N° d'ordre : …/…

Faculté des Sciences de Kénitra
CED Sciences et Techniques
FD Mathématiques, Informatique et Applications

THÈSE

Présentée pour l'obtention du

Doctorat National

Par

Mr Elkhadir Zyad

Sous le thème

Développement de nouvelles variantes d'algorithmes d'extraction des caractéristiques basées sur PCA/LDA pour la détection des intrusions dans un réseau TCP/IP.

Soutenue le 07-07-2018

Devant le jury composé de :

Pr. Belghiti Moulay Taib (ENSA de Kénitra)	President
Pr. Ibrahimi Khalil (faculté des sciences de Kénitra)	Rapporteur
Pr. Messoussi Rochdi (faculté des sciences de Kénitra)	Rapporteur
Pr. Bouhorma Mohammed (FST de Tanger)	Rapporteur
Pr. Boujiha Tarik (ENSA de Kénitra)	Examinateur
Pr. Benattou Mohammed (faculté des sciences de Kénitra)	Directeur
Pr. Chougdali Khalid (ENSA de Kénitra)	Co-Directeur

Faculté des Sciences- Université Ibn Tofail, BP 133, Kénitra-Maroc.

Tel : 0537329400 , fax : 0537329433, http://fs.uit.ac.ma/

AVANT PROPOS

Les travaux présentés dans cette thèse ont été effectués au Laboratoire Système de télécommunications et ingénierie de la décision (LASTID) au sein de la Faculté des Sciences de Kenitra sous la direction de Mr. M. Benattou Professeur à la Faculté des Sciences de Kenitra et Mr. K. Chougdali Professeur à l'école nationale des sciences appliquées de Kenitra (ENSAK).

Je tiens à exprimer ma profonde gratitude aux professeurs M. Benattou et K. Chougdali pour leur disponibilité, leur grande rigueur scientifique et pour le soutien qu'ils m'ont accordé depuis qu'ils dirigent mes travaux. Je voudrais leur exprimer ma reconnaissance pour l'aide qu'ils m'ont constamment octroyée tout au long de ce travail, qu'ils trouvent, en ce mémoire, le témoignage de mes sincères remerciements.

Je tiens à exprimer ma haute considération à Mr. MT. Belghiti, Professeur à l'école nationale des sciences appliquées de Kenitra (ENSAK), d'avoir présidé ce jury de thèse.

Je présente à Mr. M. Bouhorma professeur à la Faculté des Sciences et techniques de Tanger, Mr. R. Messoussi et Mr. K. Ibrahimi tous professeurs à la Faculté des Sciences de Kenitra l'expression de ma profonde reconnaissance d'avoir accepté d'évaluer cette thèse.

Je suis très reconnaissant à Mr. T. Boujiha Professeur à l'école nationale des sciences appliquées de Kenitra (ENSAK) d'avoir participé à mon jury de thèse. Je lui exprime toute ma gratitude.

Bien sûr, je ne peux terminer sans remercier mes proches de tout coeur et notamment mes parents qui, au cours de ces années de thèse, m'ont toujours soutenu et encouragé.

À ma source de courage, à ceux que j'ai de plus cher :
Ma mère et mon père.

Liste des publications de l'auteur

Revues internationales

1. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohamed. Intrusion Detection System Using PCA and Kernel PCA Methods. **IAENG International Journal of Computer Science**, 2016, vol. 43, no 1, p. 72-79.

2. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohamed. Network Intrusion Detection System Using PCA by Lp-Norm Maximization Based on Conjugate Gradient. **International Review on Computers and Software (IRECOS)**, 2016, vol. 11, no 1, p. 64-71.

3. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohamed. Improving Network Intrusion Detection using Geometric Mean LDA. **International Journal of Network Security**, Sept. 2018, Vol.20, No.5, PP.820-826.

Conférences internationales à comité de lecture

1. KHALID, Chougdali, ZYAD, Elkhadir, et MOHAMMED, Benattou. Network intrusion detection system using L1-norm PCA. In : Information Assurance and Security (IAS), 2015 11th International Conference on. IEEE, 2015. p. 118-122.

2. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohammed. Intrusion detection system using PCA and kernel PCA methods. In : Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015. Springer, Cham, 2016. p. 489-497.

3. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohammed. A Median Nearest Neighbors LDA for Anomaly Network Detection. In : International Conference on Codes, Cryptology, and Information Security. Springer, Cham, 2017. p. 128-141.

4. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohammed. Combination of R1-PCA and median LDA for anomaly network detection. In : 2017 Intelligent Systems and Computer Vision (ISCV). IEEE, 2017. p. 1-5.

5. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohammed. An effective cyber attack detection system based on an improved OMPCA. In : Wireless Networks and Mobile Communications (WINCOM), 2017 International Conference on. IEEE, 2017. p. 1-6.

6. ELKHADIR, Zyad, CHOUGDALI, Khalid, et BENATTOU, Mohammed. An effective network intrusion detection based on truncated mean LDA. In : Electrical and Information Technologies (ICEIT), 2017 International Conference on. IEEE, 2017. p. 1-5.

ABSTRACT

Due to the large growth of network traffic in term of size and complexity, intrusion detection systems have shown a lot of limits such as the detection rate deterioration, the rising in false positive rate and the enormous CPU time consumption. To overcome these weaknesses, this thesis proposes new improvements concerning two feature extraction algorithms namely principal component analysis (PCA) and linear discriminant analysis (LDA). These contributions come with solutions of some mathematical limits encountered by PCA and LDA on the one hand, on the other hand they extract the useful information which gives the IDS a great ability to distinguish between normal network connections and intrusive connections. Among the PCA limits, we note its nonlinear nature and its sensitivity to outliers due to the use of the L2 norm and the arithmetic mean. To remedy the first weakness, we propose to use KPCA with new kernel functions: the power kernel and the spherical kernel. In KPCA we employ the kernel trick to transform the data entry into an implicit feature space. Then, these data are processed in this space to produce nonlinear and discriminating features that facilitate the connections classification. To solve the second problem, two solutions are proposed. The first technique called PCA-Lp tries to find projections that maximize total covariance using Lp norm instead of L2 norm. This approach takes profit of the conjugate gradient. The second method called QR-OMPCA allows us to calculate a more robust average. This algorithm is characterized by the use of QR decomposition instead of SVD.

LDA also suffers from outliers. It uses the L2 norm in it mathematical formulation and uses the arithmetic mean to compute the dispersion matrices. In order to overcome this deficiency, this thesis comes with geomean LDA and R1-PCA + median LDA. The first variant replaces the average with a geometric mean which is more robust to the outliers. The second combines two powerful algorithms to reap the benefits of the R1 norm and the median. Another problem is encountered by LDA, it concerns the handled data structure type. The supervised algorithm pays more attention to the global structure of classes. Therefore, the produced discriminants characteristics are often imprecise. To remedy this situation, the variant Median NN-LDA approach can effectively get the local structure of data by working with samples that are near to the median of every data class. The further samples will be essential for preserving the global structure of every class. Therefore, this method offers better separability between classes in the reduced space and facilitates the intrusion detection.

RÉSUMÉ

Face à la grande croissance du trafic réseau en taille et en complexité, les systèmes de détection d'intrusion ont montré beaucoup de limites, tel que, la diminution importante de taux de détection, une augmentation de taux des faux positifs et une consommation fulgurante de temps au niveau de la phase d'apprentissage. Afin de pallier cette carence, les travaux de cette thèse contribuent à la mise au point de nouveaux algorithmes d'extraction de caractéristiques basées sur l'analyse en composantes principales (PCA) et l'analyse discriminante linéaire (LDA). Ces variantes visent à résoudre quelques limites d'ordre mathématique rencontrées par PCA et LDA d'une part, et l'extraction de l'information utile qui dote l'IDS d'une grande capacité à distinguer entre les connexions réseau normales et les connections intrusives d'autre part. Parmi les limites de PCA, nous notons sa nature non linéaire et sa sensibilité aux données aberrantes du a l'utilisation de la norme L2 et la moyenne arithmétique. Pour remédier à la première faiblesse, nous proposons l'utilisation de KPCA avec des nouvelles fonctions a noyau. A savoir, le noyau a puissance et le noyau sphérique. L'idée de base de KPCA est fondée sur l'utilisation de l'astuce noyau pour transformer les données d'entrée dans un espace de caractéristiques implicite. Puis, ces données sont traitées dans cet espace pour produire des caractéristiques non linéaires et discriminantes qui facilitent la classification des connections. Pour résoudre le deuxième problème, on propose deux solutions. La première appelée PCA-Lp est basée sur le gradient conjugué. Cette technique tente de trouver des projections qui maximisent la covariance totale en utilisant la norme Lp ($p < 2$) au lieu de la norme L2. La deuxième méthode appelée QR-OMPCA essaye de calculer une moyenne plus robuste. Cette approche est caractérisée par l'utilisation de la décomposition QR au lieu du SVD.

LDA souffre elle aussi des données aberrantes. Elle utilise la norme L2 dans sa formulation mathématique et exploite la moyenne arithmétique pour calculer les matrices de dispersion. Pour éviter cela, on propose geomean LDA et R1-PCA+median LDA. La première variante remplace cette moyenne par une moyenne géométrique plus robuste face aux données aberrantes. La deuxième combine deux puissants algorithmes pour profiter des avantages de la norme R1 et du médiane. Un autre problème est rencontré par LDA. Elle concerne le type de structure de données manipulée. L'algorithme supervisé accorde plus d'attention à la structure globale des classes. En conséquence, les caractéristiques discriminantes produites sont souvent imprécises. Pour remédier à cette situation, la variante Median NN-LDA se base sur les éléments les plus proches du médiane de chaque classe pour préserver les distributions locales et globales. Par conséquent, cette méthode offre une meilleure séparabilité entre les classes dans l'espace réduit. Ce qui facilite la détection des intrusions par la suite.

Table des matières

Figures

Tableaux

Acronymes

ANN	Artificial Neural Network.
CART	Classification And Regression Trees.
CCA	Canonical Correlation Analysis.
DDoS	Distributed Denial of Service.
DoS	Denial of Service.
GA	Genetic Algorithm.
HIDS	HostBased Intrusion Detection System.
HMM	Hidden Markov model.
ICA	Independent Component Analysis.
ICPSO	Improved Chaotic Particle Swarm Optimization.
IDS	Intrusion Detection System.
K-NN	K nearest neighbors.
KPCA	Kernel Principal Component Analysis.
L1-PCA	Principal Component Analysis using L1 norm.
LDA	Linear Discriminant Analysis.
LFDA	Local Fisher Discriminant Analysis.
LKPCA	Local Kernel Principal Component Analysis.
LLE	Locally Linear Embedding.
LPMIP	Locality Preserved Maximum Information Projection.
Median NN-LDA	Linear Discriminant Analysis based on Median Nearest Neighbors.
MITM	Man In The Middle.
MMC	Maximal Margin Criterion.

