1. CURRICULUM VITAE

KARIMA SEDKI

DATE DE NAISSANCE : 04 Juin 1979
LIEU DE NAISSANCE : Tizi Ouzou (Algérie)
NATIONALITE : Algérienne
ÉTAT CIVIL : Célibataire

1.1. COORDONNEES

ADRESSE PROFESSIONNELLE :	CRIL CNRS UMR 8188, Faculté des Sciences Jean Perrin Rue Jean Souvraz SP 18 62307 Lens Cedex, France
ADRESSE PERSONNELLE :	15, Rue de Turenne 62300 Lens Cedex, France
TELEPHONE & FAX :	Mobile : 06 22 47 27 89 Bureau : 03 21 79 80 71 Fax : 03 21 79 17 10
ADRESSE ELECTRONIQUE :	sedki@cril.univ-artois.fr
PAGE WEB :	http://www.cril.univ-artois.fr/~sedki

1.2. EXPERIENCES PROFESSIONNELLES

SEP. 2008 :	Attachée Temporaire d'Enseignement et de Recherche à l'IUT de LENS (Université d'Artois).
OCT. 2005 – DEC. 2008 :	Doctorante au laboratoire CRIL (Centre de Recherche Informatique de Lens), Université d'Artois.
SEP. 2006 – JUIN. 2008 :	Vacataire d'enseignement au département Informatique de l'IUT de Lens, Vacataire d'enseignement au département Informatique, faculté Jean Perrin, Université d'Artois.
FEV. 2005 – JUI. 2005 :	Stage de Master II recherche au sein du CRIL, Université d'Artois.

1.3. FORMATIONS ET DIPLOMES

2008 : Doctorat en Informatique d'Université d'Artois

Intitulé de la thèse :	Raisonnement sous incertitude et en présence de préférences : Application à la détection d'intrusions et à la corrélation d'alertes.
Directeur de thèse :	Salem BENFERHAT, Professeur des Universités à l'Université d'Artois
Date de soutenance :	04/12/2008
Mention :	Très honorable[1]

COMPOSITION DU JURY :

Philippe LERAY	Professeur des universités à Polytech'Nantes (rapporteur)
Stéphane LOISEAU	Professeur des universités à l'Université d'Angers (rapporteur)
Salem BENFERHAT	Professeur des universités à l'Université d'Artois, CRIL-LENS (directeur)
Gilles GONCALVES	Professeur des universités à l'Université d'Artois, LGI2A-Béthune (président du jury)
Benjamin MORIN	Ingénieur de recherche (examinateur)
Lakhdar SAIS	Professeur des universités à l'Université d'Artois, CRIL-LENS (examinateur)

2005 : Master II recherche "Systèmes Intelligents et Applications (SIA)" à l'Université d'Artois.
Intitulé : Autour de la négation dans la logique du choix qualitatif (QCL).
Responsables de stage : Salem BENFERHAT, Daniel LE BERRE.
Dates : Février 2005 - juillet 2005.

2004 : Formation en maintenance et montage des ordinateurs (Algérie).

2003 : Ingénieur d'état en Informatique, option systèmes parallèles & distribués, obtenu en décembre 2003 à l'Université Mouloud MAMERI de Tizi-ouzou (Algérie).
Intitulé : Configuration d'un réseau local sans fil par les algorithmes génétiques
Encadrement : Malika BELKADI (Université de Tizi-Ouzou)
Dates : Janvier 2003 - Décembre 2003

1998 : Baccalauréat Série Sciences Exactes (Mathématiques), obtenu en juin 1998.

1.4. CONNAISSANCES TECHNIQUES ET INFORMATIQUES

- Détection d'intrusions: TCPDump, Snort, Ethereal/Wireshark, Bro.
- Techniques d'apprentissage automatique/Classification : Weka
- Systèmes d'exploitation : Unix, Linux, Windows.
- Programmation : Turbo Pascal, Delphi, C++ builder, C/ C++, Java,
- Bases de données : SQL, MYSQL, ACCESS.
- Bureautique : Open Office, MS Word, Excel, Power Point, latex.
- Outils Web : Langages html/xhtml.

1.5. LANGUES

- Anglais : Lu, écrit et parlé.
- Français : Lu, écrit et parlé.
- Arabe : Lu, écrit et parlé.
- Kabyle : Langue maternelle.

[1] Le CRIL ne délivre pas les félicitations du jury.

2. ACTIVITES D'ENSEIGNEMENT

Cette section a pour but de présenter les différents enseignements que j'ai effectués à la faculté des sciences de Jean Perrin ainsi qu'à l'IUT de Lens de l'Université d'Artois depuis 2006.

- ➔ De Sep. 2006 à Sep. 2008, je suis intervenue en tant que vacataire au département informatique de l'IUT de Lens ainsi qu'à la faculté de Jean Perrin de l'Université d'Artois.
- ➔ Depuis septembre 2008, je suis Attachée Temporaire d'Enseignement et de Recherche à temps complet à l'IUT de Lens (Université d'Artois). J'interviens dans trois départements : informatique, SRC (Services et Réseaux de Communications) et GEA (Gestion et Administration des Entreprises).
- ➔ Mon expérience en enseignement est de 326 heures équivalents travaux dirigés (TD).

2.1. RECAPITULATIF DES ACTIVITES D'ENSEIGNEMENT

Année	Niveau	Intitulé du module	Cours	TD	TP
2006 et 2007	Licence Mathématique/Informatique	Documents numériques - Open Office, Tableurs, Impress - MS-DOS			45 h
2006	Licence Mathématique/Informatique	Initiation au Web			30 h
2006 et 2008	DUT Informatique	Bases de données			30 h
2006 et 2008	DUT Informatique	Réseaux - Configuration IP - IP et routage, DNS - Configuration du serveur Apache - Client/serveur - Commandes de base : hostname, telnet, ftp, etc.		6 h	16 h
2007 et 2009	DUT Informatique	ACSI (Analyse et Conception des Systèmes d'informations) - Dépendances fonctionnelles - MCD (Modèle conceptuel des données) - MLD (Modèle Logique de Données) - MCT (Modèle Conceptuel de Traitements) - MOT (Modèle Organisationnel de Traitements)		72 h	
2008	DUT Informatique	Architecture des ordinateurs - Codage des entiers et des réels - Circuits logiques, Bascules RS /T/D		18h	
2008	DUT SRC (Service et Réseaux de Communication)	Systèmes d'exploitation		6h	12 h (8h eq. TD)
2009	DUT Informatique	Structures de données avec Java			24 h
2009	DUT GEA (Gestion des Entreprises et Administration)	Outils de gestion et Informatique	96 h (96h eq. TD)		
Total			**96 h**	**102 h**	**154 h**

2.2. DESCRIPTION DETAILLEE DES ENSEIGNEMENT

2006/2007 – 2007/2008 : Enseignements effectués en tant que vacataire à l'IUT de Lens, Université d'Artois.

