux. La sélection de l'arbre de décision final est obtenue à l'aide d'un processus de validation croisée (Breiman et al. 1984).

3.2 Le perceptron multicouches (MLP)

Les réseaux de neurones artificiels ont été appliqués pour résoudre différents types de problèmes comme notamment, l'identification de systèmes dynamiques, la classification, le contrôle adaptatif, l'approximation de fonctions… Parmi tous les modèles neuronaux, le perceptron multicouches (MLP) est l'architecture la plus populaire du fait de sa structure flexible, de ses bonnes capacité de représentation et de généralisation, et du grand nombre d'algorithmes d'apprentissage (Han et Qiao 2013).
Les travaux de Cybenko (1989) et Funahashi (1989) ont prouvé qu'un MLP utilisant une seule couche cachée exploitant une fonction d'activation sigmoïdale peut approcher toute fonction non-linéaire avec la précision désirée. Sa structure est donnée par :

$$z = g_2\left(\sum_{i=1}^{n_1} w_i^2 . g_1\left(\sum_{h=1}^{n_0} w_{ih}^1 . x_h^0 + b_i^1\right) + b\right). \qquad (1)$$

où x_h^0 sont les n_0 inputs du réseau, w_{ih}^1 sont les poids connectant la couche d'entrée à la couche cachée, b_i^1 sont les biais des neurones de la couche cachée, $g_1(.)$ est la fonction d'activation des neurones cachés (ici la tangente hyperbolique) w_i^2 sont les poids connectant les neurones cachés au neurone de sortie, b est le biais du neurone de sortie et $g_2(.)$ est la fonction d'activation de neurone de sortie et z est la sortie du réseau. Le problème

considéré ici étant un problème de classification, $g_2(.)$ est choisie sigmoïdale.
La construction d'un modèle neuronal s'effectue en trois étapes : initialisation, apprentissage, et pruning.
L'initialisation consiste à déterminer le jeu de poids et biais initiaux. Cette étape est importante car l'apprentissage effectue une recherche locale de minimum. Aussi, pour éviter de tomber dans un minimum local très éloigné du minimum global, différents jeux de paramètres initiaux doivent être construits afin de permettre à l'apprentissage de commencer dans différentes zones du domaine du critère. Différents algorithmes d'initialisation ont été proposés (Thomas et Bloch 1997). L'algorithme utilisé ici est celui proposé par Nguyen et Widrow (1990) qui permet d'associer une initialisation aléatoire des paramètres à un placement optimal dans l'espace des entrées. Cette étape permet d'accroitre la diversité des classificateurs neuronaux ce qui permet de se passer, pour les réseaux de neurones de l'étape de bagging.
La seconde étape correspond à l'apprentissage proprement dit dont l'objectif consiste à faire correspondre la sortie du réseau avec les données. Dans les applications industrielles, les données sont bruitées et polluées par de nombreuses valeurs aberrantes. Dans le but de limiter l'impact des valeurs aberrantes sur le résultat, un algorithme de Levenberg-Marquard robuste est utilisé (Thomas et al. 1999). L'algorithme de Levenberg–Marquard algorithm permet d'associer la rapidité des méthodes du Hessien à la stabilité des méthodes du gradient. L'utilisation d'un critère robuste permet de limiter l'impact des valeurs aberrantes sur le résultat et fournit de surcroit un effet de régularisation permettant de prévenir le problème de sur apprentissage.
Un point important lors de la construction d'un modèle neuronal est la détermination de la structure du réseau. Pour ce faire, deux approches peuvent être utilisées. La première, constructive, consiste à ajouter des neurones cachés l'un après l'autre (Ma et Khorasani 2004). La seconde, le pruning, exploite une structure surdimensionnée et élimine les paramètres les moins significatifs dans un deuxième temps (Setiono et Leow 2000, Engelbrecht 2001). Nous utiliserons une approche de pruning qui présente l'avantage de sélectionner simultanément le nombre de neurones cachés et les neurones d'entrée. La phase de pruning est subdivisée en deux étapes. Dans un premier temps, l'algorithme proposé par Engelbrecht est utilisé ce qui permet de rapidement simplifié la structure, et dans un deuxième temps, l'algorithme proposé par Setiono et Leow plus lent, mais plus efficace est utilisé (Thomas et al. 2013b). Cette étape est également importante pour améliorer la diversité entre les différents classificateurs en vue de construire l'ensemble classificateur.