NIDS	Network Based Intrusion Detection System.
NMF	Non negative Matrix Factorization.
OMPCA	Optimal Mean Principal Component Analysis.
PCA	Principal Component Analysis.
PCA-Lp	Principal Component Analysis using Lp norm.
Pseudo LDA	Linear Discriminant Analysis using the Pseudo Inverse.
QR-OMPCA	Optimal Mean Principal Component Analysis based on QR decomposition.
QR-OMPCA	Principal Component Analysis based on R1 norm.
R1-LDA	Linear Discriminant Analysis based on R1 norm.
R2L	Remote To Local.
SOM	Self Organizing Maps.
SSL	Secure Sockets Layer.
SSS	Sample Size Problem.
SVD	Singular-value decomposition.
TCP	Transmission Control Protocol.
U2R	User To Root.
VPN	Virtual Private Network.
WMMC	Weighted Maximal Margin Criterion.

Introduction générale

Les réseaux et les systèmes informatiques sont devenus des outils indispensables au fonctionnement des entreprises. Ils sont aujourd'hui déployés dans tous les secteurs professionnels : les universités, les banques, les assurances ou encore le domaine militaire. L'information gérée par ces systèmes fait l'objet de convoitises. Elle peut être exposée à des attaques qui exploitent des éléments vulnérables du système d'information. La détection des actions malveillantes est rapidement devenue une nécessité. Les mesures de prévention se sont révélées insuffisantes et ont amené la création de systèmes de détection d'intrusions (IDS : Intrusion Detection Systems).

Une intrusion est définie comme étant toute tentative pouvant nuire à l'intégralité, la confidentialité ou la disponibilité dans le réseau ainsi que toute tentative visant à contourner les dispositifs de sécurité mis en place sur le réseau ou une machine. Ces tentatives d'intrusions peuvent être bénignes comme extrêmement dangereuses et préjudiciable pour l'entreprise. Les systèmes de détections d'intrusions peuvent être classés en trois catégories : ceux qui cherchent à détecter des malveillances (on parle alors d'approches par scénarios), ceux qui cherchent à détecter des anomalies (on parle alors d'approches comportementale) et ceux qui se basent sur les agents mobiles. Dans le premier cas, le modèle de détection repose sur la connaissance que l'on a des attaques tandis que dans le second cas, celui-ci repose sur la connaissance que l'on a de l'entité surveillée en situation de fonctionnement normal. Une approche comportementale présente l'avantage de pouvoir détecter des attaques encore inconnues au moment de la modélisation. Toutefois, la construction d'un tel modèle de détection peut être une tâche difficile. En effet, il n'est pas simple de définir le comportement normal de l'entité surveillé et toute erreur de modélisation risque d'entraîner la levée de fausses alertes. En plus de ça, ce type d'approche demande un grand temps d'apprentissage. Cela est dû principalement à la manipulation de l'énorme trafic réseau.

Pour pallier le problème de l'approche comportementale, plusieurs techniques d'apprentissage automatique ont été massivement appliquées ces dernières années. À titre d'exemple : Les arbres de décision, les algorithmes et la programmation génétique, les K plus proches voisins (KNN), les réseaux bayesiens, les réseaux de neurones ou même les SVM. Les travaux de cette thèse s'inscrivent dans l'utilisation et le développement de nouvelles méthodes d'extraction de caractéristiques pour la détection d'intrusions. Ainsi, au cours de cette thèse, nous aurons à concevoir de nouveaux algorithmes qui essayent de contourner les limitations des techniques classiques en particulier ceux basés sur l'analyse en composantes principales et l'analyse discriminante.

Dans la première partie, nous faisons un rappel sur les généralités de la sécurité informatiques. Ensuite, nous décrivons les systèmes qui contribuent à la sécurité des réseaux informatiques en particulier, à savoir les pare-feux, les systèmes cryptographiques, les scanners de vulnérabilités et les pots de miel.

Le deuxième chapitre est consacré à une description détaillée des systèmes de détection d'intrusion (IDS : Intrusion Detection System). On présente les différentes approches de détection d'intrusion (approche par scénario et approche comportementale) et les différents types d'IDS (HIDS, NIDS). Après, on décrit les différents inconvénients de l'approche comportementale. Par la suite, on présente quelques outils commerciaux de détection des intrusions.

Le troisième chapitre décrit l'architecture globale de l'IDS proposé basé sur les algorithmes d'extraction de caractéristiques. L'analyse en composante principale et l'analyse discriminante linéaire seront décrites tout en mettant en évidence leurs limitations dans le cadre de la détection des anomalies. En particulier, le problème de la singularité des matrices, et la sensibilité aux données aberrantes. Par la suite, on présente les solutions les plus connues à ce jour pour surpasser ces difficultés.

Le quatrième chapitre permettra de mettre en évidence les limitations de l'analyse en composantes principales (PCA). En particulier, la non-linéarité des données et la sensibilité aux données aberrantes. Nous proposons dans un premier temps un algorithme (KPCA) qui se base sur la théorie des noyaux pour rendre l'algorithme non linéaire. A ce stade, nous introduisons aussi deux nouvelles fonctions noyaux qui n'ont pas été utilisées pour la détection des intrusions. Ces deux fonctions ont l'avantage d'être plus performants que les noyaux conventionnels. Dans un deuxième temps, nous proposons une nouvelle variante (PCA-Lp) qui permet de résoudre le deuxième problème en s'appuyant sur la norme Lp pour limiter l'effet des données aberrantes qui peuvent surgir dans le trafic normale. Par la suite, nous développons une autre méthode (QR-OMPCA) qui est une extension de l'algorithme PCA. Cette nouvelle méthode a pour objectif principal l'optimisation de PCA en s'appuyant sur la décomposition QR et en intégrant une fonction de calcul d'une moyenne plus représentative.

Le cinquième chapitre introduira de nouvelles variantes de l'analyse discriminante linéaire (LDA) tel que Median NN-LDA, geomean LDA et R1-PCA+median LDA. Ces méthodes ont pour but l'optimisation de LDA en proposant différentes approches. La première méthode résout le problème d'inconsistance des données en se basant sur les éléments les plus proches de la médiane de chaque classe. Par conséquent, cette méthode offre une meilleure séparabilité entre classes dans l'espace réduit. Ce qui facilite la détection des intrusions par la suite. Geomean LDA propose de remplacer la moyenne classique utilisée dans LDA par la moyenne géométrique. Cette dernière est plus robuste vis-à-vis des données aberrantes. La dernière méthode combine deux puissants algorithmes R1-PCA et median LDA. En suivant cette approche, on tire profit des

avantages de la norme R1 et de la médiane. Ce qui offre une résistance remarquable face au données aberrantes et évite la singularité des matrices.

Pour évaluer nos méthodes nous avons effectué une série de tests sur des bases de données standards (KDDcup99 et NSL-KDD). Les résultats de simulation que nous avons obtenus montrent que les taux de détection se sont nettement améliorés par rapport à ceux des méthodes existantes, avec une baisse considérable des faux positifs.

CHAPITRE

1 Introduction à la Sécurité Informatique

“If you spend more on coffee than on IT security, you will be hacked. What’s more, you deserve to be hacked”
RICHARD CLARKE

1.1 Introduction

Dès le début du vingt et unième siècle, internet est devenu un outil primordial utilisé pour effectuer diverses activités tel que l’étude, l’achat en ligne, la communication, etc. Cette révolution technologique a été accompagnée par une augmentation phénoménale du nombre d’utilisateurs d’internet, on compte aujourd’hui plus de 2,5 milliards d’utilisateurs. Les informations qui transitent par internet peuvent être importantes, critiques, secrètes et confidentielles alors que les concepteurs d’internet n’ont pas prévu la sécurité de ces informations. Leur initial but était d’interconnecter les différents réseaux informatiques. Cependant, ces systèmes connaissent des faiblesses. Certains utilisateurs mal intentionnés peuvent exploiter les vulnérabilités d’internet pour essayer d’accéder à des informations sensibles dans le but de les lire, les modifier ou les détruire. Dès lors que ce réseau internet est apparu comme cible potentielle d’attaques, leur sécurité est devenue un enjeu incontournable.

Vu que les différentes infrastructures des divers secteurs sociaux, économiques, militaires, gouvernementales sont connectées à internet, une attaque informatique est considerée comme une arme très dangereuse et très destructive. Grâce à une attaque informatique on peut paralyser tout un pays comme l’attaque contre l’Estonie en avril 2007. On peut aussi retarder ou saboter un projet stratégique comme l’attaque contre le centre nucléaire d’Iran en septembre 2010. L’attaque peut mener à l’interception de données sensibles et confidentielles aussi. En effet, en février 2014, les établissements américains du groupe de loisirs Las Vegas Sands sont victimes d’une cyberattaque majeure incluant le piratage du réseau informatique, un vol massif de données confidentielles puis la mise hors service d’une partie importante du système d’information et de télécommunications. Par conséquent, l’enjeu des attaques informatiques ou les cyberattaques est sorti du cadre d’ambition et de loisir, il est devenu un projet militaire stratégique, on parle aujourd’hui du cyber

guerre.

1.2 Les causes de l'insécurité

Au sein d'un réseau informatique, on distingue généralement cinq types de faille qui peuvent causer l'état d'insécurité :

1. **Les failles physiques** : généralement dans une entreprise ou une administration la sécurité d'accès aux matériels informatiques n'a pas une grande importance. On peut facilement accéder aux materiel pour des raisons divers tel que l'élaboration des tests, la maintenance du système ou le nettoyage des disques dur. L'exploitation de cet accès physique entraine le vol de mots de passe, la suppression des données, l'usurpation des identités et l'injection des programmes malveillants.

2. **Les failles réseaux** : les réseaux informatiques sont fondés sur des normes et des standards bien réfléchis où plusieurs organismes collaborent pour les perfectionner. Malgré tous les efforts faits, il existe certaines failles ou détournements de fonctionnement des standards exploitables. Le problème avec les failles réseau c'est la complexité de leurs corrections qui varie selon la taille du réseau. À titre d'exemple, corriger les failles réseau d'internet est utopique, c'est la raison pour laquelle on se contente de faire des améliorations comme le passage vers IPV6 ou IPSec.