- ✓ **Bases de données**
  - **Niveau** : DUT informatique
  - **Responsable du cours :** Souhila Kaci
  - **Organisation :** cours, TD, TP
  - **Durée :** 18 heures
  - **Charge :** TP
  - **Contenu :** cet enseignement a pour objectif de fournir et de mettre en pratique les connaissances théoriques concernant les bases de données relationnelles : conception et création des bases de données relationnelles, interrogation des bases de données, maîtrise du langage SQL et le système de gestion de bases de données *ACCESS*.
  -

- ✓ **ACSI (Analyse et Conception des Systèmes d'informations)**
  - **Niveau :** DUT informatique
  - **Responsable du cours :** Frédéric Boussemart
  - **Organisation :** cours, TD
  - **Durée :** 48 heures
  - **Charge :** TD
  - **Contenu :** le but de cet enseignement est de fournir des connaissances approfondies de la conception et l'analyse des systèmes d'informations à savoir le dictionnaire de données, les dépendances fonctionnelles (directes et indirectes) des données, modélisation entité/association, modèle conceptuel de traitement, diagramme de flux, modèle organisationnel de traitement, et modèle organisationnel de données. Ce module contient également une partie sur les bases de données MYSQL. Un projet regroupant tous les concepts vus a été donné aux étudiants. Des mises en pratique de ce projet ont été réalisées en séance de TP sous le logiciel Windesign. J'ai participé au déroulement du projet et à l'évaluation des étudiants en corrigeant leurs projets et une partie de partiel.

- ✓ **Réseaux d'ordinateurs**
  - **Niveau :** DUT informatique
  - **Responsable du cours :** Olivier Roussel
  - **Organisation :** cours, TD, TP
  - **Durée :** 16 heures
  - **Charge :** TP
  - **Contenu :** le but de cet enseignement est de permettre aux étudiants de découvrir comment configurer le réseau sur une machine Unix, manipuler des commandes réseaux de base telles que ifconfig, ping, hostname, ftp, route, etc. Programmer quelques algorithmes (langage C++) sur les adresses IP, routage, simulation de la mise en place de plusieurs machines sur différents réseaux afin d'observer comment transitent les informations d'une machine à l'autre. Le module contient également des exercices sur l'interrogation du DNS, la configuration d'un serveur DNS simple, Client/serveur de temps TCP.

2006/2007 – 2007/2008 : Enseignements effectués en tant que vacataire à la faculté des sciences Jean Perrin, Université d'Artois.

- ✓ **Documents numériques**
  - **Niveau :** Licence Mathématique/Informatique
  - **Responsable du cours :** Nathalie Sperando-Chetcuti en 2006 et Daniel Le Berre en 2007
  - **Organisation :** cours, TP
  - **Durée :** 45 heures
  - **Charge :** TP
  - **Contenu :** cet enseignement permet aux étudiants de manipuler les outils concernant le traitement de texte sous Open Office (styles, publipostage, …), les tableurs (formules mathématiques, insertion de graphes, …), l'utilisation de MS-DOS, accès à l'environnement numérique de travail. J'ai participé à la rédaction d'un énoncé de TP et des sujets d'examens. J'ai également corrigé des contrôles TP et des examens.

- ✓ **Initiation au Web**
  - **Niveau :** Licence Mathématique/Informatique
  - **Responsable du cours :** Sylvain Lagrue
  - **Organisation :** cours, TP
  - **Durée :** 30 heures
  - **Charge :** TP
  - **Contenu :** cet enseignement permet aux étudiants d'apprendre les éléments de base concernant la création des pages Web en utilisant le langage HTML et les feuilles de styles. J'ai participé à la rédaction d'un sujet de TP et d'un sujet d'examen. J'ai également corrigé des contrôles TP et des examens.

2008 – 2009 : Enseignements effectués durant mon contrat d'Attachée Temporaire d'Enseignement et de Recherche à l'IUT de Lens, Université d'Artois**.**

- ✓ **Architecture des ordinateurs**
  - **Niveau :** DUT informatique
  - **Responsable du cours :** Vincent Vidal
  - **Organisation :** cours, TD
  - **Durée :** 24 heures (18 h TD d'architecture et 6 h TP réseaux)
  - **Charge :** TD
  - **Contenu :** dans le cadre de ce TD, les étudiants ont vu des notions de base sur l'architecture des ordinateurs en général, le codage des nombres positifs et négatifs en complément à 2/signe module, BCD, code ASCII, codage en simple et double précision, les tableaux de Karnaugh, les circuits logiques combinatoires et séquentiels. Cet enseignement contient aussi une partie (4 séances de TP) qui groupe plusieurs commandes de base nécessaires pour travailler en réseau : ftp, telnet, scp, etc. Le responsable de cette deuxième partie est : Sylvie Coste-Marquis. J'ai participé à la correction des examens.

- ✓ **Systèmes d'exploitation**
  - **Niveau :** DUT SRC (Service et Réseaux de Communication)
  - **Responsable du cours :** Sébastien Tabary
  - **Organisation :** cours, TD, TP
  - **Durée :** 12 heures TP et 6 heures TD (14 h équivalent TD)
  - **Charge :** TP/TD
  - **Contenu :** le but de cet enseignement est de permettre aux étudiants d'apprendre les concepts de base du fonctionnement du système Linux comme la création, suppression, accès aux répertoires,

droits d'accès des utilisateurs, écriture de scripts shell. J'ai participé à la correction des examens.

- ✓ **Bases de données**
  - **Niveau :** DUT informatique
  - **Responsable du cours :** Souhila Kaci
  - **Organisation :** cours, TD, TP
  - **Durée :** 12 heures
  - **Charge :** TP
  - **Contenu :** cet enseignement a pour objectif de fournir et de mettre en pratique les connaissances théoriques concernant les bases de données relationnelles : conception et création des bases de données relationnelles, maîtrise du langage SQL et ACCESS.

- ✓ **Structures de données avec Java**
  - **Niveau :** DUT informatique
  - **Responsable du cours :** Assef Chmeiss
  - **Organisation :** cours, TD, TP
  - **Durée :** 24 heures
  - **Charge :** TP
  - **Contenu :** Ce module a pour objectif de permettre aux étudiants d'apprendre l'essentiel de la programmation orienté objet Java. Pendant les séances de TP, les étudiants programment les exercices vus en séances de TD et de nouveaux exercices afin d'apprendre correctement les règles de la programmation. Les programmes concernent : les classes, les fonctions, les procédures, les tableaux, les matrices, les algorithmes de tri, les piles, les files, les listes, etc.

