Réseaux bayésiens naïfs et arbres de décision dans les systèmes de détection d'intrusions

Naive Bayesian Networks and Decision Trees in Intrusion Detection Systems

N. Ben Amor[I] S. Benferhat[2] Z. Elouedi[I]

[I] Institut Supérieur de Gestion Tunis, 41 Avenue de la liberté, 2000 Le Bardo, Tunisie
{nahla.benamor, zied.elouedi}@gmx.fr
[2] CRIL - CNRS, Université d'Artois, Rue Jean Souvraz SP 18 62307 Lens, Cedex, France
benferhat@cril.univ-artois.fr

Résumé

Les réseaux bayésiens sont des outils puissants, développés dans le cadre de la théorie des probabilités, pour le raisonnement et la décision sous incertitude. Une forme très simplifiée des réseaux bayésiens est appelée réseaux bayésiens naïfs, qui disposent d'un mécanisme d'inférence particulièrement efficace.

Cet article propose une étude expérimentale des réseaux bayésiens naïfs pour la détection d'intrusions. Nous montrons que ces réseaux sont très compétitifs malgré leurs structures très simplifiées.

L'étude expérimentale est effectuée sur la base de donnée KDD'99. Trois types d'expérimentations sont réalisés en fonction du niveau de granularité des attaques: considérer toutes les attaques, analyser les cinq importantes classes d'attaques, ou encore se focaliser uniquement sur le comportement normal ou anormal d'une connexion.

Finalement, nous comparons les performances des réseaux bayésiens naïfs à celles obtenues avec des arbres de décision, qui sont des outils très utilisés dans les problèmes de classification.

Mots Clef

Raisonnement sous incertitude, Réseaux bayésiens naïfs, Arbres de décision.

Abstract

Bayesian networks are powerful tools, in the probabilistic framework, for decision and reasoning under uncertainty. A very simple form of bayesian networks are called naive bayesian networks, which are particularly efficient for inference tasks.

This paper offers an experimental study of the use of naive bayesian networks in intrusion detection. We show that even if having a simple structure, naive bayesian networks provide very competitive results.

The experimental study is done on KDD'99 intrusion data sets. We consider three levels of attack granularities depending on whether dealing with whole attacks, or grouping them in four main categories or just focusing on normal and abnormal behaviours. Different experimentations are performed in order to find best strategies to adopt regarding the different studied cases.

In the whole experimentations, we compare the performance of naive bayesian networks with one of well known machine learning techniques which is decision tree.

Keywords

Uncertain reasoning, Naive bayesian networks, Decision trees.

1 Introduction

La détection d'intrusions a pour objectif de détecter toute violation de la politique de sécurité en vigueur sur un système d'information. Elle est basée sur l'analyse à la volée ou en temps différé de ce qui se passe sur le système. On peut distinguer deux approches pour la détection d'intrusions [1]: l'approche *par scénario* et l'approche *comportementale*. Chacune des deux approches présente des points forts ainsi que des faiblesses.

L'approche par détection de scénarios d'attaque repose sur la constitution d'une base de connaissances contenant une description des attaques connues. Le module de détection d'intrusions analyse ensuite les traces d'activités journalisées à la recherche des différents événements correspondant à la description des attaques de la base de connaissances. Ces événements

représentent la signature de l'attaque. Toutefois, si la signature n'est pas suffisamment précise, des faux positifs peuvent apparaître. Par ailleurs, si la signature est trop contrainte, des variantes même très proches de l'attaque risquent de ne pas être détectées. Comme exemple de systèmes se basant sur la détection de scénarios, on peut citer les systèmes NIDES [8] et STAT [6].

Face aux difficultés de l'approche par détection de scénarios, l'approche comportementale présente certains avantages. En effet, elle repose sur l'apprentissage du comportement normal en considérant que toute déviation de ce comportement est suspecte. Donc, on n'a pas besoin de posséder une connaissance de l'attaque qui peut être menée sur le système. Plusieurs systèmes de détection d'intrusion se basant sur l'approche comportementale sont développés, on peut citer IDES [7] et NIDES [8].

Récemment, Valdes [16] a proposé une nouvelle approche hybride pour la détection d'intrusions se basant sur les réseaux bayésiens [5, 11] qui sont des outils de raisonnement avec des informations incertaines dans le cadre de la théorie des probabilités. Les réseaux bayésiens utilisent des graphes acycliques dirigés pour la représentation des relations causales et des probabilité conditionnelles (de chaque noeud dans le contexte de ses parents) pour exprimer l'incertitude sur ces relations. Valdes [16] utilise une forme simplifiée des réseaux bayésiens, appelée *réseaux bayésiens naïfs*, composée de deux niveaux: une racine qui représente la nature de la session (normal et les différents types d'attaques), et plusieurs noeuds enfants, chacun d'entre eux correspond à un attribut de la connexion.

Les réseaux bayésiens naïfs ont plusieurs avantages dus, en particulier, à leur construction qui est très simple. Par ailleurs, l'inférence (classification) est assurée de façon linéaire (alors que l'inférence dans les réseaux bayésiens qui ont une structure générale est connue comme un problème NP-complet [3]). En plus, la construction des réseaux bayésiens naïfs est incrémentale, dans le sens qu'elle peut être facilement mise à jour (notamment, il est toujours possible de prendre en considération de nouvelles classes). Cependant, les réseaux bayésiens naïfs travaillent sous une hypothèse d'indépendance très forte entre les attributs dans le contexte de la nature de la session. Une telle hypothèse n'est pas toujours valable et peut entacher les résultats.