3.3 Ensemble classificateurs

Egalement connus sous les termes de « committees of learners », « mixture of experts », « multiple classifier systems », les ensembles classificateur ont pour but de construire une collection de classificateurs individuels suffisamment divers et précis afin de construire une décision de classification plus précise par l'intermédiaire d'un vote des différents classificateurs individuels de l'ensemble (Dietterich 2000). Typiquement, un ensemble classificateur peut être construit à quatre niveaux différents (Kuncheva 2004): au niveau des données (Breiman 1996), au niveau des attributs (Ho, 1998), au niveau des classificateurs et au niveau de la combinaison (Kuncheva 2002). Le principe des ensembles classificateurs est présenté figure 1. La construction d'un ensemble classificateur consiste en deux étapes principales : la génération de plusieurs classificateurs et leur fusion (Dai 2013). Ceci conduit à répondre à deux problèmes principaux :

- Combien de classificateurs sont nécessaires ?
- Comment effectuer la fusion ?

La sélection de classificateur est un problème étudié par de nombreux auteurs (Ruta et Gabris 2005, Hernandez-Lobato 2013, Dai 2013). Deux approches peuvent être utilisées pour effectuer cette sélection (Ruta et Gabrys 2005) :

- Sélection des classificateurs statique. La sélection optimale trouvée pour le jeu de validation est figée et utilisée pour la classification des patterns inconnus.
- Sélection des classificateurs dynamiques. La sélection s'effectue en-ligne, durant la classification en exploitant les performances de l'apprentissage et également un ensemble de paramètres du pattern inconnu à classer.

Afin de préserver la rapidité de classification, une approche statique est ici privilégiée.

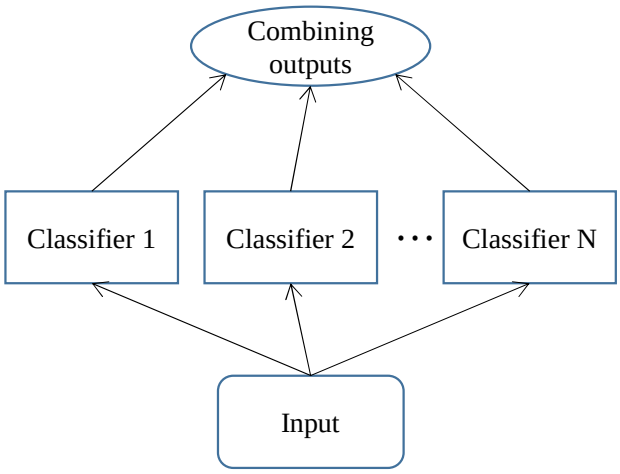

Figure 1 : Un ensemble classificateur

Différents critères de sélection ont été proposés. La performance individuelle des classificateurs est un critère universel pour la sélection du meilleur classificateur à inclure à l'ensemble. C'est l'approche la plus simple qui est également fiable et robuste. Cette approche est généralement préférée dans le cas d'applications industrielles (Ruta et Gabrys 2005). Cet indicateur appelé erreur minimale individuelle (MIE), représente le taux erreur minimal du classificateur individuel et favorise une stratégie de sélection des meilleurs classificateur individuels :

$$MIE = \min_j \left(\frac{1}{n} \sum_{i=1}^{n} e_j(i) \right). \quad (2)$$

où $e_j(i)$ représente l'erreur de classification du classificateur j pour la donnée i.
La fusion des classificateurs est généralement effectuée à l'aide d'un vote majoritaire. Cette approche de pouvoir également fournir un intervalle de confiance sur le résultat de classification.
Un point important lors de la construction d'ensemble classificateur consiste dans la mesure de la diversité entre les classificateurs individuels. Cependant, il n'existe pas de définition accepté universellement de cette diversité, et la mesurer explicitement reste difficile. Aussi, différentes mesures de la diversité et plusieurs auteurs ont testé et comparé ces mesures sur différents exemples (Kuncheva et Whitaker 2003, Tang et al. 2006, Aksela et Laaksonen 2006, Bi 2012). Kuncheva et Whitaker (2003) ont montré que l'utilisation de chacune de ces mesures conduist à des résultats sensiblement équivalents. Aussi, la mesure double faute est ici utilisée. Elle a été proposée par Giacinto et Roli (2001) et permet de construire une matrice de diversité appariée pour un jeu de classificateurs et est utilisée pour sélectionner le classificateur le plus divers à adjoindre à l'ensemble classificateur
Cette mesure est définit par la proportion des cas qui ont été mal classé par les deux classificateurs considérés :