- ✓ **ACSI (Analyse et Conception des Systèmes d'informations)**
  - **Niveau :** DUT informatique
  - **Responsable du cours :** Hachémi Bennaceur
  - **Organisation :** cours, TD,
  - **Durée :** 24 heures
  - **Charge :** TD
  - **Contenu :** Ce module permet aux étudiants de pouvoir définir l'organisation d'un système d'information en matière de données et des traitements effectués. Les étudiants réalisent des études de cas afin d'établir des diagramme de flux, modèle organisationnel de traitement, et modèle organisationnel de données.

- ✓ **Outils informatiques et de gestion**
  - **Niveau :** DUT GEA (Gestion des Entreprises et Administration),
  - **Responsable du cours/TP :** Moi-même
  - **Organisation et charge :** TP/cours
  - **Durée :** 96 heures
  - **Contenu :** Ce cours est organisé en deux parties : cours et TP. L'objectif est de permettre aux étudiants d'apprendre les différents outils Informatiques, notamment pour le traitement de texte avec Microsoft Word (Styles, publipostage, réalisation de CV, macro, formulaires, etc.) et la gestion des bases de données (SQL, ACCESS).

de prendre rapidement les responsabilités d'enseignements.

3. ACTIVITES DE RECHERCHE

Cette section a pour objectif de présenter une description synthétique de mon expérience dans la recherche en Informatique. Ma première expérience a commencé lors du stage de Master II Recherche en 2005 au Centre de Recherche Informatique de Lens (CRIL) de l'Université d'Artois. Cette expérience s'est renforcée dans le cadre de ma thèse qui s'est déroulée d'octobre 2005 jusqu'à décembre 2008. Durant les trois années de ma thèse, j'ai participé à deux projets nationaux : ACI sécurité informatique DADDi (Dependable Anomaly Detection with Diagnocis) et PLACID (Probabilistic graphical models and Logics for Alarm correlation in Intrusion Detection).

3.1. RESUME DES TRAVAUX DE RECHERCHE

Les principaux axes de recherches de ma thèse se focalisent sur le développement des formalismes pour la représentation des préférences, raisonnement sous incertitude dans le cadre des modèles graphiques probabilistes et application à la sécurité informatique (détection d'intrusions réseaux et corrélation d'alertes). Les principaux thèmes de nos travaux de recherche sont :

- Représentation des préférences
- Techniques d'apprentissage automatique et classification
- Réseaux Bayésiens /Arbres de décisions
- Structuration de données et sélection d'attributs
- Détection d'intrusions (en différé/temps réel)
- Corrélation d'alertes

Nos principales contributions sont résumées dans les sections suivantes.

3.1.1. RAISONNEMENT EN PRESENCE DE PREFERENCES

Le raisonnement en présence de préférences est un problème important car face aux problèmes quotidiens, on se retrouve souvent confronté à des situations de prise de décision. Pour prendre des décisions, il est question d'agir en fonction de nos choix et nos préférences sur un ensemble d'alternatives ou de propositions. Or, pour satisfaire nos préférences, plusieurs problèmes se posent, on trouve entre autres :

- ✓ Le nombre d'alternatives est exponentiel car ces dernières sont souvent accompagnées d'un ensemble de critères et de contraintes
- ✓ Les agents expriment souvent leurs préférences de façon contradictoire
- ✓ Les préférences évoluent en fonction du temps et changent en fonction du nombre d'alternatives
- ✓ Il existe différents types de préférences telles que les préférences prioritaires et positives.

Pour représenter les préférences, deux principales approches sont utilisées dans la littérature : *approches qualitatives ou logiques* ([16], [15]) et *approches quantitatives ou numériques* ([14], [13]). Dans le cadre de notre travail concernant le raisonnement en présence de préférences, nous nous sommes intéressés à une approche logique appelée *"Logique du Choix Qualitatif (QCL)"*. La logique QCL [17] est une nouvelle logique de représentation des préférences qui ajoute à la logique propositionnelle un nouveau connecteur, nommé *disjonction ordonnée,* qui permet d'ordonner les différentes alternatives sur lesquelles les agents définissent leurs préférences.

Cette logique est intéressante car elle donne la possibilité d'exprimer des préférences simples de la forme "A est préféré à B" ou complexes de la forme "Si A est préféré à B alors C est préféré à D". Les préférences simples sont représentées par des formules de choix de base (BCF) et les préférences complexes sont représentées par des formules de choix générales (GCF).
Après une étude détaillée de cette logique, nous avons constaté qu'elle présente plusieurs limites, ce qui restreint son champ d'application. Ainsi, nous nous somme intéressés à :

- Cerner les limites de cette logique.
- Définir un cadre qui nous permet de remédier aux limites de cette logique.
- Modifier le langage de la logique *QCL* afin de pouvoir représenter d'autres types de préférences telles que les préférences prioritaires et positives.
- Étendre le langage *QCL* dans le cadre de la logique du premier ordre afin de pouvoir modéliser le problème de la corrélation d'alertes.

CERNER LES LIMITES DE LA LOGIQUE QCL

Les limites de la logique QCL concernent plus précisément des cas du raisonnement conditionnel sur les préférences et de la négation d'un ensemble de préférences. La relation d'inférence QCL n'est pas totalement satisfaisante. En effet, les préférences exprimées par des règles conditionnelles de la forme *"Si A est préféré à B alors C "* sont équivalentes aux préférences de la forme *"Si A ou B alors C "*. Ce problème est dû à la définition de la négation, qui ignore la notion de préférence, c'est-à-dire que la disjonction ordonnée est remplacée par la disjonction standard. A titre d'exemple, la négation d'une préférence de la forme *"il n'est pas vrai que je préfère aller au cinéma à ne pas y aller"* est équivalente à une théorie contradictoire.

Cela pose une vraie limite à la logique QCL, alors que la négation a une importance capitale pour exprimer les préférences par des rejets (présence de la négation dans les préférences) et des règles conditionnelles. Les conséquences de cette limite s'élargissent à quelques propriétés logiques importantes telles que la double négation d'une formule QCL qui n'est pas équivalente à la même formule, les lois de De Morgan qui ne sont pas vérifiées, présence d'incohérence dans une base de préférences exprimées dans le cadre de QCL, notamment en présence des préférences conditionnelles, etc. Après avoir situé et cerné les limites de la logique QCL, nos principales contributions concernant ce problème sont :

REVISIONS ET MODIFICATIONS DE LA LOGIQUE QCL :