L'objectif de cet article est de présenter des résultats expérimentaux montrant que les réseaux bayésiens naïfs, avec leur structure simple et leur hypothèse forte sur l'indépendance, peuvent être compétitifs. L'étude expérimentale est effectuée sur la base de données KDD'99 [17]. Les données KDD'99 sont au fait des données formatées fournies par DARPA à MIT Lincoln Lab. Ces données représentent 7 semaines de données libellées pour l'apprentissage et 2 semaines de données non libellées pour le test (Correspondant au traffic réseau simulant un réseau local d'US Air force) . Ces données sont très appropriées pour évaluer un système de détection d'intrusion de type comportementale puisque l abase des tests contient des attaques qui ne figurent pas dans la base d'apprentissage . Il existe des travaux récents utilisant ces données [12, 16], cependant, notre étude expérimentale donne une nouvelle perspective. En particulier, les expérimentations considèrent trois niveaux de granularités des attaques: cas où on traite toutes les attaques, cas où on les regroupe en quatre classes principales et le cas où on se focalise uniquement sur le comportement normal ou anormal d'une connexion.

Par ailleurs, des résultats basés sur la discrétisation des domaines continus de certains attributs sont présentés.

Dans cet article, en utilisant les mêmes données, nous étudions et confrontons les résultats fournis par les réseaux bayésiens naïfs, à ceux donnés par des arbres de décision [13, 14] considérés comme une des méthodes de classification les plus utilisées. Nous montrons que les réseaux bayésiens naïfs sont réellement très compétitifs avec les arbres de décision.

La Section 2 et la Section 3 présentent, respectivement, les notions de base des arbres de décision et des réseaux bayésiens. La Section 4 présente les données KDD'99. La Section 5 présente le plan d'expérimentation utilisé dans cet article. La Section 6 se focalise sur toutes les attaques en utilisant les arbres de décision et les réseaux bayésiens naïfs. Puis, la Section 7 présente les résultats sur les quatre catégories d'attaques (c.à.d DOS, R2L, U2R, PROBING) lorsque le regroupement est effectué avant ou après la classification. La Section 8 traite le cas des connexions normales et anormales. La Section 9 présente quelques résultats sur la discrétisation des domaines continus. Finalement, la Section 10 résume les principaux résultats et conclut l'article.

2 Arbres de décision

Les arbres de décision représentent l'une des techniques les plus connues et les plus utilisées en classification. Leur succès est notamment dû à leur aptitude à traiter des problèmes complexes de classification. En effet, ils offrent une représentation facile à comprendre et à interpréter, ainsi qu'une capacité à produire des règles logiques de classification.

Nous donnons ici un bref rappel des arbres de décision. Pour un exposé détaillé voir [13] et [14].

Un arbre de décision est composé de:

- **Noeuds de décision** contenant chacun un test sur un attribut.

- **Branches** correspondant généralement à l'une des valeurs possibles de l'attribut sélectionné.

- **Noeuds feuilles** comprenant les objets qui appartiennent à la même classe.

L'utilisation des arbres de décision dans les problèmes de classification se fait en deux principales étapes:

1. *Etape de construction de l'arbre.* Cette étape se base sur un ensemble d'apprentissage, et consiste à sélectionner, pour chaque noeud de décision, l'attribut test 'approprié'. Elle consiste aussi à définir la classe libellant chaque feuille.

2. *Etape de classification.* Pour trouver la classe de l'objet, il suffit de suivre le chemin en partant de la racine de l'arbre de décision jusqu'aux feuilles en effectuant les différents tests à chaque noeud selon les valeurs des attributs de l'individu à classer.

Plusieurs algorithmes ont été développés afin d'assurer la phase de construction. Parmi les algorithmes les plus utilisés, nous citons **ID3** et **C4.5** dévelopés par Quinlan [13], [14] ou encore l'algorithme **CART** de Breiman et al. [2].
Les différents algorithmes se basent généralement sur trois paramètres indispensables:

- **Mesure de sélection d'attributs** qui permet de choisir l'attribut qui sera la racine de l'arbre. De façon récursive, elle permet le choix des racines des sous arbres de décision. La mesure de sélection d'attributs permet de classer les attributs entre eux et de choisir celui qui partitionne l'ensemble d'apprentissage de la manière la plus optimale.
La mesure de sélection d'attributs est généralement basée sur la théorie de l'information. Nous citons comme mesures celles proposées par Quinlan: **le gain d'information** [13] et **le ratio de gain** [14]. Dans cet article, nous utilisons le ratio de gain, utilisé dans l'algorithme C4.5 [14] qui pour chaque attribut A_k et pour un ensemble d'objets T, est défini comme suit:

$$Gain(T, A_k) = Info(T) - Info_{A_k}(T)$$

où

$$Info(T) = -\sum_{i=1}^{n} \frac{freq(c_i, T)}{|T|} log_2 \frac{freq(c_i, T)}{|T|}$$

$$Info_{A_k}(T) = \sum_{a_k \in D(A_k)} \frac{|T_{a_k}^{A_k}|}{|T|} Info(T_{a_k}^{A_k})$$

$freq(C_i, T)$ représente le nombre d'objets dans l'ensemble T appartenant à la classe C_i et $T_{a_k}^{A_k}$ est le sous ensemble d'objets pour qui l'attribut A_k a la valeur a_k (appartenant au domaine de A_k noté $D(A_k)$).
Ainsi l'attribut ayant le gain d'information le plus élevé est sélectionné comme la racine de l'arbre (ou du sous arbre) de décision.
Bien que le critère de gain d'information donne de bons résultats, il présente un sérieux inconvénient. En effet, il favorise les attributs ayant un nombre élevé de valeurs [14]. Pour remédier à cet inconvénient, Quinlan propose une sorte de normalisation connue sous le nom de **ratio de gain** qui n'est autre que le gain d'information calibré par Split Info:

$$Gain \text{ ratio}(T, A_k) = \frac{Gain(T, A_k)}{Split\ Info(A_k)}$$

où $Split\ Info(A_k)$ est définie comme étant l'information contenue dans l'attribut A_k [14]:

$$Split\ Info(T, A_k) = -\sum_{a_k \in D(A_k)} \frac{|T_{a_k}^{A_k}|}{|T|} log_2 \frac{|T_{a_k}^{A_k}|}{|T|}$$

- **Stratégie de partionnement** qui permet de partitionner l'ensemble d'apprentissage courant en tenant compte de l'attribut test.
- Pour les attributs *discrets* (avec un nombre fini de valeurs), la stratégie consiste à tester toutes les valeurs possibles de l'attribut
- Pour les attributs *continus*, la méthode utilisée, dans cet article, considère des partitions binaires des attributs continus et consiste à trouver à chaque niveau de l'arbre, le point de coupure approprié [14].