3. **Les failles systèmes** : les systèmes d'exploitation sont de plus en plus sophistiqués. Ils intègrent différent mécanisme de sécurité comme les mots de passe, les logs, séparation des privilèges,..., etc. La complexité, la mauvaise configuration ainsi que les faiblesses de certains mécanismes des systèmes d'exploitation représentent un danger pour les utilisateurs. Par exemple la complexité d'un mécanisme de sécurité pousse les utilisateurs à le désactiver, de plus la mauvaise configuration peut engendrer l'arrêt ou la saturation du système.

4. **Les failles applicatives** : Ces types de faille sont très connues et très répondues. Ils peuvent être causés par la mauvaise conception, le non-traitement des exceptions et les lacunes expoitées du langage de programmation. Ces failles peuvent engendrer beaucoup de problèmes qui influencent le fonctionnement du système.

5. **Les failles Web** : le monde du web représente la combinaison de différents entités tel que les réseaux, les système d'explitation et les applications. Les failles web peuvent être causées par l'une des failles précédemment citées ou par des failles qui résident au niveau des protocoles et des standards du fonctionnement du web.

1.3 Les différents types d'attaque/intrusion informatique

Une attaque ou intrusion informatique est littérairement définie par toute tentative de détruire, exposer, modifier, désactiver, voler ou obtenir un accès non autorisé ou toute utilisation non autorisée

d'une information, logiciels, physique comme un serveur, services, des personnes et de leurs qualifications, et les biens incorporels (ISO/IEC 27000, 2009). Pour l'aspect technique, on peut définir une attaque par l'exploitation de l'une des failles précédemment citées pour des fins illégales. Il existe cinq formes d'attaque que nous détaillons comme suit :

1. **L'attaque passive** : Ce type d'attaque représente tout acte qui nous permet de faire l'analyse et le décryptage du trafic, la surveillance des communications, et la capture des informations d'authentification. Les attaques passives peuvent entraîner la divulgation des informations ou des données à un attaquant sans que la victime soit consciente. L'interception du mot de passe, numéros de carte de crédit, des emails représente tous des attaques passives.

2. **L'attaque active** : Elle comprend toute tentative ayant pour but le contournement ou l'arrêt des fonctions de protection, l'introduction d'un code malveillant et le vol ou la modification des informations. Les attaques actives peuvent entraîner la divulgation et la diffusion des données, un déni de service, ou la modification des données.

3. **L'attaque de proximité ou externe** : Elle se base sur l'utilisation de la proximité physique du réseau ou du système qui a été obtenue grâce à l'entrée clandestine ou un accès ouvert afin de modifier, collecter ou refuser l'accès à l'information.

4. **L'attaque interne** : les attaques internes peuvent être intentionnelles ou non intentionnelles. Les attaques intentionnelles représentent les tentatives d'espionner, de voler ou d'endommager des informations, utiliser l'information de manière frauduleuse, ou interdire l'accès à d'autres utilisateurs autorisés. Les attaques non intentionnelles représentent le résultat d'une mauvaise manipulation, la négligence ou le manque de connaissances.

5. **L'attaque de distribution** : Ce type d'attaque comporte toute modification malveillante du matériel ou du logiciel lors de sa distribution. Elle consiste à introduire un code malveillant dans un produit comme un port dérobé pour obtenir un accès non autorisé à des informations ou une fonction système.

1.4 Exemples des attaques informatiques

Il existe plusieurs types d'attaque très connues dans le domaine de securité informatique. Nous détaillons ici trois exemples réputées par leurs dangerosités et les dégâts qui peuvent causer.

1.4.1 L'attaque de l'homme du milieu (man-in-the-middle)

Cette attaque est parfois appelée attaque de l'intercepteur. Elle a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis [39]. Pour réussir une telle attaque, il faut que la machine de l'attaquant soit physiquement entre les deux machines victimes ou que l'attaquant arrive

à modifier le routage réseau afin que sa machine devienne un des points de passage. Cette attaque adopte en general l'une des trois techniques.

Le Relais applicatif

Le principal point fort de cette technique réside dans le fait que le pirate n'est pas obligé de se trouver logiquement entre le client et le serveur. Le relais applicatif prend place quand le pirate a besoin de décoder des flux chiffrés entre deux machines. L'intrus cherche donc à se placer convenablement dans le reseau afin d'intercepter les flux circulant entre le client et le serveur d'une part, et être en mesure de les modifier d'autre part.

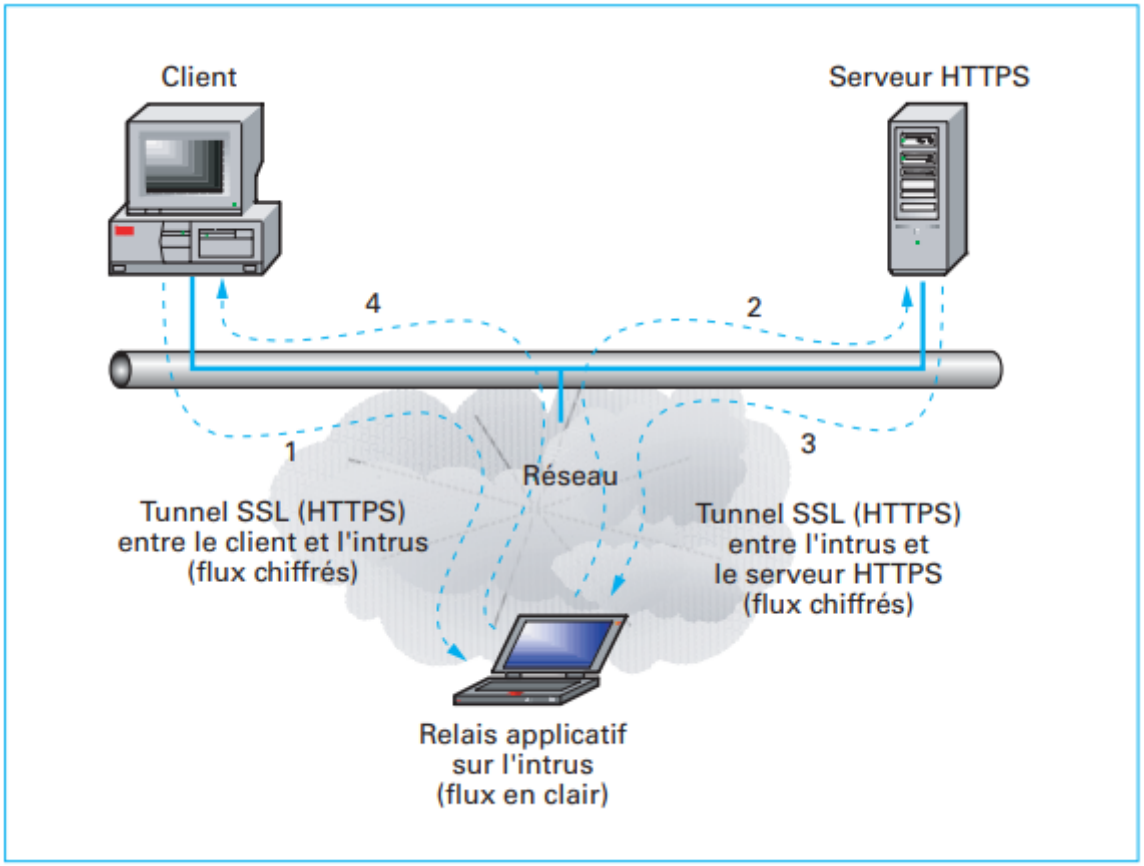


Figure 1.1: *Le relais applicatif*

La figure 1.1 montre le trafic du réseau tel qu'il se produit :

1. le client envoie sa requête HTTPS vers l'intrus qui n'est pas du tout sur le chemin logique entre le client et le serveur HTTPS.
2. l'intrus déchiffre la requête et la renvoie vers le serveur HTTPS.
3. le serveur HTTPS répond à l'intrus.
4. l'intrus répond au client.

On constate alors que :

L'attaquant possède les flux HTTPS (non chiffrés) et peut à n'importe quel moment modifier la demande HTTPS renvoyée vers le serveur HTTPS. Comme dernière étape, l'intrus peut falsifier les réponses renvoyées par le serveur HTTPS vers le client.

Le relais transparent

La technicité déployée dans ce type de relais est bien plus complexe que celle utilisée dans le relais applicatif. En effet, il ne s'agit plus de relayer des requêtes qui sont prévues pour être relayées, mais de se comporter comme un équipement du réseau transparent et de modifier les données à la volée dans le cas le plus difficile et le plus fréquent.

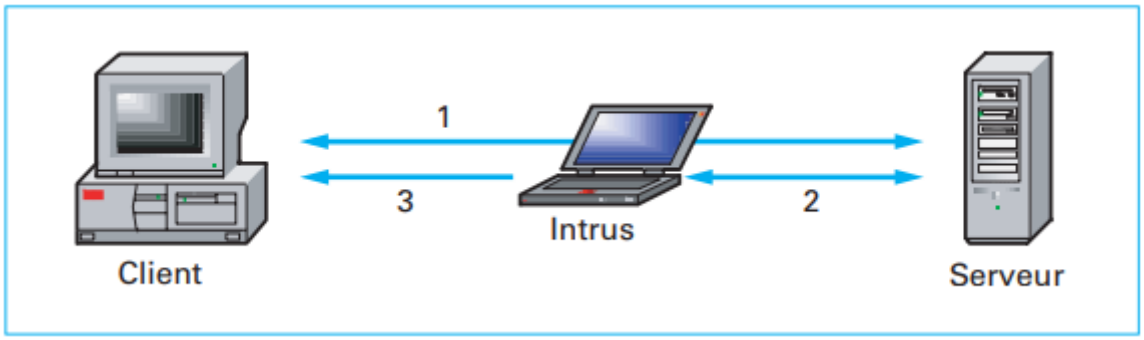


Figure 1.2: *Le relais transparent*

La figure 1.2 illustre les etapes suivies par le relais transparent. Dans l'étape 1 le client lance une session. De son coté, l'intrus est en position de relais transparent et se contente de réacheminer les paquets. Dans l'étape 2, l'intrus décide de passer à l'offensive. Il arrête donc de répondre au client et continue la session établie avec le serveur en tant que client. Finalement, dans l'étape 3, l'intrus met un terme a la session que le client avait avec le serveur en renvoyant au client une demande de fin de session (paquet FIN). Le client voit sa session interrompue et pense qu'il s'agit d'un problème ponctuel du réseau. Soit il arrête alors de travailler, soit il relance une nouvelle session que l'intrus se contente de réacheminer de manière transparente, puisqu'il dispose maintenant d'une session rien que pour lui.