Comme les limites de la logique QCL se présentent principalement au niveau de la négation dans les préférences, nous avons proposé dans un premier temps une nouvelle définition de la négation car cette dernière, telle qu'elle a été définie dans QCL, ignore la notion de préférence. Cette première modification donne ainsi une nouvelle logique que nous avons appelée "Logique du Choix Qualitatif Minimale (*MQCL*)", qui se caractérise par une nouvelle définition de la négation, mais garde les définitions de conjonction et de disjonction des préférences de la logique *QCL*. La logique *MQCL* est intéressante, car elle propose une nouvelle négation qui permet de surmonter les limites de *QCL*. En effet, la nouvelle négation que nous avons définie prend en compte la notion de préférence, c'est-à-dire que les rejets peuvent être ordonnés. Par ailleurs, se limiter uniquement à modifier la définition de la négation n'est pas intéressant du fait, que la nouvelle logique ne vérifie pas les propriétés de De Morgan comme c'est le cas pour *QCL* et ne permet pas de représenter d'autres types de préférences telles que les préférences prioritaires et positives. C'est pourquoi, nous avons proposé d'apporter d'autres modifications à la logique *QCL* au niveau de la conjonction et de la disjonction des préférences. Ces modifications donnent deux autres logiques : la première est appelée *Logique du Choix Qualitatif Prioritaire*, notée par *PQCL*, et la seconde est appelée *Logique du Choix Qualitatif Positive*, notée par *QCL+*. Ces deux logiques se caractérisent par une nouvelle définition de la négation qui est, également, la même que celle définie dans *MQCL*. Nous nous sommes intéressés à développer un cadre permettant d'exprimer les préférences prioritaires et positives car lorsque les agents effectuent leurs choix sur un ensemble d'alternatives, ils se basent généralement sur ce qu'ils souhaitent fortement, partiellement ou faiblement, ce qu'ils aimeraient et ce qu'ils ne souhaitent pas ou n'aiment pas. Cela signifie que différents critères interviennent dans leurs choix tels que les priorités, les buts, les rejets, etc.

La logique PQCL est adaptée à la représentation des préférences prioritaires, c'est-à-dire des préférences ayant différents niveaux de priorité. Notre choix pour ce type de logique est motivé par le fait que les préférences des agents sont certainement différentes au niveau de l'importance qu'elles procurent. Pour un problème donné, les agents expriment souvent plusieurs préférences sur un grand nombre d'alternatives afin de pouvoir satisfaire au minimum une des préférences souhaitées même si toutes les préférences exprimées ne sont pas totalement souhaitable. Prenons un exemple où un agent doit acheter en urgence un billet pour un voyage. Supposons que ses préférences sont exprimées comme suit :

- Il préfère voyager en avion qu'en train
- Il préfère voyager le soir plutôt que le matin
- Il préfère voyager en première classe plutôt qu'en seconde classe
- Il préfère un voyage sans escale
- Il préfère le voyage le moins cher

Il est clair que même si toutes ces préférences exprimées ne peuvent pas être satisfaites, l'agent préfère quand même acheter le billet disponible vu l'urgence de la situation. Ainsi, cette logique permet de donner des priorités aux préférences exprimées. Pour ces préférences, il est par exemple plus intéressant de considérer la préférence "Il préfère voyager en avion qu'en train" prioritaire à la préférence "Il préfère voyager le soir plutôt que le matin". C'est-à-dire que s'il y a une place disponible dans un avion ou dans un train, l'agent peut faire abstraction de la contrainte du temps (soir ou matin).

La logique QCL+ est particulièrement adaptée à la représentation des préférences positives, c'est-à-dire aux préférences qui permettent de représenter ce qui est souhaitable par l'agent et n'excluent jamais de solutions.

Chacune des logiques proposées se caractérise par une sémantique bien particulière, en ce qui concerne le type des préférences qu'elle permet de représenter, mais les trois, partagent plusieurs propriétés comme la définition de la négation des préférences par exemple. Lorsque nous appliquons la définition de la négation que nous avons proposée dans le cadre de nos logiques, nous n'obtenons jamais une contradiction, et nous présentons la négation de la préférence de la forme *"il n'est pas vrai que je préfère aller au cinéma à ne pas y aller"* simplement par "je ne préfère pas aller au cinéma plutôt que l'inverse ».

Chacune de nos logiques est caractérisée aussi par plusieurs définitions[2] telles que la relation d'inférence des formules de choix de base et des formules de choix générales qui consiste à définir le degré de satisfaction de chaque formule étant donnée une interprétation (par exemple, la logique PQCL offre deux possibilités pour définir la relation d'inférence des formules de choix générales; la première méthode est directe c'est-à-dire qu'elle permet de définir directement le degré de satisfaction, quant à la seconde, elle propose d'abord de normaliser les préférences complexes c'est-à-dire les transformer en préférences simples représentées par des formules de choix de base), la notion d'optionalité qui indique le nombre d'options possibles pour satisfaire une formule, la définition des modèles préférés qui consiste à définir un modèle préféré d'un ensemble de connaissances représentées par des formules propositionnelles et d'un ensemble de préférences représentées par des formules de choix de base ou par des formules de choix générales. Un modèle préféré dans le cadre de la logique MQCL par exemple doit satisfaire toutes les connaissances et préférences avec un certain degré.

Nos logiques offrent aussi la possibilité de normalisation des préférences à l'aide de fonctions de normalisation qui permettent de transformer des préférences complexes représentées par des formules de choix générales en préférences simples représentées par des formules de choix de base. La normalisation des préférences est très importante du fait qu'elle permet de transformer des préférences dans d'autres formats.

Notons aussi qu'une nouvelle définition possibiliste des préférences et des relations entre le cadre de nos logiques et la logique possibiliste ont été aussi présentées.

[2] Chaque logique MQCL (PQCL, QCL+) propose des définitions propres aux types de préférences qu'elle présente.

Nous avons également proposé l'extension du langage des logiques développées en un fragment de la logique du premier ordre afin de pouvoir exprimer des informations générales et l'appliquer au problème de la corrélation que nous décrivons plus tard.

3.1.2. TECHNIQUES D'APPRENTISSAGE AUTOMATIQUE/CLASSIFICATION POUR LA DETECTION D'INTRUSIONS

La détection d'intrusions consiste à analyser à base des techniques mises au point au sein d'un système tout évènement circulant sur le réseau dans le but de reconnaître toute activité suspecte et toute violation de la politique de sécurité [18]. Deux principales approches de détection sont utilisées par les systèmes de détection d'intrusions existants : l'approche comportementale [19] et l'approche par signatures [20]. L'approche comportementale se base sur la définition d'un profile qui modélise les différentes activités normales d'un système. Toute déviation du comportement normal sera interprétée comme une éventuelle intrusion. L'approche par signatures consiste à détecter les attaques en recherchant leurs signatures (accès à un fichier système par exemple) dans les données analysées. Toute attaque qui n'a pas sa signature dans la base de signatures ne sera pas détectée. Ces approches sont indispensables pour la sécurité des systèmes d'informations. Cependant plusieurs problèmes demeurent sans solutions satisfaisantes tels que par exemple uniquement les attaques connues dans la base de signatures seront détectées dans le cas d'une approche par signatures et la difficulté d'analyse due au nombre important d'alertes générées pour tout changement dans les comportements habituels des utilisateurs dans le cas d'une approche comportementale.