- **Critères d'arrêt** qui permettent de traiter les conditions d'arrêt du développement d'une partie d'un arbre ou d'un sous arbre. Ainsi, ils déterminent si un sous ensemble d'apprentissage doit être partitionné ou pas.
En plus des phases de construction et de classification, la plupart des algorithmes d'arbres de décision présente une étape d'*élagage* qui est liée à la construction de l'arbre [9, 14]. Il s'agit généralement de supprimer les parties 'inutiles' de l'arbre et de les remplacer par un noeud terminal qui représente la classe majoritaire des individus classés par cette partie de l'arbre. Ainsi, au lieu d'avoir un arbre complet, on aura un arbre plus petit qui donne une meilleure performance de classification.

3 Les réseaux bayésiens naïfs

Les réseaux bayésiens [5, 11] sont des outils de représentation de connaissances en présence d'incertitude. Le succès de ces modèles est fortement lié à leur capacité de représenter et de manipuler des relations de (in)dépendance qui sont importantes pour une gestion efficace des informations incer-

taines. Les réseaux bayésiens utilisent une représentation basée sur le conditionnement, où les connaissances sont structurées sous la forme d'un graphe acyclique orienté. Les noeuds représentent des variables et les arcs codent le lien causal (ou l'influence) entre ces variables. L'incertitude est représentée au niveau de chaque noeud en explicitant toutes les probabilités conditionnelles attachées aux valeurs associées à ce noeud sachant celles de ses parents. Cette incertitude exprime la force de la relation de *causalité* entre les variables.
Une simple variante des réseaux bayésiens est appelée réseaux bayésiens naïfs. Ces réseaux ont une structure unique qui se compose de deux niveaux seulement. Le premier contient un seul noeud parent qui n'est pas observé et le second plusieurs enfants de ce noeud correspondant aux noeuds observés. Les réseaux bayésiens naïfs travaillent sous la forte hypothèse d'indépendance entre les noeuds enfants dans le contexte de leur parent.
L'utilisation des réseaux bayésiens naïfs est assurée en considérant le noeud parent comme un noeud caché précisant à quelle classe appartient chaque objet de la base de données et les noeud enfants représentent les différents attributs spécifiant cet objet.
En présence d'un ensemble d'apprentissage on doit juste calculer les probabilités conditionnelles puisque la structure du graphe est unique. Ce calcul peut être résumé comme suit:
- Les probabilités conditionnelles pour les attributs *discrets* sont calculées à partir des fréquences en comptant combien de fois chaque valeur d'attribut apparaît avec chaque valeur possible du noeud parent.
- Les attributs *continus* sont généralement traités en supposant qu'ils suivent une distribution de probabilité *normale*. Cette hypothèse peut être considérée comme une restriction des réseaux bayésiens naïfs puisque des attributs peuvent ne pas suivre une destribution normale. Dans ce cas, nous pouvons utiliser une méthode non-parametrique telle que *l'estimation de densité par le noyau* qui ne suppose aucune distribution particulière pour les attributs continus [4]. Une autre alternative serait de simplement de discrétiser les attributs continus.
Dans notre travail, nous avons utilisé les distributions normales et l'estimation de densité par noyau. Cependant dans la suite, nous ne présentons que l'estimation de densité par noyau, puisque dans toutes les expérimentations, elle a donné de meilleurs résultats
Une fois le réseau quantifié, il peut être utilisé pour classer de nouveaux objets étant donné leurs valeurs d'attributs en utilisant la règle de Bayes exprimée par:

$$P(c_i \mid A) = \frac{P(A \mid c_i) \cdot P(c_i)}{P(A)}$$

où c_i est une valeur possible de la classe C et A est l'évidence totale sur les attributs. L'évidence A peut être vue comme un vecteur d'insatnces $a_1, a_2, ..., a_n$ relatives aux attributs $A_1, A_2, ..., A_n$, respectivement. Puisque les réseaux bayésiens naïfs travaillent sous l'hypothèse que ces attributs sont indépendants (sachnat le noeud parent C), leur probabilité jointe peut être calculée comme suit:

$$P(c_i \mid A) = \frac{P(a_1 \mid c_i) \cdot P(a_2 \mid c_i) \cdot ... \cdot P(a_n \mid c_i) \cdot P(c_i)}{P(A)}$$

Notons que nous n'avons pas besoin de calculer explicitement le dénominateur $P(A)$ puisqu'il est déterminé par la condition de normalisation. Donc il est suffisant de calculer pour chaque classe c_i son degré de vraisemblance, c.à.d $P(a_1 \mid c_i) \cdot P(a_2 \mid c_i) \cdot ... \cdot P(a_n \mid c_i) \cdot P(c_i)$ afin de classer n'importe quel nouveau objet caractérisé par ses valeurs d'attributs: $a_1, a_2, ..., a_n$.

4 Description de KDD'99

Les données, utilisées dans cet article, sont proposées par KDD'99 et sont orientées détection d'intrusions [17]. Elles représentent des lignes de TCP/IP dump où chaque ligne est une connexion caractérisée par 41 attributs tels que la durée de la connexion, le type du protocole, etc. En tenant compte des valeurs de ses attributs, chaque connexion dans KDD'99 est considerée comme étant une connexion normale ou bien une attaque.
La base KDD recense 38 attaques possibles qui peuvent être regroupées en quatre catégories (listés dans la Table 1):

- **Déni de Service - Denial-Of-Service (DOS):** Il s'agit d'empêcher par tous les moyens les utilisateurs de se servir des ressources disponibles en temps normal. Ces attaques sont à but purement "destructeur", au sens où une telle attaque ne permet pas à l'attaquant de compromettre une machine, ni de voler des informations. De telles attaques sont souvent très simples à mettre en place et donnent une sensation de puissance à l'attaquant, ce qui explique leur fréquence.