La modification du routage

Pour se placer en situation de relais transparent, une modification du comportement du réseau est nécessaire de la part de l'intrus. Ce dernier essaye de changer le comportement du réseau afin d'être considéré comme un point de passage entre deux machines. Cependant, lorsque l'intrus se trouve en dehors du réseau local, ce type d'attaque est plus difficile à mener. Il faut alors s'attaquer au routage des paquets qui permet de franchir le concept de réseau local au profit du réseau global.

Pour ce faire, le pirate se base sur la redirection du protocole ICMP [39]. Le principe consiste à convaincre le réseau (les routeurs ou les systèmes présents sur le réseau) que le meilleur chemin n'est pas celui qu'il croit, mais plutôt celui qui passe par l'intrus.

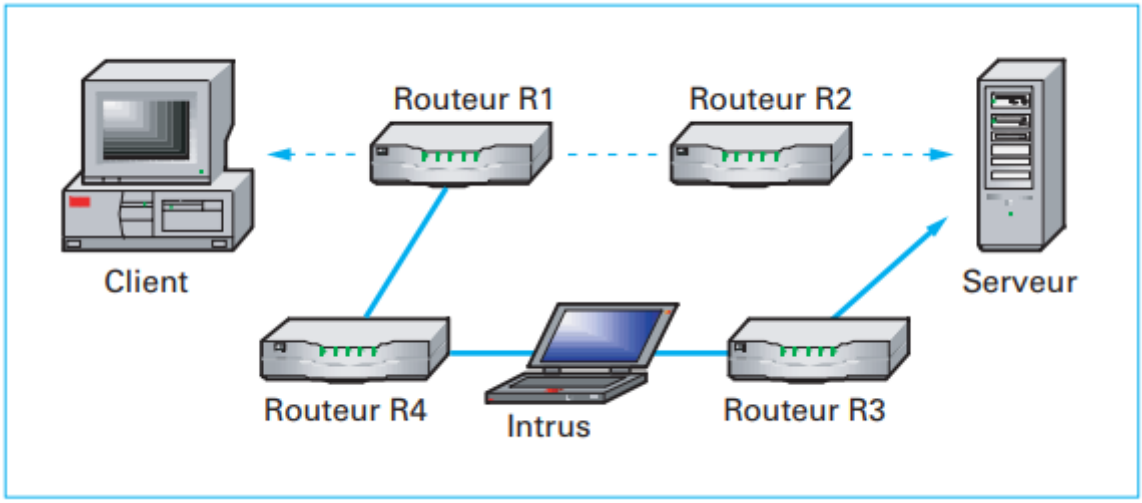


Figure 1.3: *La modification du routage*

Ainsi, comme c'est illustré dans la figure 1.3, l'intrus peut écouter ou modifier tout le trafic qui passe entre les routeurs R3 et R4. Afin de pouvoir faire de même pour le trafic transitant entre le client et le serveur, l'intrus envoie au serveur des paquets ICMP pour le convaincre que le routeur R3 est le meilleur chemin pour parler avec le client. L'intrus peut aussi procéder autrement en envoyant au routeur R1 des paquets ICMP afin de le convaincre que passer par le routeur R3 est le meilleur chemin pour atteindre le serveur.

1.4.2 L'attaque de déni de service distribué (DDoS)

Les attaques DDoS sont des attaques DoS effectuées à partir de plusieurs machines contrôlées par un attaquant appelées "bots". Leur but est l'inondation du trafic réseau afin d'empêcher son fonctionnement. Dans le scénario le plus fréquemment utilisé, toutes les machines sont engagées simultanément et commencent à générer autant de paquets que possible envers la machine cible afin de la surcharger. L'attaquant peut également contrôler le volume du trafic généré par chaque machine.

Les attaques DDoS présentent les caractéristiques suivantes :

1. **Usurpation d'adresse IP** : Généralement, les attaquants utilisent une technique d'usurpation des adresses IP source (IP spoofing) augmentant ainsi la difficulté à tracer les bots et rendant possible leur ré-utilisation pour d'autres attaques. En outre, bien que le traçage des bots puisse avoir lieu, ce dernier ne permet pas de tracer l'attaquant. Enfin, la grande variété des adresses des paquets envoyés complique la tâche des systèmes de défense. Outre cette caractéristique de dissimulation de l'attaquant et de ses bots, l'usurpation d'adresse permet aussi de mettre en oeuvre des attaques réflectives (reflector attack) [184], simples ou distribuées. Le principe de ce type d'attaque consiste pour un attaquant à envoyer des requêtes à un ou plusieurs serveurs sur Internet en remplaçant son adresse source par celle de la victime. Par conséquent, c'est cette dernière qui recevra les

réponses des serveurs interrogés. Dans ce cas, l'attaquant ne dispose pas uniquement d'un large botnet de machines détournées, mais il maintient également une liste de réflecteurs. Un réflecteur représente n'importe quelle adresse de serveur en mesure de retourner une réponse à une requête comme par exemple un serveur web, un serveur DNS ou un serveur NTP.

2. **Taille du botnet** : Il existe de nombreuses techniques performantes pour tracer des machines attaquantes. Cependant, bien que les bots puissent être tracés, la question des actions qui pourraient être prises contre des milliers voir des centaines de milliers de machines reste ouverte. En outre, dans le cas d'une attaque flooding générée par un grand nombre de bots, bien que les systèmes de défense puissent bloquer ou dévier le trafic de l'attaque, c'est le chemin vers la victime et son point d'entrée réseau qui seront complètement submergés. Afin d'éviter de telles situations, les mécanismes de défense doivent détecter et bloquer ce trafic malveillant au plus proche de sa source [57, 119].

3. **Similitude entre le trafic d'attaque et le trafic légitime** : Tout type de trafic peut être utilisé afin d'effectuer un déni de service mais certains types de trafic nécessitent plus de volume que d'autres pour atteindre cet objectif. Ainsi, une attaque de déni de service peut être dissimulée dans une quantité conséquente de trafic qui contient des paquets légitimes rendant difficile la différentiation entre une attaque et un phénomène légitime de pic de charge (flash crowd). Ceci explique la difficulté à réaliser un système de défense contre les DDoS qui repose sur le contrôle des paquets.

1.4.3 Attaque par virus

Un virus informatique est tout programme capable de se reproduire par lui-même. Il peut prendre la forme d'une routine ou d'un programme une fois activé, il utilise tous les moyens pour empoisonner un système d'information. Les virus informatiques représentent le type d'attaque le plus fréquent. Le cycle de vie d'un virus commence par la création, puis la reproduction, ensuite l'activation. Il existe plusieurs types de virus qu'on peut les résumer par :

1. Virus de secteur d'amorçage.
2. Virus d'infection des fichiers (parasites).
3. Virus non-résidents mémoire.
4. Virus résidents mémoire.
5. Virus multiformes.
6. Virus réseau et vers (worms).
7. Chevaux de Troie.
8. Virus furtifs.

1.5 Objectifs de la sécurité informatique

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de stocker ou de faire circuler ces données. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger. La sécurité informatique, d'une manière générale, consiste à s'assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. La sécurité informatique vise généralement cinq principaux objectifs :

1. La confidentialité : consiste à préserver la révélation non autorisée d'information sensible. La révélation pourrait être intentionnelle comme les attaques qui visent a casser le chiffrement des données et lire les informations, ou involontaire dû au manque de vigilance ou de l'incompétence des individus qui manient les informations .

2. L'intégrité : c'est-à-dire garantir que les données sont bien celles que l'on croit être. Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).

3. La disponibilité : assure que les utilisateurs autorisés ont un accès opportun et non interrompu aux informations dans le système et le réseau.

4. la non-répudiation de l'information : c'est la garantie qu'aucun des correspondants ne pourra nier la transaction. Des transactions permettent de prouver de façon certaine et non contestable par les parties en présence que telle ou telle action a bien été effectuée par une personne et non par une autre. Ce n'est pas à vous de prouver que vous n'avez pas acheté un yacht à Monaco il y a 8 jours avec votre carte bancaire, mais à votre banque de fournir la preuve que vous avez bien tapé votre code PIN sur le terminal du vendeur de bateau.

5. L'authentification : Consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

1.6 Protection du système d'information

Les attaquants peuvent suivre un plan d'attaque bien précis pour exécuter leurs exploits. Leurs objectifs sont distincts et multiples. Il existe l'attaquant hacker, qui dans un but d'approfondissement de connaissances, essaye d'explorer les failles de sécurité dans un système informatique. Cette personne partage publiquement ses découvertes et évite la destruction intentionnelle des données. Le deuxième type d'attaquant, appelé cracker, essaie de violer l'intégrité du système. D'une manière générale, il est facilement identifiable à cause de ses actions destructives. Néanmoins il faut connaitre la différence entre un expert qui cherche les exploits et conçoit lui même les programmes, et un gamin scripteur qui utilise la technologie existante dans un but malveillant. Les

différents types d'attaquants cherchent à découvrir les propriétés du réseau cible avant de lancer les attaques. On parle généralement de la reconnaissance qui peut être passive ou active. Ayant récolté les informations nécessaires, ils lancent leurs vraies attaques pour exploiter le système. Ensuite ils créent des portes dérobées pour garantir des futurs accès faciles au système compromis. Enfin ils effacent leurs traces des journaux de sécurité [3].

Les attaquants manipulent plusieurs moyens pour réussir chaque phase d'attaque. La disponibilité des outils d'attaques et la richesse des sources d'informations augmentent le risque des intrusions. Par conséquent les administrateurs sécurisent de plus en plus leurs systèmes informatiques. Ils s'appuient sur diverses solutions comme les pare feux, la cryptographie, les scanners de vulnérabilités et les pots de miel. Nous détaillons dans la suite chacune de ces méthodes et nous soulignons ces avantages et ces limites.

1.6.1 Pare-feux

Un pare-feu est considéré comme l'essentielle ligne de défense protégeant les services et les applications d'un réseau. Pour être efficace, cet outil se base sur quelques ressources matérielles ainsi que leur disponibilité. Si par exemple un pare-feu est surchargé suite à une attaque DDoS, l'accès aux applications de l'entreprise devient presque impossible. Alors, pour eviter que cette situation ne se produise, les entreprises sont obligées d'investir massivement dans l'achat d'équipements supplémentaires. Cette situation impose des coûts importants pour la plupart des entreprises, notamment les petites et moyennes entreprises. Selon [79] et [148], le coût de déploiement et d'entretien d'un pare-feu physique est estimé à 116.075$ pour la première année et un coût annuel de 108.200$ pour une société américaine de taille moyenne caractérisé par 5Mbps de connectivité Internet. De plus, des coûts supplémentaires sont engagés pour l'embauche, l'entretien, la surveillance et les mises à jour. Les politiques de sécurité du pare-feu sont des règles de filtrage ordonnées qui définissent les actions effectuées sur les paquets afin de satisfaire à des conditions spéciales. Il existe trois principaux types de pare-feu [22, 28, 149] :

1. **Le pare-feu qui filtre les paquets** : plus communément appelé « Packet-Filter ». Il possède quelques avantages : faible surcharge de trafic à haut débit et peu coûteux. Cependant, son niveau de sécurité est très faible. Les routeurs possèdent aussi cette fonctionnalité qui consiste à examiner les paquets au niveau des couches réseau et/ou transport, permettant à des paquets seulement autorisés à être transférés.