Dans le cadre du problème de la détection d'intrusions, les principaux problèmes auxquels nous nous intéressons sont les suivants :

- Toutes les approches analysant les connexions ne détectent pas les intrusions en temps réel car on attend la fin d'une connexion afin d'extraire ses attributs pour l'analyser. Cela peut être très dommageable car le temps que la connexion se termine, l'attaquant aura accompli son forfait.
- Etant donné un trafic réseau brut, l'analyse n'est pas une tâche facile, vue la nature des données qui est brute et la quantité de données qui est très volumineuse.
- Le volume de fausses alertes générées est très important, notamment dans le cas d'une approche comportementale lorsque, par exemple, des utilisateurs autorisés changent leurs comportements habituels au sein du système.

Afin de pouvoir détecter des activités malveillantes dans un réseau, il est important de définir les informations nécessaires à analyser. Or, cette tâche n'est pas facile, car plusieurs éléments peuvent intervenir dans la réalisation des attaques : types des protocoles et services, moments et durées des attaques, nature des ressources et applications exploitées, types et nombre d'actions exécutées, etc. En fait, la difficulté majeure d'un problème de détection d'intrusions est principalement liée à la complexité des données à analyser, car les données réseaux sont de nature brute, souvent hétérogènes (différentes sources) et disponibles en grandes quantités (plusieurs giga-octets). Comme les attaques sont réalisées au niveau des paquets, il n'est donc pas possible de pouvoir les analyser telles qu'elles sont. Ainsi, nous considérons que le problème de la détection d'intrusions est, avant tout, un problème de description des données réseaux qui nécessitent deux éléments importants : la définition et l'extraction des informations (caractéristiques d'activités normales et attaques). Nos principales contributions concernant ce problème sont les suivantes :

DEFINITION D'UN JEU D'ATTRIBUTS

Pour répondre à nos objectifs, nous nous sommes intéressés dans un premier temps à la modélisation du problème de la détection d'intrusions par un modèle graphique probabiliste dans le but de détecter les

attaques réseaux en temps réel, c'est-à-dire avec des connexions non finies. Pour modéliser le problème de détection d'intrusions à l'aide d'un modèle graphique probabiliste, il est nécessaire, dans un premier temps, de représenter les données du problème dans un format exploitable par ces modèles. C'est pourquoi, nous avons proposé de définir un jeu d'attributs décrivant les données réseaux qui sont de nature brute : certains attributs sont communs à tous les événements (normaux ou anormaux) comme l'adresse IP source, numéro de port, service, etc. D'autres sont de haut niveau permettant de décrire les actions des attaques, comme le nombre de tentatives d'accès en mode root par exemple, etc. Le jeu d'attributs que nous avons défini contient en tout 44 attributs. L'objectif principal de la définition d'attributs pour un problème de détection d'intrusions consiste à proposer un jeu d'attributs garantissant un taux de détection élevé avec un faible taux de fausses alertes. Cette tâche ressemble à celle de l'écriture de signatures dans le sens où les deux ont pour objectif de décrire les données réseaux, les caractéristiques du système surveillé et toute information servant à reconnaître des attaques. Cependant, l'écriture de signatures s'avère une tâche plus délicate car chaque signature doit concerner une attaque particulière, ce qui nécessite des connaissances expertes sur les comportements et les stratégies suivies par des attaques.

Par exemple, étant donnée l'alerte suivante concernant une attaque donnée :