$$DF_{i,j} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}}. \quad (2)$$

où i et j représentent les deux classificateurs considérés et N^{ab} correspond au nombre de cas de la base de données pour lesquels la sortie du classificateur i est correcte si a=1 (resp. fausse si a = 0) et celle du classificateur j est juste si b=1 (resp. fausse si b = 0) simultanément.
Différente stratégie ont été proposées pour sélectionner les classificateurs à inclure dans l'ensemble (Kim et Oh 2008, Ko et al. 2008, Tsoumakas et al. 2009, Yang 2011, Guo et Boukir 2013, Soto et al 2013).
La stratégie utilisée ici est une évolution de l'algorithme « Selection by Accuracy and Diversity (SAD) » proposé par Yang (2011). Cet algorithme est simple et récursif dont les différentes étapes sont :

1. Construction d'un jeu de différents classificateurs ;
2. Evaluation de la performance de chaque classificateur sur un jeu de validation ;
3. Sélectionner les trois meilleurs classificateurs afin d'être capable de réaliser un vote ;
4. Calculer la diversité entre l'ensemble classificateur obtenu à l'étape 3 et tous les autres classificateurs en utilisant la mesure double faute ;
5. Sélectionner les deux classificateurs présentant la plus grande diversité d'avec l'ensemble et les adjoindre à l'ensemble afin d'obtenir un nouvel ensemble possédant toujours un nombre impair de classificateur permettant de faire un vote ;
6. Evaluer les performances du nouvel ensemble. Si tous les classificateurs ont été introduit dans l'ensemble comparer tous les ensembles classificateurs construit et sélectionner le meilleur, sinon, répéter les étapes 4 à 6.

Cet algorithme est utilisé pour déterminer la structure de l'ensemble classificateur en exploitant une mesure de diversité. Une deuxième approche est également testée afin d'évaluer l'impact de la diversité sur les performances de l'ensemble classificateur. Dans cette seconde approche, les étapes 4 et 5 sont remplacées par :

- Sélectionner les deux classificateurs non inclus dans l'ensemble présentant les meilleures performances individuelles et les adjoindre à l'ensemble.

Le reste de l'algorithme reste inchangé.

4 APPLICATION INDUSTRIELLE

4.1 Description du cas d'étude

Le site d'expérimentation est une maison particulière de plein pied dont le plan est présenté figure 2.
Les ronds rouges indiquent la localisation des 5 Alimas installées dans la maison. L'expérimentation a été menée sur une période de 1 mois durant lequel les différentes alimas ont collecté les valeurs des différents paramètres toutes les minutes.

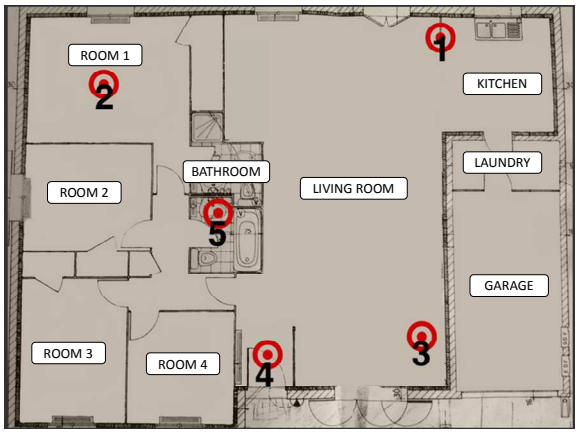

Figure 2: Implantation des 5 Alima dans la maison.

4.2 Présentation des résultats

4.2.1 Organisation des données

L'objectif est ici de pouvoir détecter divers types de comportement de l'utilisateur (cuisine, présence dans une pièce, ouverture de fenêtre…) en fonction de l'évolution des taux des divers polluants mesurés. Nous nous focalisons ici sur le problème de détecter la présence d'occupants dans la chambre. Le problème considéré est un problème de classification de comportement dynamique dans lequel nous ne nous intéressons pas uniquement à l'état présent mais également à la manière