2. **Le pare-feu applicatif** : ce dispositif contient un agent mandataire agissant comme un lien transparent entre deux hôtes qui souhaitent communiquer entre eux. Le pare-feu applicatif ne permet pas de connexion directe entre les deux hôtes. Parmi ces inconvénient, il ne protège pas contre les attaques au niveau des couches inférieures, il nécessite un programme distinct pour chaque application et une consommation de ressources très élevée pour des performances limitées.

3. **Le pare-feu contemporain (NGWF)**: cette technologie représente une évolution par rapport au pare-feu traditionnel [129]. Elle intégre une variété de fonctions de sécurité comme le filtrage anti-spam, anti-virus, un système de détection ou de prévention d'intrusion (IDS / IPS). Les NGFWs fournissent également une inspection plus granulaire et une plus grande visibilité du trafic que celle des pare-feu traditionnels [4]. Néanmoins, ce type de technologie fait face à une augmentation du trafic "crypté" ce qui met en lumière ces limitations.

Même si l'emploi d'un pare-feu présente plusieurs points forts en termes de sécurité, la capacité de filtrage de l'équipement demeure gravement limitée. En effet, cela dépend de son intégration dans le réseau, la complexité de la politique de sécurité et le débit de paquets qui provoquent dans la majorité des cas un déni de service. Certains articles comme celui de Liu [104] proposent des améliorations aux algorithmes et méthodes d'analyse des pare-feu. Par ailleurs, pour obtenir des résultats satisfaisants, ces algorithmes demandent un grand nombre de ressources matérielles.

1.6.2 Scanners de vulnérabilités

Les scanners de vulnérabilités facilitent la découverte des failles de sécurité. Ils sont employés par les attaquants afin de detecter les points faibles du réseau cible d'une part. D'autre part, les administrateurs peuvent en tirer profit pour renforcer la sécurité de leurs systèmes informatique. Nous citons à titre d'exemple Nessus [13] , Whisker[134] et Saint. Néanmoins, les scanners contiennent quelques faiblesses qui peuvent être résumées en trois points : l'exhaustivité, la mise à jour et l'exactitude. Effectivement, malgré le nombre elevé de failles détectées, les scanners actuels demeurent impuissants face à toutes les faiblesses possibles. De plus, la mise à jour de ces produits ne suit pas le rythme de la découverte des nouvelles vulnérabilités. Enfin, la modification des bannières des services scannés permet de dissuader facilement le scanner ce qui entraîne parfois un responsable de sécurité à chasser des vulnérabilités fantômes.

1.6.3 Outils d'archivage

Plusieurs utilitaires d'archivage sont offerts par la majorité des systèmes d'exploitation. Par exemple, le système UNIX propose d'utiliser l'outil daemon syslogd. Ce dernier enregistre dans des logs de sécurité les principales opérations exécutées sur le système. Parmi les fichiers log créés, trois sont susceptibles d'être manipulés par les attaquants à savoir wtmp, utmp et lastlog.

1. wtmp : contient un historique des connexions/déconnexions avec l'heure, le service et le terminal concerné,

2. utmp : liste les utilisateurs connectés à un moment donné,

3. lastlog : contient un historique des dernières connexions.

Les attaquants eliminent dans la plupart du temps les entrées des journaux de sécurité et principalement des trois fichiers mentionnés ci-dessus. De leur part, les administrateurs analysent l'intégrité

de ces fichiers afin d'identifier les éventuelles modifications. Ils gardent egalement des copies de ces fichiers sur des machines distantes inconnues par les attaquants. En dernière étape, et afin de résister aux arrêts intentionnels des daemons d'archivage, les responsables de sécurité diversifient l'utilisation des outils de sauvegarde. Par conséquent les journaux de sécurité constituent une source intéressante pour analyser et détecter les attaques. Cependant, ces fichiers contiennent beaucoup d'informations normales et anormales. La taille énorme de ces fichiers pose souvent des problèmes de stockage et d'exploration du contenu. Les administrateurs fournissent aussi un effort important pour localiser dans ces fichiers les activités anormales, comprendre les objectifs des attaquants et déterminer les vulnérabilités exploitées du système.

1.6.4 Cryptographie

La cryptographie assure quatre qualité caractérisant n'importe quel système d'information: la confidentialité, l'intégrité, la non répudiation et l'authenticité des données. Elle est souvent employée dans diverses applications réseaux telles que la messagerie, les connexions à distance, les réseaux privés et les serveurs web. Même si les administrateurs l'utilisent pour protéger et fortifier leurs systèmes informatiques, elle ne représente pas une solution unique et définitive . En effet, diverses implémentations des protocoles de sécurité se sont révélées vulnérables. De plus la sécurité peut être rompue via plusieurs types d'attaques. Par exemple l'homme du milieu (MITM) constitue une menace lors des créations des clés. Par ailleurs les mots de passe courts et simples utilisés comme des clés de sécurité par les algorithmes symétriques sont facilement cassables via des attaques par dictionnaires ou de recherche exhaustive. En outre la cryptographie empêche l'analyse aisée du contenu des paquets et rend donc difficile la détection des attaques si elles sont déjà insérées dans des protocoles réseaux. Elle constitue même un moyen de camoufler les attaques et par conséquent de contourner les pare-feux et les systèmes de détection d'intrusions.

Principe de fonctionnement de la cryptographie

Un algorithme cryptographique, ou chiffre, est une fonction mathématique utilisée dans le processus de chiffrement et de déchiffrement. Celui ci fonctionne en combinaison avec une clé – un mot, un nombre, ou une phrase – pour chiffrer le texte clair. Ce dernier se chiffre en un texte chiffré différent si l'on utilise des clés différentes. La sécurité des données chiffrées est entièrement dépendante de deux choses: la force de l'algorithme cryptographique et le secret de la clé.

Cryptographie conventionnelle

Dans la cryptographie conventionnelle, aussi appelée chiffrement à clé secrète ou à clé symétrique, on utilise une seule et même clé à la fois afin de chiffrer et de déchiffrer. A titre d'exemple, on cite le Data Encryption Standard (DES) qui est largement employé par le Gouvernement fédéral américain. La Figure 1.4 est une illustration du processus du chiffrement conventionnel [77].

Figure 1.4: *Chiffrement conventionnel*

Gestion de clé et chiffrement conventionnel

Le chiffrement conventionnel a des points forts. Il est caractérisé par sa rapidité et son utilité pour chiffrer des données qui ne vont aller nulle part. Neamoins, la dépendance sur le chiffrement conventionnel seulement en tant que moyen de transmission de données sécurisées s'avère onéreux à cause de la difficulté de la distribution sécurisée de la clé.

Afin qu'un expéditeur et un destinataire puissent communiquer d'une manière sûre en exploitant un chiffrement conventionnel, il est necessaire de se mettre d'accord sur une clé et la garder secrète entre les deux. S'ils se trouvent dans des lieux géographiques différents, ils doivent faire confiance à un messager ou à un autre moyen de communication pour empêcher la divulgation de la clé secrète pendant la transmission. Quiconque a entendu par hasard ou intercepté la clé en transit peut plus tard lire, modifier, et contrefaire toutes les informations chiffrées ou authentifiées avec cette clé.

La cryptographie à clé publique

Afin de resoudre les problèmes de distribution de clé, un concept fut inventé par Whitfield Diffie et Martin Hellman en 1975 appelé la cryptographie à clé publique.

Le concept de la cryptographie à clé publique se base sur un schéma asymétrique qui utilise une paire de clés pour le chiffrement : une clé publique, qui chiffre les données, et une clé privée correspondante, aussi appelée clé secrète, qui sera utilisée pour le déchiffrement (comme le montre la Figure 1.5). La clé publique sera disponible pour le grand public, contrairement a la clé privée que seul la personne concernée par le dechiffrement la connait. Toute personne en possession d'une copie de votre clé publique peut ensuite chiffrer des informations que vous seul pourrez lire.

Actuellement, Il est impossible de déduire la clé privée de la clé publique. Quiconque ayant une clé publique peut chiffrer des informations mais ne peut pas les déchiffrer. Seule la personne qui a la clé privée correspondante peut déchiffrer les informations.

Le principal point fort de la cryptographie à clé publique est qu'elle permet d'échanger des messages de manière securisée. La condition de partage des clés secrètes via un canal sûr n'est

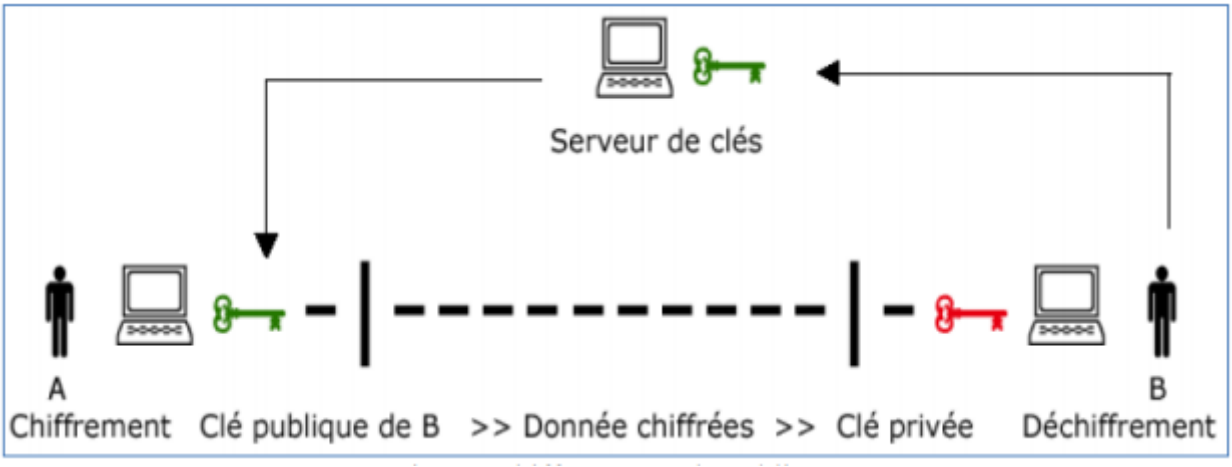


Figure 1.5: *La cryptographie à clé publique*

pas necessaire, toutes les communications impliquent uniquement des clés publiques, et aucune clé privée n'est jamais transmise ou partagée. Des exemples de cryptosystèmes à clé publique sont : Elgamal [48](du nom de son inventeur, Taher Elgamal), RSA [135](du nom de ses inventeurs, Ron Rivest, Adi Shamir, et Leonard Aldeman), Diffie-Hellman [45], et DSA , l'Algorithme de Signature Digitale (inventé par David Kravitz). Parce que la cryptographie conventionnelle était autrefois le seul moyen disponible pour transmettre des informations secrètes, le coût des canaux sûrs et de la distribution des clés réservait son utilisation uniquement à ceux qui pouvaient se l'offrir, comme les gouvernements et les grandes banques. Le chiffrement à clé publique est la révolution technologique qui permet aux masses d'accéder à la cryptographie forte [77].