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg :"FTP .rhosts") ;
flow :to_server,established ; content :"*:rhosts*" ; metadata :service ftp ;
reference :archnids,328 ; classtype :suspecious-filename-detect ; sid :335,rev :6 ;)
```

Si les techniques de détection ne disposent pas d'informations sur cette attaque (signature), alors elle ne sera pas détectée, c'est pourquoi, nous avons défini l'attribut hot par exemple qui prend la valeur 1 lorsque la chaîne de caractère « *.rhosts »* est trouvée*.*

STRUCTURATION ET FORMATAGE DES DONNEES RESEAU BRUTES

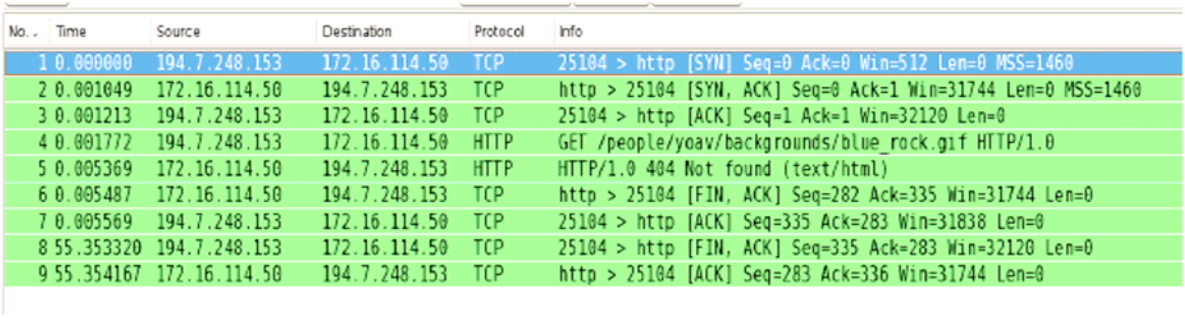
Les données réseaux sont de nature brute et leurs quantités sont énormes. Pour pouvoir les modéliser et les analyser avec des modèles graphiques probabilistes par exemple, il est impératif de les structurer et les décrire avec des attributs pertinents et discriminants. Ainsi, nous avons développé un outil de formatage en ajoutant de nouvelles fonctionnalités de formatage à l'analyseur de trafic réseau Ethereal [25]. Notre outil est réalisé conjointement avec K. Tabia [4] dans le cadre de l'ACI sécurité informatique DADDi.

Fonctionnalités de notre outil

L'objectif de notre outil de formatage est de nous permettre de structurer et transformer le trafic réseau (collecté en temps réel par un sniffeur ou préalablement sauvegardé dans des fichiers) dans un format compréhensible et exploitable. Il permet de construire des connexions à partir des paquets constituant ces connexions. Chaque connexion est caractérisée par les d'attributs que nous avons définis.

- **Construction des connexions :** Selon les protocoles utilisés, trois types de connexions sont construites : TCP, UDP et ICMP. Notons que nous avons utilisé le terme connexion pour les protocoles connectés (utilisant le protocole TCP) mais aussi pour les protocoles non connectés (utilisant soit UDP soit ICMP).

  Exemple :



No. .	Time	Source	Destination	Protocol	Info
1	0.000000	194.7.248.153	172.16.114.50	TCP	25104 > http [SYN] Seq=0 Ack=0 Win=512 Len=0 MSS=1460
2	0.001049	172.16.114.50	194.7.248.153	TCP	http > 25104 [SYN, ACK] Seq=0 Ack=1 Win=31744 Len=0 MSS=1460
3	0.001213	194.7.248.153	172.16.114.50	TCP	25104 > http [ACK] Seq=1 Ack=1 Win=32120 Len=0
4	0.001772	194.7.248.153	172.16.114.50	HTTP	GET /people/yoav/backgrounds/blue_rock.gif HTTP/1.0
5	0.005369	172.16.114.50	194.7.248.153	HTTP	HTTP/1.0 404 Not found (text/html)
6	0.005487	172.16.114.50	194.7.248.153	TCP	http > 25104 [FIN, ACK] Seq=282 Ack=335 Win=31744 Len=0
7	0.005569	194.7.248.153	172.16.114.50	TCP	25104 > http [ACK] Seq=335 Ack=283 Win=31838 Len=0
8	55.353320	194.7.248.153	172.16.114.50	TCP	25104 > http [FIN, ACK] Seq=335 Ack=283 Win=32120 Len=0
9	55.354167	172.16.114.50	194.7.248.153	TCP	http > 25104 [ACK] Seq=283 Ack=336 Win=31744 Len=0

Fig. 1 : Exemple de paquets constituant une connexion TCP

Les neuf paquets de la figure 1 constituent une seule connexion TCP. Comme nous l'observons sur les informations affichées dans les paquets, le service utilisé par cette connexion est http, l'établissement de la connexion est fait par l'envoi de trois drapeaux : SYN envoyé par la machine source qui initie la connexion suivi par un SYN-ACK envoyé par la machine de destination et se termine par ACK pour accuser la réception de l'établissement de la connexion.

- **Types de formatage :** Notre outil offre deux possibilités de formatage du trafic réseau brut :

1) **Formatage hors ligne** : ce type de formatage consiste à construire des connexions finies (complètes ou terminées) ou non finies (incomplètes ou pas encore terminées) à partir du trafic réseau capturé et enregistré dans des fichiers. Le formatage hors ligne est intéressant et utile dans le cadre de la détection d'intrusions dans la mesure où la construction des connexions permet d'analyser et de retracer tous les événements d'une connexion, ce qui permet une meilleure analyse que lorsque l'on considère des paquets séparément.

2) **Formatage en temps réel :** afin de réaliser une détection en temps réel, il est nécessaire de formater le trafic brut en temps réel également. Comme le formatage hors ligne, le formatage en temps réel permet de construire des connexions finies ou non finies mais à partir du trafic en cours de capture. La fonctionnalité de formatage en temps réel est indispensable pour l'analyse en temps réel, notamment pour prendre des contre-mesures le plus tôt possible.

Dans chaque type de formatage, deux types de connexions peuvent être construites : connexions finies (complètes) et connexions non finies (incomplètes). Une connexion finie est caractérisée par les attributs calculés sur l'ensemble des paquets appartenant à cette connexion. Pour construire des connexions finies, notre outil crée une nouvelle connexion dès la réception de son premier paquet, et la met à jour à chaque fois qu'un nouveau paquet appartenant à cette connexion est reçu.

Notre outil permet de construire à partir de la liste de paquets présentée dans la figure 1 la connexion correspondante et de la décrire par un ensemble d'attributs. A titre d'exemple, nous avons dans les trois premiers paquets des échanges (SYN, SYN-ACK et ACK), cela signifie que la connexion est établie d'une façon normale et dans les quatre derniers paquets, nous avons les flags (FIN-ACK, ACK, FIN-ACK et ACK), ce qui signifie également la terminaison normale de la connexion. Donc, l'état de la connexion est représenté par l'attribut flag qui prend la valeur "SF". A partir des informations dans les paquets, le service utilisé par cette connexion est "http", les adresses IP source et destination sont "194.7.248.153" et "172.16.114.50" , etc. La connexion finie construite est donc telle qu'elle est affichée dans le tableau suivant (nous donnons quelques attributs uniquement) :

Durée	Protocole	Service	Flag	Src-bytes	Dst-bytes	land	Direction	….
55.354	tcp	http	SF	334	281	0	inbound	.....

Exemple d'une connexion TCP finie

Une connexion non finie est une connexion qui concerne seulement un ensemble de paquets à un certain

instant (connexion qui n'a pas encore atteint sa fin). Les attributs qui la décrivent sont les mêmes que ceux d'une connexion finie sauf que certains attributs prennent des valeurs décrivant la connexion à cet instant. Comme exemples d'attributs évoluant dans le temps, on trouve : durée, flag, etc.

DETECTION D'INTRUSIONS HORS LIGNE ET EN TEMPS REEL