par laquelle l'état présent a été atteint. Il est par exemple évident qu'un même niveau de taux de CO_2 n'aura pas la même signification, si ce taux fait suite à un accroissement ou à une décroissance. Le module Alima collecte les données de cinq polluants (humidité, température, CO_2, pm, COV) toutes les minutes. Pour la construction des modèles, nous n'utilisons que l'Alima1.
Nous utilisons les 5 dernières valeurs de chaque polluant afin de déterminer la situation d'usage. Pour ce faire, deux solutions s'offre à nous. Utiliser directement les valeurs brutes à t, t_{-1}, t_{-2}, …t_{-5} ou utiliser la différence entre la valeur actuelle des polluants et leurs valeurs antérieures t, t-t_{-1}, t-t_{-2}, … t-t_{-5}. Nous créons donc deux jeux de données d'entrée que nous nommerons « brut » pour l'utilisation des valeurs brutes et « diff » pour l'utilisation des différentiels de valeurs. Par la suite, nous comparerons les résultats obtenus sur ces deux jeux de données. En vue de la validation des modèles ces jeux de données seront subdivisés en deux, une partie pour l'identification du modèle, une autre pour sa validation.

4.2.2 Performances des classificateurs individuels

Dans un premier temps, nous allons comparer les performances des deux outils de classification précédemment décrits (perceptron multicouches « NN » et arbre de décisions « Tree ») sur les deux jeux de données précédemment construits. Pour chacun des outils 100 modèles différents ont été construits. Les tableaux 2 et 3 présentent respectivement les résultats obtenus avec le meilleur modèle pour les jeux de données de validation « brut » et « diff ».

	taux mauvaise classif	taux fausse alarme	taux non détection
meilleur NN	4.3%	4.0%	5.4%
meilleur Tree	11.7%	12.1%	10.5%

Tableau 2 : Résultats des meilleurs modèles – « brut »

	taux mauvaise classif	taux fausse alarme	taux non détection
meilleur NN	4.5%	4.3%	4.8%
meilleur Tree	2.1%	1.6%	3.8%

Tableau 3 : Résultats des meilleurs modèles – « diff »

Ces résultats montrent que si le NN fournit des résultats équivalent pour les deux jeux de données ce qui indique qu'un perceptron multicouches ne nécessite pas de traitement particulier des données avant usage, il n'en est pas de même pour les arbres de décisions qui donnent des résultats grandement améliorés avec ce prétraitement (données « diff »). Il est à noter que dans ce dernier cas, les arbres de décision donnent des résultats légèrement meilleurs que les perceptrons multicouches.

4.2.3 Performances des ensembles classificateurs

Nous avons dans l'étape précédente construit 100 modèles de chaque type (NN et Tree) dans le but de construire un ensemble classificateur. Les tableaux 4 et 5 présentent respectivement les résultats obtenus pour les différents types d'ensemble classificateur pour les jeux de données de validation « brut » et « diff ».

	sélection	taille	taux mauvaise classif	taux fausse alarme	taux non détection
NN ensemble	précision	29	2.9%	1.9%	6.2%
Tree ensemble		19	7.3%	4.4%	16.3%
classifier ensemble		189	2.5%	1.2%	6.4%
NN ensemble	diversité	5	2.8%	1.5%	6.9%
Tree ensemble		11	5.0%	3.5%	9.9%
classifier ensemble		41	1.6%	0.7%	4.5%

Tableau 4 : Résultats des meilleurs ensembles – « brut »

	sélection	taille	taux mauvaise classif	taux fausse alarme	taux non détection
NN ensemble	précision	15	3.1%	2.1%	6.3%
Tree ensemble		83	1.1%	0.5%	2.9%
classifier ensemble		59	1.0%	0.5%	2.8%
NN ensemble	diversité	13	2.7%	1.3%	7.3%
Tree ensemble		9	0.9%	0.4%	2.5%
classifier ensemble		35	1.0%	0.3%	2.9%

Tableau 5 : Résultats des meilleurs ensembles – « diff »

Pour chacun des jeux de données, les deux stratégies de construction d'ensemble (sélection sur la précision et sélection sur la diversité) sont testées. Trois types d'ensembles sont construits utilisant respectivement, uniquement des « NN », uniquement des « Tree » et les deux types de modèles (NN et Tree). La colonne « taille » indique le nombre de classificateurs individuels inclus dans l'ensemble. Ces résultats montrent que dans tous les cas, l'utilisation d'un ensemble classificateur améliore les résultats par rapport au meilleur classificateur unique dont les résultats sont présentés tables 2 et 3. Sans surprise, les moins bons résultats sont obtenus sur le jeu de données « brut » pour les ensembles classificateur construits uniquement avec des « Tree ».
Dans tous les cas, l'utilisation d'un critère de sélection basé sur la diversité permet d'améliorer à la fois la taille et les performances de l'ensemble comparativement à l'utilisation d'un critère basé sur la précision. Les meilleurs résultats sont obtenus sur le jeu de données « diff » en utilisant un ensemble uniquement constitués de « Tree » sélectionnés sur un critère de diversité.