- **Les attaques de type Remote to User (R2L):** Pour ce genre d'attaque, les attaquants essaient d'exploiter la vulnérabilité du système afin de contrôler la machine distante.

- **Les attaques de type User to Root Attacks (U2R):** Où l'attaquant essaie d'avoir les droits d'accès à partir d'un poste afin d'accéder au système.

- **Reconnaissance - Probing:** Ces attaques ne sont pas "destructrices" au sens où elles empêchent une entité de fonctionner correctement, mais permettent d'acquérir des informations parfois cruciales pour mener une attaque de plus grande envergure plus tard.

Table 1: Types d'attaques

DOS	Probing	R2L	U2R
Apache2	Ipsweep	Ftp_write	Buffer_overflow
Back	Mscan	Guess_passwd	Httptunnel
Land	Nmap	Imap	Loadmodule
Mailbomb	Portsweep	Multihop	Xterm
Neptune	Saint	Named	Perl
Pod	Satan	Phf	Ps
Processtable		Dict	Rootkit
Smurf		Snmpguess	
Teardrop		Spy	
Udpstorm		Sqlattack	
		Warezclient	
		Warezmaster	
		Xlock	
		Xsnoop	
		Guest	

Les attributs caractérisant chaque connexion, sont détaillés dans la Table 2. En effet, on peut distinguer les *attributs basiques* des connexions TCP individuelles, les *attributs relatifs au contenu*, les *attributs relatifs au temps* calculés en utilisant des fenêtres de temps de deux secondes et les *attributs basés sur le hôte* calculés en utilisant des fenêtres de temps de 100 connexions. Ces attributs sont utilisés pour caractériser les attaques qui scanent les hosts (ou les ports) en utilisant un intervalle de temps plus large que deux secondes.

5 Cas d'expérimentation

Nous traitons 10% de la base KDD'99 correspondant à 494019 de connexions d'apprentissage. La base de données test contient 311029 de connexions. La taille de la base de test est donc *à* peu pres 6% de la taille de la base d'apprentissage.

Les différentes expérimentations que nous présentons dans ce article tiennent compte des points suivants:

TROIS NIVEAUX DE GRANULARITÉ D'ATTAQUES: Nous dintinguons trois niveaux de granularités:

- *Multi-classes*: qui traite toutes les attaques présentes dans les bases KDD'99 et listées dans la Table 1, ainsi que la classe connexion normale.

- *Cinq-classes*: qui ne considère que les quatre catégories d'attaque (c.a.d. DOS, R2L, U2R, Probing). Dans la base d'apprentissage il y a 19.65% (resp. 79.07%, 0.23%, 0.22%, 0.83%) de connexions d'apprentissage normales (resp. DOS, R2L, U2R, Probing) et 19.48% (resp. 73.90%, 5.21%, 0.07%, 1.34%) de connexions de test normales (resp. DOS, R2L, U2R, Probing).

- *Deux-classes*: c.à.d. Normale et Anormale en groupant toutes les attaques en une même et unique classe (c.á.d Anormale).

Table 2: Liste des attributs

	Attributs basiques
A1	durée de la connexion (nb de secondes)
A2	type du protocole, ex. tcp, udp, etc.
A3	service réseau (destination) ex. http, telnet
A4	statut de la connexion (normal ou erreur)
A5	nb de données (en octets) de la source vers la destination
A6	nb de données (en octets) de la destination vers la source
A7	1 si la connexion est de/vers le même hôte/port; 0 autrement
A8	nb de fraguements "erronnés"
A9	nb de paquets urgents
	Attributs relatifs au contenu
A10	nb d'indicateurs "hot"
A11	nb d'essais login ratés
A12	1 si succés du login ; 0 autrement
A13	nb de conditions de "compromis"
A14	1 si la racine shell est obtenu; 0 autrement
A15	1 si la commande on a la commande "racine su" ; 0 autrement
A16	nb d'accés à la "racine"
A17	nb de créations d'opérations de fichiers
A18	nb de shell prompts
A19	nb d'opérations sur les fichiers de contrôle d'accès
A20	nb de commandes outbound dans une session ftp
A21	1 si le login appartient à la liste "hot" ; 0 autrement
A22	1 si le login est login "invité" ; 0 autrement
	Attributs basés sur le temps utilisant des fenêtres de temps de deux secondes
A23	nb de connex. pour le *même hôte*
A24	nb de connex. pour le *même service*
A25	% de connex. pour le *même hôte* ayant l'erreur "SYN"
A26	% de connex. pour le *même service* ayant l'erreur "SYN"
A27	% de connex. pour le *même hôte* ayant l'erreur "REJ"
A28	% de connex. pour le *même service* ayant l'erreur "REJ"
A29	% de connex. pour le *même hôte* utilisant le *même service*
A30	% de connex. pour le *même hôte* utilisant *différents services*
A31	% de connex. pour le *même service* utilisant *différents hosts*
	Attributs basés sur le temps utilisant des fenêtres de temps de 100 connex.
A32	nb de connex. pour le *même hôte*
A33	nb de connex. pour le *même hôte* utilisant le *même service*
A34	% de connex. pour le *même hôte* utilisant le *même service*
A35	% de connex. pour le *même hôte* utilisant *différents services*
A36	% de connex. pour le *même hôte* ayant le port src
A37	% de connex. pour le *même hôte* et le *même service* utilisant *différents hosts*
A38	% de connex. pour le *même hôte* ayant l'erreur "SYN"
A39	% de connex. pour le *même hôte* et le *même service* ayant l'erreur "SYN"
A40	% de connex. pour le *même hôte* ayant l'erreur "REJ"
A41	% de connex. pour le *même hôte* et le *même service* ayant l'erreur "REJ"

GROUPEMENT DES ATTAQUES: il y a deux stratégies de groupement des résulats, avant ou après la phase de classification:

- *Groupement avant classification*: L'idée est de modifier l'ensemble des données. Dans le cas des cinq attaques, le groupement se fait dans l'une des quatre catégories (DOS, R2L, U2R, ou Probing) listées dans la Table 1. Dans le cas deux classes, toutes les attaques sont groupées dans la classe anormale.