Une fois les données formatées, c'est-à-dire qu'elles sont représentées sous forme de connexions décrites par une liste d'attributs, nous appliquons alors une technique d'apprentissage automatique/classification pour construire le classifieur qui va être utilisé pour déterminer la classe d'une nouvelle connexion (normale ou attaque). Les techniques de classification et d'apprentissage automatique permettent d'élaborer sur les données d'apprentissage préalablement étiquetées, des modèles de classification pouvant être généralisés sur l'ensemble des données du domaine. Nous nous sommes intéressés en particulier à la détection d'intrusions avec des connexions non finies pour analyser le trafic réseau en temps réel véritablement. La détection avec des connexions finies est intéressante du fait que ce type de connexions permet de retracer les activités complètes d'une attaque. Quant à la détection avec des connexions non finies, notre objectif est de pouvoir identifier en temps réel un trafic malveillant sur le réseau, c'est-à-dire détecter le plutôt possible les attaques, ce qui exige l'analyse de connexions non finies.

Pour nos expérimentations, nous avons utilisé les réseaux bayésiens[3] [22] qui sont des modèles graphiques probabilistes largement appliqués dans le domaine de la détection d'intrusions. Les données que nous avons utilisées concernent les données DARPA'99 [21] qui sont disponibles publiquement en format brut que nous avons formatées avec notre outil de formatage. Les résultats expérimentaux basés sur les attributs définis sont satisfaisants et confirment leur pertinence. Quant à la détection avec des connexions non finies ou incomplètes (détection en temps réel), les résultats obtenus sont également satisfaisants.

3.1.3. LOGIQUE DE REPRESENTATION DE PREFERENCES POUR LA CORRELATION D'ALERTES

Un autre problème important de la détection d'intrusions que nous avons déjà souligné plus haut, concerne le nombre important d'alertes que les systèmes de détection d'intrusions produisent. La majorité de ces alertes ne correspondent pas réellement à des attaques (fausses alertes, alertes redondantes, etc.). Ainsi, l'administrateur qui a pour tâche d'analyser et de prendre des décisions nécessaires, se retrouve rapidement débordé, et laisse certainement passer des alertes suspectes sans les analyser. Ce problème relève de la corrélation d'alertes [26] qui est une branche de la détection d'intrusions. D'une manière générale, la corrélation d'alertes est définie comme l'interprétation conceptuelle de plusieurs alertes afin de leur attribuer une meilleure sémantique d'une part et de réduire le volume important d'alertes d'autre part. Plusieurs approches de corrélation d'alertes ont été développées dans le but de remédier à ce problème. Cependant, malgré l'efficacité de la majorité de ces approches pour éliminer les informations redondantes ou détecter des attaques coordonnées et complexes; la tâche d'analyse d'alertes reste difficile car le nombre d'alertes reste important. Nos propositions concernant ce problème est une nouvelle approche de corrélation d'alertes. Nous nous sommes basés sur le fait qu'un administrateur réseau soit le mieux placé pour connaître le fonctionnement de son système, le type de failles qui existent, le nombre et le type d'alertes générées, les alertes qui nécessitent l'analyse ou non, etc. Ainsi, toutes ces informations peuvent être représentées par deux éléments :

- Toute information que l'administrateur réseau dispose par expérience, ou qu'il peut avoir sur les alertes, est considérée comme une connaissance ou une contrainte d'intégrité.
- Toute information pouvant aider l'administrateur à écarter ou ignorer certaines alertes et privilégier d'autres est considérée comme une préférence.

Notre intérêt, s'est focalisé sur le raisonnement en présence des préférences. Ainsi, nous avons proposé une

[3] Après la fonction de formatage et d'extraction d'attributs, les données réseaux sont facilement représentables par un modèle graphique. Dans le cas d'un réseau bayésien par exemple, le nœud du graphe correspond à la classe des données réseaux (normale ou anormale) et les nœuds fils correspondent aux attributs qui caractérisent les données (Flag, protocole, Service, Durée, etc.).

nouvelle approche basée sur l'une des logiques que nous avons développées. Il s'agit de la logique du choix qualitatif minimale (MQCL). Comme le langage de cette logique est propositionnel, nous avons proposé dans un premier temps d'étendre ce langage en un fragment de la logique du premier ordre qui est un cadre plus général. Dans le nouveau cadre que nous avons appelé FO-MQCL (First Order – Minimal Qualitative Choice Logic), les connaissances de l'administrateur réseau sont représentées par des formules du premier ordre universellement quantifiées, ses préférences simples sont représentées par des formules de choix de base universellement quantifiées et ses préférences générales ou complexes sont représentées par des formules de choix générales universellement quantifiées. Nous donnons dans ce qui suit, un exemple d'une préférence codée dans le cadre de la logique FO – MQCL :

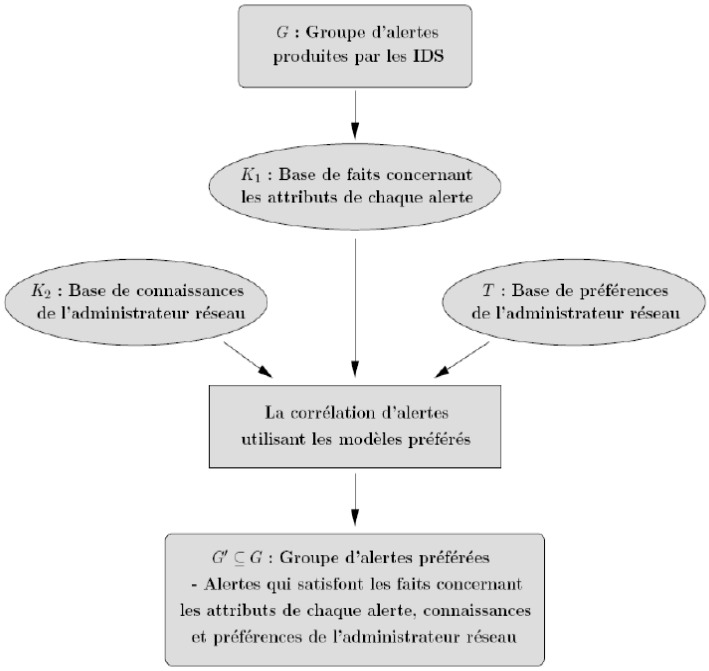
$$\forall x, \forall y \ \text{Direction}(x, \text{inbound}) \wedge \text{Direction}(y, \text{outbound}) \wedge \text{Differ}(x, y) \Rightarrow \text{Present-alert}(x) \ \vec{\times} \ \text{Present-alert}(y).$$

Par cette formule, l'administrateur préfère les alertes entrantes à celles qui sont sortantes. Les alertes sont considérées "entrantes (inbound)" si les adresses IP source des événements qu'elles décrivent ne font pas partie du réseau. Elles sont "sortantes (outbound)" si leurs adresses IP source font partie du réseau, mais pas les adresses IP destination.

Pour définir les alertes préférées qui sont les alertes qui satisfont les connaissances et les préférences de l'administrateur réseau, notre approche se base sur la définition des modèles préférés que nous avons définie dans le cadre de la logique MQCL. Le principe de fonctionnement de notre modèle est le suivant :

- ➔ Représenter toutes les connaissances de l'administrateur réseau dans une base de connaissances,
- ➔ Représenter toutes ses préférences dans une base de préférences,
- ➔ Coder les connaissances et les préférences dans le cadre de la logique FO-MQCL,
- ➔ Extraire une base de faits qui représente des informations sur chaque alerte à analyser (exemples : le protocole, la direction de chaque alerte, service, etc.). Les préférences sur les alertes sont exprimées en considérant l'ensemble des faits qui décrivent ces alertes.
- ➔ Coder les faits de la base de faits dans le cadre de la logique FO-MQCL
- ➔ Appliquer la relation d'inférence de la logique FO-MQCL en tenant compte des trois bases (faits, connaissances et préférences).