4.2.4 Résultats sur Alima5

L'objectif final est de d'implanter un modèle de classification de comportement dans le module Alima capable de s'adapter aux variations des composants et aux variations d'emplacement (maison ou appartement, cuisine ou chambre…). Les modèles précédemment construit l'ont été pour classer un comportement lié à l'occupation de la chambre basé sur des données fournies par une Alima placée dans la cuisine. La qualité des résultats, montre qu'une Alima placée dans une pièce est capable de détecter des comportements associés à d'autres pièces (dans la limite où les portes sont gardées ouvertes ce qui était le cas dans notre étude). Afin d'étudier la portabilité des modèles construits, nous allons présenter les résultats obtenus avec ces modèles (construits avec les données de l'Alima1) sur les données fournies par l'Alima5 (située dans la salle de bain). Les tableaux 6 et 7 présentent les résultats obtenus avec les meilleurs classificateurs individuels sur les jeux de données « brut » et « diff ».

	taux mauvaise classif	taux fausse alarme	taux non détection
meilleur NN	38.0%	30.0%	63.0%
meilleur Tree	30.5%	0.0%	100.0%

Tableau 6 : Résultats des meilleurs modèles – Alima 5 – « brut »

	taux mauvaise classif	taux fausse alarme	taux non détection
meilleur NN	41.1%	29.0%	69.0%
meilleur Tree	29.2%	20.5%	49.1%

Tableau 7 : Résultats des meilleurs modèles – Alima 5 – « diff »

Ces résultats montrent que même si les classificateurs individuels donnent des résultats performants sur le jeu de données de validation de l'Alima1, ces modèles sont peu portables sur une autre Alima utilisée dans d'autres conditions. Ceci est particulièrement flagrant pour le meilleur modèle « Tree » obtenu sur les données « brut » devient, avec les données de l'Alima5, le pire modèle puisqu'il est incapable de détecter quoi que ce soit.
Les tableaux 8 et 9, dans le même esprit, présentent les résultats obtenus avec les meilleurs ensembles classificateurs sur les données « brut » et « diff » respectivement.

	sélection	taille	taux mauvaise classif	taux fausse alarme	taux non détection
NN ensemble	précision	29	27.2%	7.3%	72.8%
Tree ensemble		19	30.5%	0.0%	100.0%
classifier ensemble		189	30.5%	0.0%	100.0%
NN ensemble	diversité	5	26.5%	5.5%	74.7%
Tree ensemble		11	30.5%	0.0%	100.0%
classifier ensemble		41	30.5%	0.0%	100.0%

Tableau 8 : Résultats des meilleurs ensembles Alima 5 – « brut »

	sélection	taille	taux mauvaise classif	taux fausse alarme	taux non détection
NN ensemble	précision	15	31.7%	8.9%	94.0%
Tree ensemble		83	28.3%	17.3%	53.5%
classifier ensemble		59	27.7%	15.9%	54.7%
NN ensemble	diversité	13	31.6%	11.5%	87.0%
Tree ensemble		9	28.8%	17.9%	53.6%
classifier ensemble		35	27.6%	18.1%	48.5%

Tableau 9 : Résultats des meilleurs ensembles Alima 5 – « diff »

Ces résultats montrent que les ensembles basés sur des modèles « Tree » et exploitant les données « brut » sont incapables de détecter quoi que ce soit. Les modèles obtenus sur les données « diff » sont un peu meilleurs mais avec des taux de fausse alarme très important. L'ensemble classificateur le plus portable est le NN ensemble obtenu sur les données « brut » avec le critère de sélection basé sur la diversité. Quoi qu'il en soit, même pour ce modèle, une phase d'adaptation par réapprentissage sera nécessaire pour retrouver des performances acceptables. Si un tel réapprentissage ne pose pas de difficulté dans le cas de modèle neuronaux dont la structure est déjà définie (si l'on exclut la phase de collecte de données) il n'en est pas de même pour des arbres de décisions dont on chercherait à conserver la structure et à juste adapter les tests.