- *Groupement après classification*:
  - Pour le cas des cinq classes, l'ensemble d'apprentissage reste inchangé. Cependant, chaque connexion classée parmi les 38 attaques est assignée à l'une des quatre catégories à laquelle elle appartient.
  - Pour le cas de deux classes, il y a deux possibilités de regroupement: soit nous ne modifions pas l'ensemble d'apprentissage et chaque connexion classée parmi les 38 attaques est simplement libellée anormale, soit nous commencons par modifier l'ensemble d'apprentissage en groupant les attaques en quatre catégories, puis chaque connexion classée dans l'une de ces catégories sera libellée anormale.

DISCRÉTISATION: comme on peut le voir dans la Table 2, certains attributs sont discrets tels que le type du protocole, alors que d'autres sont continus tels que la durée. Ceci n'est pas le cas de tous les attributs continus du fait que dans certains cas, le type est ambigu. Typiquement, certains attributs sont déclarés comme étant continus mais ils prennent un nombre fini de valeurs. Ceci est particulièrement vrai pour les attributs de type pourcetntage, comme le pourcentage de connexions au même service ayant l'erreur "SYN" qui peut prendre au plus 101 (de 0.0 à 1.0). Donc, l'idée est de discrétiser ces attributs afin de tester leur incidence sur les résultats de classification. Dans la suite nous présentons les expérimantations traitant toutes les attaques (Section 6), traitant les quatre classes d'attaque (Section 7), se focalisant sur normal / anormal (Section 8), et montrant les apports de l'étape de discrétisation (Section 9).
Dans chacune des expérimentations effectuées, l'évaluation de l'éfficacité de classification est basée sur le *pourcentage de classification correcte* (PCC) des instances appartenant à l'ensemble test défini par:

$$PCC = \frac{\text{nombre des instances correctement classées}}{\text{nombre des instances classées}}$$

Nous présentons aussi, les matrices de confusion pour le cas de multi-classe et deux classes (la diagonale de ces matrices correspond au critère "*recall*"). Dans toutes les expérimentations, les arbres de décision et les réseaux bayésiens sont évalués exactement sur les mêmes données test.

6 Traitement de toutes les attaques

La Table 3 présente les valeurs du PCC relatives aux ensembles d'apprentissage et test pour le cas multi-classes. La Table 3 montre que les arbres de décision ainsi que les réseaux bayésiens naïfs sont complètement en accord avec l'ensemble d'apprentissage qui est donc cohérent. En d'autres termes, la majorité des instances d'apprentissage caractérisées par les mêmes valeurs d'attributs appartiennent à la même classe. Ce comportement reste également le même pour la phase de classification avec un petit avantage pour les arbres de déscision.

Table 3: PCC relatifs au cas des multi-classes

ENSEMBLE D'APPRENTISSAGE	ENSEMBLE TEST
Arbre de décision non élagué	
99.99%	91.41%
Arbre de décision élagué	
99.98%	91.42%
Réseau bayésien naïf	
99.64%	91.13%

Notons que les résultats obtenus avec les arbres de décision élagués et non élagués sont pratiquement les mêmes. La seule différence réside dans la taille de l'arbre qui contient 816 noeuds s'il est élagué et 1228 s'il est non élagué. Pour des raisons de simplicité, dans ce qui suit nous allons présenter uniquement les résultats relatifs aux arbres de décision non élagués.

7 Traitement des quatre catégories d'attaques

Afin de mieux sélectionner la stratégie permettant le traitement des quatre catégories d'attaques principales, nous avons groupé les attaques appartenant à la même classe. Ce traitement est effectué avant à après la classification. Pour le dernier cas, nous avons utilisé les résultats relatifs à toutes les attaques en additionnant le nombre d'occurrences relatives à chaque catégorie d'attaque (DOS, R2L, U2R, Probing). La Table 4 donne les PCC relatifs à ces deux expérimentations.

Table 4: PCC relatifs aux cinq classes (les valeurs entre parenthèses sont relatives au groupement des résultats de toutes les attaques en 5 classes après classification)

ENSEMBLE D'APPRENTISSAGE	ENSEMBLE TEST
Arbre de décision non élagué	
99.99% (99.99%)	92.28% (91.81%)
Réseau bayésien naïf	
98.85% (99.67%)	90.83% (91.40%)

Comme dans le cas de toutes les attaques, les PCC relatifs aux arbres de décision dépassent légèrement ceux relatifs aux réseaux bayésiens naïfs dans les phases d'apprentissage et de classification. Notons que le regroupement avant et après la classification n'a aucune influence sur les arbres de décision. Cependant, avec les réseaux bayésiens naïfs il est préférable d'effectuer le regroupement après la classification. La Table 5 présente les matrices de confusion. Elle montre que les connexions Normal, DOS et Probing sont bien classées avec les deux techniques. Ceci n'est pas le cas avec les connexions R2L et U2R qui sont toujours mal-classées.

Table 5: Matrices de confusion relatives aux cinq classes (Les valeurs entre parenthèses sont relatives au regroupement des résultats de toutes les attaques en cinq classes après classification)

Arbre de décision non élagué					
→	Normal	DOS	R2L	U2R	Probing
Normal (60593)	**99.50%** (**99.43%**)	0.13% (0.14%)	0.01% (0.02%)	0.01% (0.02%)	0.36% (0.39%)
DOS (229853)	2.76% (2.94%)	**97.24%** (**96.57%**)	0.00% (0.10%)	0.00% (0.00%)	0.00% (0.39%)
R2L (16189)	96.55% (75.77%)	0.02% (2.79%)	**0.52%** (**0.45%**)	0.15% (4.27%)	2.76% (16.71%)
U2R (228)	79.82% (23.25%)	2.63% (0.00%)	1.75% (5.26%)	**7.89%** (**13.60%**)	7.89% (57.89%)
Probing (4166)	19.54% (15.22%)	5.16% (6.67%)	0.34% (0.19%)	0.00% (0.00%)	**74.96%** (**77.92%**)
PCC	92.28% (91.81%)				