- ➔ Appliquer la définition des modèles préférés pour présenter à l'administrateur uniquement des alertes préférées, c'est-à-dire des alertes qui devraient satisfaire les connaissances et les préférences de l'administrateur réseau. Si besoin, des alertes préférées en second degré peuvent être également présentées, etc. Le schéma général de notre approche est le suivant :



Nous avons validé notre approche sur deux bases de données : données DARPA'99 et données DADDi qui concernent un trafic réel généré dans le cadre du projet DADDi. Les résultats obtenus sont très satisfaisants.

3.2. PUBLICATIONS

Revue internationale

[1] Salem Benferhat, Karima Sedki,
Two alternatives for handling preferences in qualitative choice logic.
Fuzzy Sets and Systems Journal *(FSS 2008),* 159(15), pages 1889-1912, august 2008.

Communications dans des conférences d'audience internationale avec publication des actes

[2] Salem BENFERHAT, Daniel LE BERRE, Karma SEDKI,
Alternative Inference for Qualitative Choice Logic
Procedings of 17h European Conference on Artificial Intelligence (*ECAI 2006)*, IOS Press, pages 741-742, Riva del Garda, Italie, 2006.

[3] Salem BENFERHAT, Daniel LE BERRE, Karima SEDKI,
Handling Qualitatif Preferences Using Normal Form Functions
The Florida Artificial Intelligence Research Society conference *(FLAIRS 2007),* pages 38-43, Key West, Florida, mai 2007.

[4] Salem BENFERHAT, Karima SEDKI and Karim TABIA,
Preprocessing rough network data for intrusion detection purposes
Dans International Conference Telecommunications, Networks and Systems *(IADIS 2007),* pages 105-109, Lisbone, Portugal, juillet 2007.

[5] Salem BENFERHAT, Karima SEDKI, Sylvain GOMBAULT,
Towards Selecting Relevant Attributes using Decision Trees for Intrusion Detection
Proceedings of the International Conference on High Performance Computing, Networking and Communication Systems *(HPCNC 2007)*, pages 224-230, Orlando, Florida, juillet 2007.

[6] Salem BENFERHAT, Karima SEDKI,
A Revised Qualitative Choice Logic for Handling Prioritized Preferences
Proceedings of Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty *(ECSQARU 2007)***,** pages 635-647, Hammamet, Tunisie, octobre 2007.

[7] Salem BENFERHAT, Karima SEDKI,
Alert Correlation based on a Logical Handling of Administrator Preferences and Knowledge
Proceedings of International Conference on Security and Cryptography *(SECRYPT 2008)*, pages 50-56, Porto, Portugal, Juillet 2008.

[8] Salem BENFERHAT, Karima SEDKI,
Intrusions Detection *For Incomplete Connexions*
Computer Security Conférence *(CSC 2009), (à apparaître).*

Communication(s) dans des conférences d'audience nationale

[9] Salem BENFERHAT, Daniel LE BERRE, Karima SEDKI,
La logique du choix qualitatif révisée
Rencontres francophones sur la logique floue et ses applications *(LFA 2006),* pages 95-102, Toulouse, France, Octobre 2006.

[10] Salem BENFERHAT, Karima SEDKI,
Corrélation d'alertes basée sur les connaissances et les préférences d'un opérateur de sécurité
4è^{me} conférence sur la Sécurité des Architecture Réseaux et des Systèmes d'Information *(SARSSI 2009),* Luchon, France, juin 2009*, (à apparaître).*

Thèse et master recherche

[11] Karima SEDKI,
Raisonnement sous Incertitude et en présence de préférences : Application à la détection d'intrusions et à la corrélation d'alertes.
Thèse de doctorat de l'Université d'Artois, Décembre 2008**.**

[12] Karima SEDKI,
Autour de la négation dans la logique du choix qualitatif.
Thèse de Master Recherche SIA (Systèmes Intelligents et Applications). Université d'Artois, Juillet 2005.

3.3. CONFERENCES ET SEMINAIRES

- **Septembre 2006 :** Présentation d'un poster à la conférence ECAI 2006, « *An Alternative Inference for Qualitative Choice Logic »* à Riva del Garda, Italie.
- **Octobre 2006 :** Présentation d'un article à la conférence LFA 2006, « *La logique du choix qualitatif révisée* » à Toulouse, France.
- **25/10/2007 :** Présentation d'un séminaire aux Journées des Doctorants JDD 2007, « Traitements des informations incertaines dans le cadre de la détection d'intrusions ».
- **30/10/2007 :** Présentation d'un article à la conférence ECSQARU 2007, « *A Revised Qualitative Choice Logic for Handling Prioritized Preferences » à Hammamet, Tunisie.*
- **05/12/2007** *:* Présentation d'un travail intitulé « Application de la logique des préférences à la corrélation d'alertes » dans le cadre d'une réunion de travail du projet PLACID (Probabilistic graphical models and Logic's for Alarm Correlation in Intrusion Detection) à Paris.
- **27/07/2008 :** Présentation d'un article intitulé « Alert Correlation based on a Logical Handling of Administrator Preferences and Knowledge », à la conférence SECRYPT 2008, Portugal.

3.4. PARTICIPATION A DES PROJETS

1. DADDi (Dependable and Anomaly Detection with Diagnocis)

- **Lien :** http://www.rennes.supelec.fr/DADDi/
- **Période :** 2004-2008
- **Partenaires :** CRIL (Centre de Recherche Informatique de Lens), ENSTB (Département Réseaux et Services Multimédia - équipe SERES), IRISA (projet ADEPT), Supélec(Campus de Rennes - équipe SSIR), France Télécom R&D (site de Caen).
- **Contributions :**
  - Arbres de décision/Réseaux Bayésiens pour la sélection d'attributs pertinents (voir publication [5]).
  - Développent d'un outil de formatage du trafic réseau brut (formatage hors ligne et en temps réel), voir la publication [4])
  - Définition d'un jeu d'attributs pour la détection des activités malveillantes [11].
  - Formatage d'une base de données contenant du trafic brut capturé dans le campus universitaire Supélec. Cette base est d'une capacité 100 Go, sauvegardée dans 100 fichiers Tcpdump.

2. PLACID (Probabilistic graphical models and Logics for Alarm correlation in Intrusion Detection)

- **Lien :** http://placid.insa-rouen.fr/
- **Période :** 2006-2009
- **Partenaires :** Supélec Rennes, SSIR team (Sécurité des Réseaux et des Systèmes d'Information) , CRIL (Centre de Recherche en Informatique de Lens), LITIS (Laboratoire d'Informatique, Traitement de l'Information et des Systèmes), Institut National des Sciences Appliquées (INSA) de Rouen / Universités de Rouen et du Havre.
- **Contributions :**
  - Développement d'une nouvelle approche de corrélation d'alertes. Cette approche se base sur une approche logique permettant d'exprimer les connaissances et préférences d'un administrateur réseau afin de réduire le nombre d'alertes à analyser (voir les publications [6], [7]).

3.5. ANIMATIONS SCIENTIFIQUES

- ➔ Membre du comité d'organisation de la conférence « Logique Floue et ses Applications » *LFA 200*8 à Lens, Université d'Artois.
- ➔ Participation aux journées portes ouvertes à l'université d'Artois et à l'IUT de LENS (2007, 2008 et 2009).
- ➔ Membre de comité d'organisation des journées des doctorants du CRIL (Lens), Octobre 2007.