5 CONCLUSION

Cet article porte sur la construction d'un modèle de classification de situation d'usage en fonction de l'évolution de polluants relevés par une Alima. Dans cet article, nous avons présenté une étude comparative de deux outils (perceptron multicouches et arbres de décisions) pour un problème de classification d'évènements dynamiques. L'impact de l'utilisation d'ensemble classificateur et du critère de sélection pour les construire est également évalué. Une étude a également été menée pour déterminer la manière la plus performante pour présenter les données au modèle dans le cas d'un problème dynamique (utilisation des données brutes, ou de la différence entre les données actuelles et les données passées). Enfin, la portabilité des modèles d'une Alima à une autre a été étudiée. Les résultats ont montré que les arbres de décision donnent les meilleurs résultats à condition de leurs présenter les données sous une forme adaptée. L'utilisation d'ensemble classificateur permet d'améliorer les résultats, plus particulièrement lorsque le critère de sélection des classificateur individuel est basé sur la diversité. Par contre, en vue de la portabilité des modèles, l'utilisation de perceptron multicouches semble plus adapté, ces derniers donnant les moins mauvais résultats et pouvant être très simplement adaptés par réapprentissage.
Dans nos travaux futurs, comme les arbres de décisions ont fournis les meilleurs résultats initiaux, nous nous intéresserons à l'adaptation de ce type de modèle par réapprentissage et en conservant la structure.

REMERCIEMENTS

Les auteurs remercient la société ANAXIMEN pour son soutien à leurs travaux.

REFERENCES

Alima, http://getalima.com, 2013.
Aksela M., Laaksonen J., 2006. Using diversity of errors for selecting members of a committee classifier. Pattern Recognition, 39, 608-623.
Bi Y., 2012. The impact of diversity on the accuracy of evidential classifier ensembles. International Journal of Approximate Reasoning, 53, 584-607.
Breiman L., 1996. Bagging predictors. Machine Learning, 24, 123-140.
Breiman L., Friedman J.H., Olshen R.A., Stone C.J., 1984. Classification and regression trees, Chapmann & Hall, Boca Raton, USA.
Cybenko G., 1989. Approximation by superposition of a sigmoidal function. Math. Control Systems Signals, 2, 4, 303-314.
Dai Q., 2013. A competitive ensemble pruning approach based on cross-validation technique. Knowledge-Based Systems, 37, 394-414.
Dietterich T.G., 2000. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Machine Learning, 40, 139-157.
Engelbrecht A.P., 2001. A new pruning heuristic based on variance analysis of sensitivity information. IEEE transactions on Neural Networks, 1386-1399.
Freund Y., Schapire R.E., 1996. Experiments with a new boosting algorithm. 13th International Conference on Machine Learning ICML'96, Bari, Italy, July 3-6.

Funahashi K., 1989. On the approximate realization of continuous mapping by neural networks. Neural Networks, 2, 183-192.

Giacinto G., Roli F., 2001. Design of effective neural networks ensembles for image classification processes. Image Vision and Computing Journal, 19, 699-707.

Guo L., Boukir S., 2013. Margin-based ordered aggregation for ensemble pruning. Pattern Recognition Letters, 34, 603-609.

Hajek P., Olej V., 2010. Municipal revenue prediction by ensembles of neural networks and support vector machines. WSEAS Transactions on Computers, 9, 1255-1264.

Han H.G., Qiao J.F., 2013. A structure optimisation algorithm for feedforward neural network construction. Neurocomputing, 99, 347-357

Hernandez-Lobato D., Martinez-Munoz G., Suarez A., 2013. How large should ensembles of classifiers be? Pattern Recognition, 46, 1323-1336.

Ho T., 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 8, 832-844.

Jain A.K, Murty M.N., Flynn P., 1999. Data clustering: a review. ACM Computing Surveys, 31, 264-323.

Jones, A.P, 1999. Indoor air quality and health. *Atmospheric Environment*, 33, 28, 4535-4564.

Kim Y.W., Oh I.S., 2008. Classifier ensemble selection using hybrid genetic algorithms. Pattern Recognition Letters, 29, 796-802.

Ko A.H.R., Sabourin R., Britto A.S., 2008. From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition, 41, 1718-1731.