Réseau bayésien naïf					
→	Normal	DOS	R2L	U2R	Probing
Normal (60593)	**96.97%** (**98.54%**)	2.86% (1.16%)	0.01% (0.02%)	0.00% (0.05%)	0.17% (0.24%)
DOS (229853)	3.32% (3.16%)	**96.25%** (**96.40%**)	0.02% (0.00%)	0.00% (0.00%)	0.41% (0.44%)
R2L 16189)	94.78% (99.39%)	5.16% (0.00%)	**0.02%** (**0.16%**)	0.04% (0.45%)	0.00% (0.00%)
U2R (228)	96.49% (95.61%)	0.00% (0.00%)	0.44% (0.00%)	**1.75%** (**3.07%**)	1.32% (1.32%)
Probing (4166)	30.36% (28.78%)	8.43% (0.38%)	0.79% (0.00%)	0.05% (0.00%)	**60.37%** (**70.84%**)
PCC	90.83% (91.40%)				

Expliquons maintenant les raisons du mauvais classement des connexions U2R et R2L. Concernant les arbres de décision, signalons d'abord que ce comportement ne reflète pas les résultats optimistes obtenus avec l'ensemble d'apprentissage où 82.69% (resp. 98.93%) des connexions U2R (resp. R2L) sont bien-classées. Ensuite les proportions, dans l'ensemble d'apprentissage, des attaques U2R et R2L sont très faibles (0.22% pour U2R et 0.23% pour R2L).
Dans les arbres de décision, lorsqu'une classe est faiblement représentée dans l'ensemble d'apprentissage, alors cette dernière est mal apprise, ce qui entraîne une mauvaise classification des connexions tests appartenant réellement à cette classe.
Finalement, le mauvais classement des connexions R2L et U2R est expliqué par le fait que certaines valeurs d'attributs dans les connexions test R2L (resp. U2R) n'apparissent pas dans les connexions R2L (resp. U2R) de la base d'apprentissage.
Afin d'illustrer ceci, nous allons analyser la base de règles relative aux attaques U2R induite par l'arbre de décision (noter que le nombre de règles dans nos expérimentations est en général entre 1200 et 1300 règles) :

- **R1:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 \leq 0$, $A36 \leq 0.99$, $A29 > 0.32$, $A4 = SF$, $A5 > 6$, $A18 \leq 0$, $A39 \leq 0.03$, $A6 > 6$, $A5 \leq 36530$, $A17 \leq 0$, $A33 \leq 4$, $A3 = telnet$, $A6 \leq 1342$, $A1 > 20$ **alors** $U2R$
- **R2:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 \leq 0$, $A36 \leq 0.99$, $A29 > 0.32$, $A4 = SF$, $A5 > 6$, $A18 \leq 0$, $A39 \leq 0.03$, $A6 > 6$, $A5 \leq 36530$, $A17 > 0$, $A33 \leq 3$ **alors** $U2R$
- **R3:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 \leq 0$, $A36 \leq 0.99$, $A29 > 0.32$, $A4 = SF$, $A5 > 6$, $A18 > 0$, $A3 = telnet$ **alors** $U2R$
- **R4:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 \leq 0$, $A36 > 0.99$, $A5 \leq 33$, $A5 \leq 19$, $A32 \leq 1$, $A2 = tcp$, $A7 = 0$, $A3 = ftp\_data$, $A4 = SF$ **alors** $U2R$
- **R5:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 \leq 0$, $A36 > 0.99$, $A5 \leq 33$, $A5 \leq 19$, $A32 > 1$, $A6 > 1$, $A3 = ftp\_data$, $A12 = 1$ **alors** $U2R$
- **R6:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 > 0$, $A4 = SF$, $A5 \leq 971$, $A5 \leq 132$, $A37 > 0.01$ **alors** $U2R$
- **R7:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 > 0$, $A4 = SF$, $A5 \leq 971$, $A5 \leq 132$, $A1 \leq 140$, $A33 \leq 2$, $A17 > 2$ **alors** $U2R$
- **R8:** **si** $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A6 > 6$, $A40 \leq 0.45$, $A11 \leq 0$, $A5 \leq 1$ **alors** $U2R$
- **R9:** **si** $A23 \leq 64$, $A39 \leq 0.82$ $A13 > 0$, $A5 \leq 10073$, $A6 > 0.09$, $A13 \leq 13$, $A3 = telnet$, $A6 \leq 6223$ **alors** $U2R$
- **R10:** **si** $A23 > 64$, $A6 > 1$, $A33 \leq 69$, $A17 > 0$ **alors** $U2R$

En analysant le sous ensemble relatif aux attaques U2R, nous avons noté, en premier lieu, que toutes les valeurs relatives à l'attribut $A23$, qui est la racine de l'arbre, sont inférieures ou égales à 64 ce qui exclut l'utilisation de la règle **R10**. Par ailleurs, nous avons remarqué que la majorité des connexions tests de U2R

devraient suivre les règles de **R1** à **R5** (187 de ces connexions satisfont $A23 \leq 64$, $A39 \leq 0.82$, $A13 \leq 0$, $A8 \leq 0$, $A37 \leq 0.48$, $A35 \leq 0.91$, $A10 \leq 0$). Cependant, dans toutes ces règles la valeur de $A4$ doit être égale à SF ce qui n'est pas le cas dans la majorité des connexions tests et ceci explique leur mauvaise classification.
En effet, $A4$ apparaît, dans les connexions test, avec la valeur REJ qui n'apparaît jamais dans l'ensemble d'apprentissage avec les attaques U2R.
La même explication est valable avec les réseaux bayésiens naïfs puisque dans la phase d'apprentissage la probabilité conditionnelle de REJ dans le contexte de U2R va être égale à zéro ($P(REJ \mid U2R) = 0$). Par conséquent, les connexions tests appartenant, réellement, à U2R mais ayant la valeur REJ avec l'attribut $A4$ vont être mal-classées.