1.6.5 Pots de miel

Ce type de protection est considéré comme une machine présentant ou simulant des failles de sécurité très répandues. Disposant de moyens renforcés de surveillance, la machine peut servir d'appât pour apprendre la stratégie des attaquants et construire des signatures exactes d'attaques. Par ailleurs la simulation du comportement d'une machine doit aussi être réaliste pour ne pas éveiller les soupçons des attaquants. Un pot de miel dispose de plusieurs outils de surveillance et d'archivage, nécessaires pour collecter les informations des activités suspectes. Ces outils doivent être maintenus en permanence puisqu'ils sont déployés dans un environnement fréquenté principalement par des attaquants. De plus, l'isolation du pot de miel du reste du réseau est indispensable pour qu'il ne se transforme pas en une base pour compromettre d'autres machines [54].

Historique des pots de miel

L'histoire des pots de miel remonte à la fin des années 80, lorsque le concept fut établi par Clifford Stoll dans [147]. Dans ces travaux, l'auteur décrit les manières d'observer et pister un intrus. Pour ce faire, il a imaginé un projet gouvernemental factice, avec des informations factices, pour que les intrus passent un temps non négligeable à télécharger et analyser ces fichiers. Dans les années

90, Cheswick implémenta et déploya un véritable pot de miel [34]. Ce n'est qu'en 2001 que le terme de pot de miel a été pour la première fois utilisé. Depuis, plusieurs auteurs ont proposé des définitions et des classifications. Le document [130] propose une terminologie et un aperçu des différentes technologies de pot de miel. Un réseau de miel (en anglais, honeynet) est un réseau de pots de miel. Un jeton de miel (en anglais, honeytoken) est un pot de miel inviolable. Les pots de miel peuvent se décliner en plusieurs catégories, en fonction du niveau d'interaction fourni. Dans la littérature, une classification en deux catégories est utilisée. Elle distingue les pots de miel basse interaction et les pots de miel haute interaction.

Les pots de miel basse interaction

Les pots de miel basse interaction exploitent des services réseaux pour piéger des programmes malveillants. Ces services ne peuvent donc pas être utilisés pour obtenir un accès au système. Ces types de pots de miel sont qualifiés par la simplicité de mise en œuvre. De plus, ils permettent le contrôle des comportements et gestes des attaquants. Ils collectent les informations circulées entre le programme malveillant et le service. Ensuite, on tire profit de ces informations pour identifier au plus tôt une activité malveillante. Parmi les outils les plus courants, nous pouvons en citer les moniteurs de ports tels que NukeNabber et Netcat. Ces implementations sont capables d'écouter sur un port et de journaliser les connexions et les informations qui viennent avec. Afin de minimiser les risques d'infection, seuls les premiers paquets des connexions sont récupérées.

Certains logiciels, tel que Specter, Fakebo et Deception toolkit, suggérent d'émuler des solutions peu securisées : du système de transfert de fichier (ftp), au système de gestion des courriers électroniques (pop). Ils trompent les attaquants en leur présentant les bannières de ces logiciels vulnérables afin de récupérer les débuts des interactions avec les attaquants.

L'outil Nepenthes [10] fournit un mécanisme intéressant pour traiter les attaques. Il simule des vulnérabilités afin d'inciter les attaquants à lui envoyer leurs maliciels. Il est constitué d'une base qui comprend des routines élémentaires de gestion des connexions, des fichiers,..., etc. Des modules peuvent lui etre ajoutes afin de détecter les vulnérabilités utilisées et les logiciels employés [5].

Les pots de miel haute interaction

Les pots de miel haute interaction simulent un système complet pour receuillir plus d'informations concernant l'attaquant. En effet, ce type de pot de miel laisse les attaquants pénétrer le cœur du système, ce qui attribue aux administrateurs le temps necessaire pour collecter le maximum d'informations sur l'intrus. Ces types de pots de miel constituent un excellent complément des pots de miel basse interaction. Leur inconvénient vient de l'attention importante à apporter au cloisonnement des activités des attaquants pour se protéger des risques d'utilisation du pot de miel pour attaquer d'autres machines (phénomène de rebond).

Parmi ces applications, nous pouvons distinguer deux catégories. La première est constituée d'implémentations qui modifient le comportement du système d'exploitation pour y ajouter les

mécanismes nécessaires à la collecte des données. Sebek [53] et Uberlogger [6] représentent des exemples de ces implémentations. Elles se reposent sur des systèmes d'exploitation Gnu-Linux. Les noyaux de ces systèmes ont été modifies afin d'y intégrer les mécanismes nécessaires pour intercepter les informations. Ces modifications ont aussi été apportées dans le but de ne pas être détectées par des attaquants [5]. La deuxième catégorie est constituée d'implémentations qui utilisent des outils tierces pour collecter des informations. Nous pouvons citer, à titre d'exemple, l'implémentation proposée dans [52], qui utilise le renifleur de réseau snort afin de collecter les échanges entre le pot de miel et les attaquants. Le contrôle des connexions est réalisé avec un pare-feu. Une autre implémentation [68] propose d'utiliser un outil de détection d'intrusion, Prelude, pour la collecte des informations.

1.7 conclusion

Les pirates adoptent un plan d'attaque bien etudié pour réussir leurs exploits. Ils tirent profit de plusieurs sources d'information et de divers outils pour compromettre le système informatique ou intercepter des données très sensibles. Par conséquent, les administrateurs déploient des solutions de sécurité efficaces capables de protéger le réseau de l'entreprise, tel que les pare-feux, les scanners de vulnérabilités, les outils d'archivage, la cryptographie et les pots de miel. Dans ce contexte, d'autre outils appelés systèmes de détection d'intrusions constituent une solution complémentaire pour mieux sécuriser le réseau informatique. Nous détaillons dans le chapitre suivant les qualités nécessaires des systèmes de détection d'intrusions. Nous discutons aussi des approches proposées dans la littérature à savoir la détection comportementale et la détection par scenarios.

CHAPITRE

2 Les Systèmes de Détection d'Intrusions

> *"People often represent the weakest link in the security chain and are chronically responsible for the failure of security systems"*
>
> BRUCE SCHNEIER, SECRETS AND LIES

2.1 Introduction

Les intrusions électroniques représentent un phénomène critique de l'informatique moderne, faisant de sorte que même à plusieurs kilomètres quelqu'un peut accéder à vos ressources et données, à votre insu, et les manipuler à sa guise. Face à ce danger, l'approche préventive, qui consiste à définir une politique de sécurité et de déployer les outils pour l'appliquer, n'est plus, désormais, suffisante. La nécessité de détecter toute tentative de violation de la politique de sécurité s'impose. Autrement dit le recours à un outil de détection automatique des intrusions (IDS, Intrusion Detection System) s'impose.

Suite à cet intérêt, ce type d'outils de sécurité connait de nos jours un essor important et constitue un investissement des entreprises. Les IDS sont déployés dans des zones précises du réseau ou sur des machines particulières pour compléter le travail des pare-feux et détecter les attaques passées inaperçues comme le montre la Figure 2.1. Considérés comme une dernière barrière de sécurité, les IDS sont capables de comprendre la nature et les caractéristiques du trafic réseau afin de mieux détecter les intrusions. Ce chapitre est consacré à la présentation des détails concernant ces outils. Dans un premier temps, nous donnons quelques définitions concernant les IDS. Ensuite, nous présentons les différents classes auquel l'IDS peut appartenir. Ensuite, nous dresserons un état de l'art sur les différentes techniques utilisées par un système de détection d'intrusion : l'approche par scenarios et l'approche comportementale. Dans un deuxième temps, nous mettons en lumière les limites de l'approche comportementale. Pour conclure, on donne une liste de quelques IDS opérationnels.

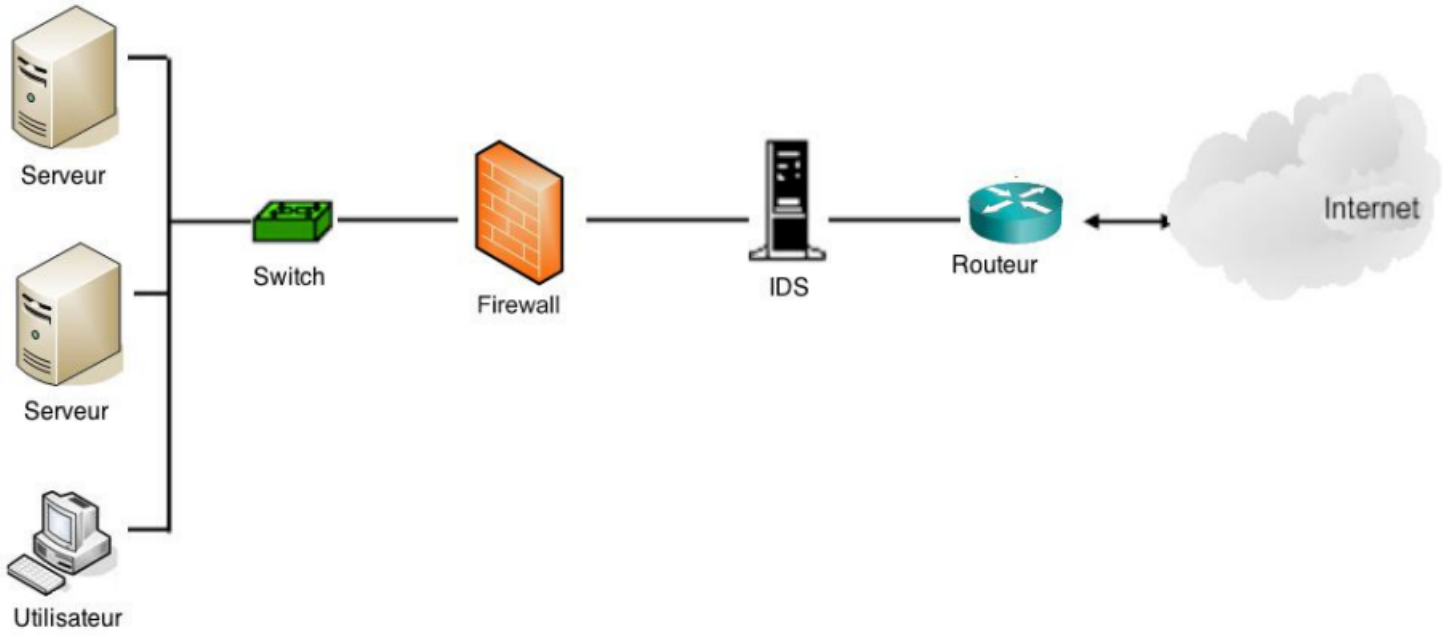


Figure 2.1: *Emplacement d'un IDS*

2.2 Définitions

Le concept de système de détection d'intrusions a été introduit en 1980 par Anderson [8]. Mais le réel départ était marqué par la publication d'un modèle de détection des intrusions par Denning en 1987 [44]. Au préalable, il est utile d'expliquer certains concepts pour pouvoir définir par la suite ce qu'est un système de détection d'intrusions.