Kotsiantis, S.B., 2007. Supervised machine learning: a review of classification techniques. Informatica, 31, 249-268.

Kotsiantis S.B., 2011. Combining bagging, boosting, rotation forest and random subspace methods. Artificial Intelligence Review, 35, 225-240.

Kotsiantis S.B., 2013. Decision trees: a recent overview. Artificial Intelligence Review, 39, 261-283.

Kuncheva L.I., 2002. Switching between selection and fusion in combining classifiers: An experiment. IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics, 32, 2, 146-156.

Kuncheva L.I., 2004. Combining pattern classifiers: Methods and algorithms. Wiley-Intersciences.

Kuncheva L.I., Whitaker C.J., 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning, 51, 181-207.

Lewis R.J., 2000. An introduction to classification and regression tree (CART) analysis. Annual Meeting of the Society for Academic Emergency Medicine, San Francisco, California, May 22–25.

Ma L., Khorasani K., 2004. New training strategies for constructive neural networks with application to regression problems. Neural Network, 589-609.

Maroni M., Seifert B., Lindvall T., 1995. Indoor Air Quality – a Comprehensive Reference Book. Elsevier, Amsterdam.

Mehta M., Agrawal R., Rissanen J., 1996. SLIQ: A fast scalable classifier for data mining. Advances in Database Technology — EDBT '96, Lecture Notes in Computer Science, 1057, 18-32.

Meyer D., Leisch F., Hornik K., 2003. The support vector machine under test. Neurocomputing, 55, 169-186.

Murthy S.K., 1998. Automatic construction of decision trees from data: a multi-disciplinary survey. Data Mining and Knowledge Discovery, 2, 345-389.

Nguyen D., Widrow B., 1990. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptative weights. Proc. of the Int. Joint Conference on Neural Networks IJCNN'90, 3, 21-26.

Paliwal M., Kumar U.A., 2009. Neural networks and statistical techniques: A review of applications. Expert Systems with Applications, 36, 2-17.

Patel M.C., Panchal M., 2012. A review on ensemble of diverse artificial neural networks. Int. J. of Advanced Research in Computer Engineering and Technology, 1, 10, 63-70.

Quinlan, J.R., 1996. Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research, 4, 77-90.

Rodriguez J.J., Kuncheva L.I., Alonso C.J., 2006. Rotation forest: a new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell., 28, 1619-1630.

Ruta D., Gabrys B., 2005. Classifier selection for majority voting. Information Fusion. 6, 63-81.

Setiono R., Leow W.K., 2000. Pruned neural networks for regression. 6th Pacific RIM Int. Conf. on Artificial Intelligence PRICAI'00, Melbourne, Australia, 500-509.

Shafer J., Agrawal R., Mehta M., 1996. SPRINT: a scalable parallel classifier for data mining. 22th International Conference on Very Large Data VLDB'96, Bombay, India September 3-6.

Soto V., Martinez-Munoz G., Hernadez-Lobato D., Suarez A., 2013. A double pruning algorithm for classification ensembles. 11th International Workshop on Multiple Classifier Systems MCS'13, Nanjing, China, May 15-17.

Tang E.K, Suganthan P.N., Yao X., 2006. An analysis of diversity measures. Machine Learning, 65, 247-271.

Thomas P., Bloch G., 1997. Initialization of one hidden layer feedforward neural networks for non-linear system identification. 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics WC'97, 4, 295-300.

Thomas P., Bloch G., Sirou F., Eustache V., 1999. Neural modeling of an induction furnace using robust learning criteria. J. of Integrated Computer Aided Engineering, 6, 1, 5-23.

Thomas P., Suhner M.C., Thomas A., 2013b. Variance Sensitivity Analysis of Parameters for Pruning of a

Multilayer Perceptron: Application to a Sawmill Supply Chain Simulation Model. Advances in Artificial Neural Systems, Article ID 284570, http://dx.doi.org/10.1155/2013/284570.

Tsoumakas G., Patalas I., Vlahavas I., 2009. An ensemble pruning primer. in Applications of supervised and unsupervised ensemble methods O. Okun, G. Valentini Ed. Studies in Computational Intelligence, Springer.

Walsh P.J., Dudney C.S., Copenhaver E.D., 1987. In-door air quality, ISBM 0-8493-5015-8.

Yang L., 2011. Classifiers selection for ensemble learning based on accuracy and diversity. Procedia Engineering, 15, 4266-4270.