8 Traitement des connexions normales et anormales

Dans cette section, nous nous intéressons au comportement normal. Afin d'assurer cet objectif, nous avons étudié les matrices de confusion et les valeurs du PCC uniquement sur 2 classes: normal et anormal. Nous considérons, en premier lieu, le cas où le regroupement de toutes les attaques s'est effectué avant classification, ensuite nous traitons le cas où le regroupement est effectué après la classification en utilisant les résultats de toutes les attaques et ceux relatifs aux cinq classes. Les résultats sont résumés dans la Table 6 où on remarque que les valeurs du PCC relatives aux deux classes (normal et anormal) est très élevées.
Ces valeurs montrent aussi que, comme pour les autres expérimentations, l'écart entre les arbres de décision et les réseaux bayésiens naïfs est insignifiant puisque le PCC est pratiquement le même avec ces deux approches avec un léger avantage pour les arbres de décision. Plus précisément, la matrice de confusion présentée par la Table 7 montre que les arbres de décision sont légèrement meilleurs que les réseaux bayésiens naïfs sauf dans le cas des connexions normales lorsque le regroupement est effectué avant la classification.

Table 6: PCC relatifs aux connexions normales et anormales (les valeurs entre parenthèses sont relatives au regroupement des résultats de toutes les attaques et de ceux relatifs au cinq classes en deux classes après classification)

ENSEMBLE D'APPRENTISSAGE	ENSEMBLE TEST
Arbre de décision non élagué	
99.99% (99.99%, 99.99%)	93.02% (93.55%, 92.52%)
Réseau bayésien naïf	
98.75% (99.69%, 98.91%)	91.45% (91.75%, 91.55%)

Table 7: Matrice de confusion relative aux classes normales et anormales (les valeurs entre parenthèses sont relatives au regroupement des résultats de toutes les attaques et de ceux relatifs au cinq classes en deux classes après classification)

Arbre de décision non élagué		
→	Normal	Anormal
Normal (60593)	**98.24% (99.43%, 98.50%)**	0.61% (0.57%, 0.50%)
Abnormal (250436)	8.52% (7.87%, 9.17%)	**91.48% (92.13%, 90.83%)**
PCC	93.02% (93.55%, 92.52%)	

Réseau bayésien naïf		
→	Normal	Anormal
Normal (60593)	**98.48% (98.54%, 96.97%)**	1.52% (1.46%, 3.03%)
Anormal (250436)	10.25% (9.89%, 9.76%)	**89.75% (90.11%, 90.24%)**
PCC	91.45% (91.75%, 91.55%)	

9 Discrétisation

L'idée de cette expérimentation est de raffiner l'analyse des 32 attributs continus énumérés dans la Table 2. Nous avons remarqué qu'il existe 15 attributs définis comme des pourcentages (A25,..., A31, A34,..., A41).
Ces attributs sont traités comme continus, mais en réalité ils sont discrets puisqu'ils peuvent prendre 101 valeurs au maximum (entre 0.00 et 1.00). Par conséquent, leur discrétisation peut être directement effectuée en énumérant les valeurs possibles qu'ils peuvent prendre.
Les résultats de cette expérimentation sont résumés dans la Table 8, pour les trois cas de figures: muulticlasses, cinq classes et normal/anormal.

Table 8: PCC après discrétisation (les valeurs entre parenthèses sont relatives au regroupement des résultats de toutes les attaques et de ceux relatifs aux cinq classes en deux classes après classification)

MULTI-CLASSES	CINQ-CLASSES	NORMAL ET ANORMAL
Arbre de décision non élagué		
91.69%	92.06% (92.8%)	92.22% (92.28%, 92.41%)
Réseau bayésien naïf		
91.20%	91.48% (92.10%)	91.45% (92.69%, 92.40%)

Table 9: Matrices de confusion relatives aux cinq classes après discrétisation avec les réseaux bayésiens naïfs (les valeurs entre parenthèses sont relatives au regroupement des résultats de toutes les attaques en cinq classes après classification)

→	Normal	DOS	R2L	U2R	Probing
Normal	**96.64%**	2.78%	0.23%	0.11%	0.25%
(60593)	(**97.68%**)	(1.29%)	(0.30%)	(0.15%)	(0.58%)
DOS	3.09%	**96.38%**	0.10%	0.00%	0.43%
(229853)	(2.75%)	(**96.65%**)	(0.01%)	(0.00%)	(0.58%)
R2L	85.09%	4.84%	**7.11%**	2.92%	0.04%
(16189)	(88.80%)	(0.01%)	(**8.66%**)	(1.54%)	(0.99%)
U2R	54.39%	0.00%	8.33%	**11.84%**	25.44%
(228)	(76.32%)	(0.00%)	(3.95%)	(**10.96%**)	(8.77%)
Probing	14.19%	3.36%	3.79%	0.48%	**78.18%**
(4166)	(10.80%)	(0.38%)	(0.38%)	(0.10%)	(**88.33%**)
PCC	91.48% (92.10%)				

La discrétisation dans les réseaux bayésiens naïfs apporte une amélioration (bien que minimale) du PCC dans presque toutes les expérimentations. Par ailleurs, une analyse approfondie de la matrice de confusion relative au cas des cinq classes, présentée par la Table 9, montre que le PCC relatif à chacune des cinq catégories (normal, DOS, R2L, U2R, Probing) s'est amélioré et que dans certains cas tels que R2L et Probing, ces valeurs sont meilleures que celles obtenues dans arbres de décision (voir Table 5). Notons, par ailleurs, qu'il n'y a pas d'amélioration considérable avec les arbres de décision lorsque nous discrétisons ces attributs.

10 Conclusions

La Table 10 résume les PCC des résulats expérimentaux donnés dans cet article. Les trois principales conclusions sont les suivantes:

- A partir de la Table 10, nous remarquons que considérer toutes les attaques, les cinq principales attaques ou uniquement les 2 classes d'attaques, n'affecte pas considérablement les résultats de classification. La différencene ne dépasse jamais 1% entre les différentes stratégies.

  Pour le cas normal/anormal, ce résultat signifie que les performances ne dépendent pas d'une stratégie de regroupements particuliers.