1. **Mécanisme d'audit de sécurité** : C'est l'enregistrement des actions effectuées par les utilisateurs sur un système informatique, pour en faire une analyse ultérieure permettant de démasquer les auteurs d'intrusions.

2. **Journal d'audit** : C'est l'ensemble des informations générées par les mécanismes d'audit.

3. **Flux d'audit élémentaire** : C'est une suite temporelle reportant les événements se produisant sur une même partie du système et traduisant son comportement.

4. **Flux d'audit** : C'est une suite temporelle d'événements, pouvant être le mélange de plusieurs flux élémentaires.

5. **Détection des intrusions** consiste, alors, à analyser les informations collectées par les mécanismes d'audit de sécurité, à la recherche d'éventuelles attaques ou intrusions. Les systèmes de détection des intrusions peuvent revêtir plusieurs formes:

   (a) Outils de diagnostic : Ces produits scrutent le réseau à la recherche des points de vulnérabilités connus, de part les ordinateurs et tous les périphériques servant à acheminer

les données (routeurs, concentrateurs et commutateurs). Ces outils peuvent également tester le pare-feu et générer un rapport complet sur l'état du réseau et suggérer les actions à entreprendre pour colmater les brèches recensées.

(b) Anti-renifleurs : Ces outils permettent d'évaluer les ordinateurs individuels d'un réseau afin de dépister ceux qui sont susceptibles d'être dans une situation de " promiscuité électronique", c'est-à-dire pouvant être manipulés par un logiciel renifleur pour épier l'ensemble des activités du réseau.

(c) Détecteurs d'attaques : Ces logiciels servent à sonner l'alarme dès que survient une attaque électronique connue.

(d) Les leurres : Ces produits simulent un serveur présentant des failles connues. Ces serveurs factices attirent ainsi l'attention des véritables serveurs critiques.

2.2.1 L'audit de sécurité

J. P. ANDERSON [8] a été le premier à montrer le rôle crucial que joue l'audit de sécurité dans un processus de sécurisation d'un système informatique. La reconstitution des opérations entreprises par certains utilisateurs spécifiques du système faite à partir de l'étude des séquences d'événements contenues dans le journal d'audit doit permettre de répondre aux six questions suivantes [115] :

1. Quelle opération a été exécutée?
2. Qui a exécuté l'opération ?
3. Quelles ressources du système ont-t-elles été affectées par l'opération?
4. Quand l'opération a-t-elle été réalisée?
5. Où l'opération est-elle arrivée? Par exemple, dans le cas où l'opération a conduit à un enregistrement sur un serveur distant, on enregistre l'identifiant de ce serveur.
6. Quelles sont les raisons qui ont fait qu'une opération a échoué?

La réponse à ces questions nécessite la spécification, préalable, des utilisateurs et des activités système à auditer, la garantie de la collecte des événements dans le fichier d'audit et l'analyse régulière de ce fichier. Enfin il est nécessaire de prévoir la réparation des dégâts éventuellement détectés.

2.2.2 Spécification des activités système à auditer

Les activités systèmes sont diverses et n'ont pas toutes le même niveau d'importance en matière de sécurité. Au vue du niveau de sécurité souhaité, on peut avoir cette liste d'informations pertinentes à auditer:

1. **Informations sur les accès au système**

   L'identifiant de l'utilisateur ou du processus, l'horodatage de l'accès au système, l'identifiant du terminal, l'adresse du site cible de l'opération et le mode d'entrée (interactif,batch local, connexion distante) constituent des informations utiles dans le sens où elles peuvent nous renseigner sur le début d'un processus de violation éventuelle de la sécurité.

2. **Informations sur l'usage fait du système**

   Ce sont des informations concernant des ressources systèmes qui ont été utilisées. Elles décrivent aussi comment ces ressources ont été exploitées.

3. **Informations sur l'usage fait des fichiers**

   Ces informations relèveront des détails tels que l'horodatage de l'accès, source de l'accès (utilisateur, terminal ou application), type de l'accès (ouverture, fermeture, lecture, modification, purge du fichier, etc), volume d'informations échangées lors de l'accès.

4. **Informations relatives à chaque application**

   On s'intéresse aux événements produits par toute application et qui peuvent avoir une influence sur la sécurité du système. Parmi les événements on pourra enregistrer [115] :

   (a) Les commandes exécutées et leurs résultats,

   (b) Les lancements et arrêts d'application,

   (c) Les modules réellement exécutés,

   (d) Les données entrées,

   (e) Les sorties produites.

5. **Informations sur les violations éventuelles de la sécurité**

   C'est des informations relatives à tous les événements pour lesquels il y a eu une tentative d'accès illégale par rapport à la politique de sécurité en vigueur. Parmi ces événements, on peut citer [115] :

   (a) Le changement des droits d'accès à des fichiers sensibles,

   (b) La tentative d'exécution de certaines commandes du système réservées à des utilisateurs privilégiés.

   (c) La tentative d'accès à un fichier non autorisée ou la fourniture d'un mot de passe erroné pour cet accès.

   (d) La tentative d'exécution d'une application dans un mode privilégié (par exemple la modification des droits d'accès sous Unix),

   (e) L'accès au système à des moments ou depuis des lieux inhabituels.

6. **Informations statistiques sur le système**

   Les remarques de tout excès ou absence d'un événement d'une manière inhabituelle (le niveau anormalement élevé des refus d'accès au système, le niveau anormalement élevé ou bas de l'usage de certaines commandes du système) sont utiles pour pouvoir tirer des conclusions en matière de sécurité [115].

2.2.3 Collecte des événements

La plupart des systèmes d'exploitation disposent d'un sous-système d'audit capable de générer certains types d'événement. Le noyau du système assure alors la génération et la collecte de ces événements [115].

2.2.4 Analyse du journal d'audit

Analyser le journal d'audit revient à identifier toute violation des règles de la politique de sécurité. Explicitement l'analyse permet de déterminer les responsables, identifier les dégâts éventuels, tenter de les réparer et proposer des plans d'action qui assurent la sécurité du système. L'analyse du journal d'audit nous permet de détecter des actions malveillantes portant atteinte à la confidentialité (vol de données, inférence illégitime, etc) ou à l'intégrité (modification non autorisée des fichiers de données) ou à la disponibilité (utilisation illégitime ou abusive des ressources, destruction illicite ou abusive de fichier, réduction illégale des droits des utilisateurs,...etc) [115].

Les actions qui peuvent être entreprises à l'encontre d'une intrusion en cours, sont la déconnexion de l'intrus ou son confinement dans des répertoires particuliers en vue de réduire les dégâts possibles.

2.2.5 Fréquence de l'analyse des traces d'audits

Il n'est plus suffisant aujourd'hui de faire des analyses journalières. Pratiquement, il est d'usage d'envisager un écart de quelques minutes et même de quelques secondes entre deux analyses successives. En fait, les analyses doivent être fréquentes afin que le minimum de prévarications reste indétecté [115].

2.2.6 Protection du journal d'audit

Le journal d'audit constitue la base et le point de départ de l'analyse faite en vue de détecter tout comportement déviant des règles de sécurité. Pour cette raison, veiller à la confidentialité, l'intégrité et la disponibilité de ce journal est une pierre angulaire si on veut tirer des conclusions utiles et cohérentes à partir de son contenu [115].

2.2.7 L'audit dans le cas des réseaux

Lorsqu'on est amené à faire de l'audit sur un système distribué, il est impératif de disposer d'une base de temps commune qui permettra de gérer les événements survenant sur toutes les machines

connectées. En outre, il faut être capable de diffuser les types d'événements à auditer et de rassembler les journaux d'audit en provenance des différentes machines du réseau, de manière à constituer un journal d'audit global. Des considérations de sécurités doivent être prises pour ces transferts de données [115].

2.3 Classification des IDS

On peut classifier les systèmes de détection d'intrusions en se basant sur leurs emplacements dans le système informatique, et leurs sources de données en trois grandes familles distinctes : les N-IDS, les H-IDS et les IDS hybrides. Ces derniers seront décrits dans les sous sections suivantes.

2.3.1 Les N-IDS (Network Based IDS)

Les NIDS (Network Based Intrusion Detection Systems) sont probablement les systèmes les plus connus. Ces systèmes pouvant être assimilés à un sniffer, capturent et décodent toutes les trames qui transitent par le segment sur lequel il est connecté. Toutefois, contrairement à un sniffer, cette sonde analyse les paquets IP dans leur intégralité afin de repérer des signatures d'attaque déjà connues ou des anomalies dans les entêtes des paquets. Parmi ces avantages on cite:

1. La possibilité de surveillance d'un grand réseau.
2. Son déploiement a peu d'impact sur un réseau existant. En effet, les NIDS sont habituellement des dispositifs passifs qui écoutent sur un fil de réseau sans interférer l'opération normale d'un réseau.

Ce type d'IDS presente quelques limites tel que :

1. L'incapacité d'analyser des informations chiffrées (cryptées). Ce problème a lieu dans les organisations utilisant le VPN.
2. La provoquation des paquets en fragments.

2.3.2 Les H-IDS (Host Based IDS)

Les HIDS (Host Based Intrusion Detection Systems) représentent le complément naturel des NIDS. Installer des HIDS sur les machines à protéger revient à les pourvoir d'agents logiciels qui offrent trois services:

1. Détection d'attaques contre des applications installées sur le système protégé,
2. Vérification de l'intégrité des fichiers sensibles,
3. Corrélation des fichiers journaux en provenance d'applications ou d'équipements tiers tels les routeurs, firewalls, commutateurs.