  Cependant de point de vue calculatoire, le choix de la stratégie peut avoir son importance. Par exemple si on s'interesse uniquement au comportement normal / anormal, il est préférable de regrouper les attaques avant classification.

  En effet, dans les réseaux bayésiens le nombre de paramètres (probabilités conditionnelles) à stocker dans le regroupement avant classification est à peu près 20 fois le nombre de paramètres à stocker dans le cas où le regroupement se fait après classification.

  Dans le cas des quatre classes d'attaques, si la différence au niveau des PCC globaux n'est pas significative, la différence entre les PCC locaux peut être importante. Par exemple, dans le cas de la classe Probing, avec les réseaux bayésiens, la différence dépasse 10%.

Table 10: Résumé des expérimentations (les valeurs entre parenthèses sont relatives au regroupement des résultats de toutes les attaques et de ceux relatifs aux cinq classes en deux classes après classification)

MULTI-CLASSES	CINQ-CLASSES	NORMAL ET ANORMAL
Arbre de décision non élagué		
91.41%	**92.28%** (91.81%)	93.02% (**93.55%**, 92.52%)
Arbre de décision non élagué avec discrétisation		
91.69%	92.06% (92.8%)	92.22% (92.28%, 92.41%)
Réseau bayésien naïf		
91.13%	90.83% (91.40%)	91.45% (91.40%, 91.73%)
Réseau bayésien naïf avec discrétisation		
91.20%	91.48% (**92.10%**)	91.45% (**91.75%**, 91.55%)

- Les deux techniques présentées dans cet article sont compétitives avec la stratégie gagnante avec la compétition KDD [17]. Les straégies gagnantes utilisent une variante des arbres de décision basée sur la notion de "bagging" et "boosting" [18], [15]. Même si nous n'avons pas utilisé ces deux notions dans nos expérimentaions, il existe des situations où les réseaux bayésiens et les arbres de décision donnent d'aussi bons résulats, voire légèrement meilleurs que les stratégies gagnantes.

  En effet, il existe des stratégies où les arbres de décision donnent de meilleurs résultats pour les classes normales, DOS et R2L, et il existe des stratégies où les réseaux bayésiens donnent de meilleurs résulats pour les classes U2R et Probing. Ces résulats intéressants (sur des PCC locaux relatifs à chacune des classes) sont obtenus à l'utilisation des différentes stratégies de regroupement (avant et aprés classification) et grâce à la discrétisation de certaines variables continues.

  Un travail futur serait d'exploiter les complémentarités entre les réseaux bayésiens naïfs et arbres de décision pour le développement d'un méta-classificateur.

- Dans la Table 10, il est clair que les résulats donnés par les arbres de décision sont légèrement meilleurs que ceux obtenus avec les réseaux bayésiens. Cependant de point de vue complexité calculatoire, la construction d'un réseau bayésien (qui se fait en un temps polynomial) est largement moins coûteuse que la construction d'un arbre de décision, où la recherche d'un arbre de décision minimal est NP-complet.

  Cette différence de complexité s'est vue lors des expériementations, puisqu'avec une machine Pentium III, 700Mhz, il est impossible de construire l'arbre de décision (avec une base d'apprentissage contenant 494019 instances) alors qu'avec les réseaux bayésiens la construction ne prend que quelques minutes. Dans nos expérimentations, l'apprentissage et la classification en utilisant les réseaux bayésiens naïfs sont généralement 7 fois plus rapides que ceux effectués en utilisant les arbres de décision.

Remerciement

Ce travail entre dans le cadre d'un projet national Français RNTL (Réseau National des Technologies Logicielles), DICO (Détection d'Intrusions COopérative).

References

[1] Axelsson, S.: Intrusion detection systems: a survey and taxonomy. Technical report 99-15, March 2000.

[2] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and regression trees. Monterey, CA Wadsworth & Brooks, 1984.

[3] Cooper, G. F.: Computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence, Vol. 42, 393-405, 1990.

[4] Duda, R. O. , Hart, P. E., Stork, D. G.: Pattern Classification. Hardcover, 2000.

[5] Jensen, F. V.: Introduction to Bayesien networks. UCL Press, 1996.

[6] Ilgun, K., Kemmerer., R. A., Porras, P. A.: State transition: A rule-based intrusion detection approach. IEEE Transactions on Software Engineering, 21(3), 181-199, 1995.

[7] Lunt, T., Tamaru, A., Gilham, F., Jagannathan, R., Neumann, P., Javitz, H., Valdes, A., Gravey, T.: A real-time intrusion detection expert system (IDES). Final technical report. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, 1992.

[8] Lunt, T.: Detecting intruders in computer systems. In proceedings of the Sixth Annual Symposium and Technical Displays on Physical and Electronic Security, 1993.

[9] Mingers, J.: An empirical comparaison of pruning methods for decision tree induction. Machine Learning, Vol. 4, 227-243, 1989.

[10] Friedman, N., Goldszmidt, M.: Building classifiers using Bayesian networks. In proceedings of American Association for Artificial Intelligence Conference (AAAI'96), 1996.

[11] Pearl J.: Probabilistic Reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmman , Los Altos, CA, 1988.

[12] Portier, P., Froment-Curtil J.: Data mining techniques for intrusion detection. Technical report, University of Texas at Austin, Spring, 2000.

[13] Quinlan, J. R.: Induction of decision trees. Machine Learning 1, 1-106, 1986.

[14] Quinlan, J. R.: C4.5, Programs for machine learning. Morgan Kaufmann San Mateo Ca, 1993.

[15] Quinlan, J. R.: Bagging, boosting, and C4.5. Proceedings of the thirteenth national conference on artificial intelligence and the eighth innovative applications of artificial intelligence conference, volume one, pp 725-730, 1997.

[16] Valdes, A., Skinner K.: Adaptive Model-based Monitoring for Cyber Attack Detection. In proceedings of Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France, 80-92, 2000.

[17] http://kdd.ccs.uci.edu/databases/kddcup99/task.html

[18] Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36:105-139, 1999.