Les HIDS utilisent des informations d'audit collectées, provenant essentiellement de la machine à protéger à partir de diverses sources (traces d'audit système, historique des commandes exécutées, etc).

2.3.3 Les systèmes de détection d'intrusions hybrides

Généralement utilisés dans un environnement décentralisé, ils permettent de réunir les informations de diverses sondes placées sur le réseau. Leur appellation « hybride » provient du fait qu'ils sont capables de réunir aussi bien des informations provenant d'un système HIDS qu'un NIDS.

2.4 Testabilité des systèmes de détection d'intrusions

Malgré le fait que les systèmes de détection d'intrusions sont devenus les outils de défense omniprésents dans les systèmes informatiques d'aujourd'hui, jusqu'à présent on n'a aucune méthodologie complète et scientifiquement rigoureuse pour examiner l'efficacité de ces systèmes [116]. Dans cette section on va donner un ensemble de mesures partielles qui peuvent être utilisées pour tester le rendement des IDS. On va se concentrer sur des mesures quantitatives pour mesurer l'exactitude de la détection.

2.4.1 Taux de Faux Positifs

Cette notion concerne principalement le taux des faux positifs produits par un IDS dans un environnement donné pendant une période de temps particulière. Un faux positif, ou fausse alarme, est une alerte provoquée par le trafic en absence d'attaque. Sachant qu'il n'existe pas de réseau « standard », il sera difficile de déterminer les aspects du trafic réseau ou les activités de l'hôte qui causent de telles alertes. Pour essayer de réduire le taux des faux positifs, on joue sur la multitude de configuration des IDS. Autrement dit, on adopte la configuration d'IDS qui produit un minimum de faux positifs. Par ailleurs, ce type de manipulation ne permet pas une solution définitive. En general, pour calculer le taux des faux positives (FPR), la communauté scientifique se base sur l'equation :

$$FPR = \frac{FP}{FP + TN} \times 100 \tag{2.1}$$

Tel que FP (False Positives) sont considérées comme le nombre des connexions normales mal classées , et TN (True Negatives) correspondent au nombre de connexions normales classées correctement.

2.4.2 Taux de Détéction

Cette notion concerne le taux d'attaques détectées correctement par un IDS dans un environnement donné pendant une période particulière. Cette mesure montre à quel point un IDS peut identifier l'attaque qu'il a détecté en l'affectant à une catégorie, et en la marquant par un nom.

Pour calculer le taux de Détéction (DR), la communauté scientifique se base sur l'equation :

$$DR = \frac{TP}{TP+FN} \times 100 \tag{2.2}$$

Sachant que TP (True Positives) représente le nombre des intrusions classés correctement et FN (False Negatives) est le nombre des intrusions mal classées.

2.4.3 Résistance aux attaques

Cette mesure montre à quel point un IDS peut résister aux tentatives d'attaques. Parmi les formes des attaques contre un IDS on peut citer :

L'envoie d'un trafic volumineux excèdant les capacités du traitement d'un IDS. Avec un tel trafic, l'IDS plantes et risque de ne pas fonctionner correctement.

L'exploitation d'une vulnérabilité existante dans l'IDS. De telles attaques seront réussies seulement si les IDS sont mal developpés et contiennent beaucoup de bugs.

Parmi les mesures qu'un IDS doit entreprendre pour remédier à ces attaques, le fait de pouvoir corréler les événements d'une attaque. Ces événements peuvent être recueillis des routeurs, des pare-feux, des applications logs, ou d'une large variété d'autres dispositifs. Cette corrélation permet aussi d'identifier les attaques de pénétration par étapes.

Actuellement, les IDS ont seulement des aptitudes limitées dans ce domaine. Une autre mesure est la capacité de déterminer le succès d'une attaque qui est essentielle pour l'analyse de la corrélation et du scénario de l'attaque. Cela va également simplifier considérablement le travail d'un analyste en distinguant les attaques réussites qui sont les plus importantes et les attaques échouées habituellement moins préjudiciables.

2.4.4 Capacité de manipuler le trafic réseau

Cette mesure montre à quel point un IDS fonctionnera en présence d'un trafic volumineux. La plupart des N-IDS commenceront à ignorer les paquets arrivés à mesure que le volume du trafic augmente, par consequent, certaines attaques risquent de ne pas être détectées. Cette mesure est presque identique à « la mesure de résistance aux dénis de service » quand l'attaquant envoie un grand volume de trafic aux IDS. La seule différence est que cette mesure calcule l'aptitude de l'IDS à manipuler les volumes particuliers par rapport au fond normal du trafic.

Il est à noter aussi qu'un N-IDS peut être éludé par des versions furtives des attaques. Les techniques utilisées pour rendre les attaques furtives incluent la fragmentation des paquets, l'utilisation de divers types de codage des données en utilisant des drapeaux TCP inhabituels des paquets d'attaques cryptés, l'extension des attaques à travers des sessions multiples du réseau, et le lancement des attaques à partir des sources multiples [133]. Pour remédier à ces attaques, un N-IDS exige des capacités d'inspection dans des niveaux plus profonds des paquets du réseau. Par conséquent, il est important de mesurer les capacités d'un N-IDS à capturer et à traiter, avec le même niveau d'exactitude, sous une charge réseau donnée, comme pour le cas d'un réseau passif. Pour

cela, Hall et al. [64] ont proposé une méthodologie de test et des métriques concernant le trafic pour normaliser les tests d'efficacité des N-IDS.

2.5 Les méthodes de détection d'intrusions

Trois familles de méthodes ont été proposées, à ce jour. La première consiste à observer le trafic sur le réseau, généralement en observant le contenu des paquets de données qui circulent d'une machine à l'autre, afin de détecter une signature d'attaque connue. Celle ci est connue sous le nom d'approche par scenarios. La seconde approche appelée approche comportementale consiste à définir un comportement normal du système et à rechercher tout ce qui dévie de ce comportement. La troisième technique se base sur les agents mobiles pour detecter l'intrusion.

2.5.1 Approche par scénarios

la construction d'une base de motifs représentant des signatures d'attaque connus au préalable est considérée comme le pilier de l'approche par scénarios. On emploi cette base de signatures, le plus souvent en temps réel, sur les informations fournies par les sondes de détection. Ce type d'approche peut être défini comme un système de reconnaissance de motifs permettant de mettre en évidence à partir de ces informations la présence d'une intrusion connue de la base de signatures.

En general, il existe deux raisons qui favorisent le choix d'utilisation de cette approche. Ces mécanismes de reconnaissance sont souvent peu consommateur de ressources et la pertinence de la détection de ce type d'approche est élevée. Cette démarche est semblable à celle utilisée par les mécanismes de détection des programmes malveillants. Par exemple, l'outil Hancock [62] propose d'extraire automatiquement à partir d'un ensemble de programmes malveillants des séquences d'octets ayant une très faible probabilité d'apparaître dans d'autres programmes.

Le langage de description d'attaque [47, 101, 131] joue un grand rôle dans la création des signatures d'intrusions. Leur création relève le plus souvent d'une tâche manuelle. Toutefois, des méthodes ont été proposées pour générer ces signatures de manière automatique [117, 122]. Le taux de couverture de ce type d'approche repose principalement sur la complétude de la base de signature ainsi que sur la qualité des motifs qui y sont contenus. En effet, le plus souvent une signature est associée à une attaque particulière. Par consequent, seules les attaques dont la signature est présente dans la base sont détectées. Par ailleurs, l'approche par scenarios présente deux inconvénients majeurs.

D'une part, l'approche n'arrive pas à identifier des attaques récemment conçues. Cela signifie qu'entre le moment où une nouvelle attaque est developpée et le moment où celle-ci est connue, le système d'information est vulnérable malgré la présence d'un mécanisme de détection. Notons qu'il est possible de chercher à généraliser les signatures afin qu'il ne suffise pas de légèrement modifier une attaque pour réussir à contourner le mécanisme de détection. Toutefois, cette généralisation des signatures risque de faire augmenter le taux de faux positifs [132]. Notons aussi qu'une

autre manière de généraliser une signature n'est pas de décrire l'exploitation de la vulnérabilité mais la vulnérabilité elle même [26].

D'autre part, la base de signatures doit être régulièrement mise à jour. Par consequent, il faut que les personnes chargées de la maintenance veillent en permanence à ce que chaque nouvelle attaque puisse avoir sa signatures correspondante dans la base de données. Dans ce cas, le système d'information demeure vulnérable entre le moment où une nouvelle attaque apparait et le moment où la base est mise à jour.

Langages de description d'attaques

La connaissance d'activités caractéristiques de chaque intrusion joue un grand rôle dans la détection par scenarios. En effet, chaque activité correspond à un évènement observable du système. Cependant, pour déchiffrer une attaque, il faut pouvoir analyser ces différents évènements. Pour cela, il faut avoir recours à des langages adaptés à cette tâche.

Plusieurs langages ont été proposés afin de se focaliser sur l'exploitation des failles. Ils sont capables de décrire minutieusement les étapes nécessaires à l'exploitation d'une faille particulière [47, 162]. Pour reussir leur exploits, certaines attaques ont besoin d'un certain nombre de conditions (type de matériel ciblé, version du système d'exploitation, version de l'application, etc.). Cependant, il est important de pouvoir préciser l'état du système avant la réalisation de l'attaque. Des langages permettent explicitement d'exprimer ce genre de conditions [40, 102]. D'autre langages choisissent de se concentrer sur les évènements observables au niveau du système. Par exemple, c'est notamment le cas des signatures des projets ORCHIDS [61, 125] et GnG [158]. Des évènements tels que les appels système ou l'activation de certaines fonctions du noyau sont utilisés pour décrire les attaques. Par la suite, on tire profit de ces évènements pour construire des automates décrivant les états et les transitions du système qui reconnaîssent la présence d'une intrusion en cours de réalisation. De plus, pour les évènements qui possèdent des paramètres, il est possible d'utiliser des opérateurs sur ces paramètres pour simplifier la description de l'attaque mais aussi pour exprimer des contraintes complexes sur ces derniers [132].

Découverte de signature d'attaques

Des articles ont essayé d'identifier automatiquement des signatures d'attaques [117]. Ces travaux réussissent à isoler un type de comportement bien précis dans les applications. Pour y parvenir, cette approche repose sur l'utilisation d'une machine à états finis pour apprendre le type de comportement d'un groupe d'applications similaires.

Le but de la machine à états finis est de savoir si ce groupe contient une application donnée ou non. A titre d'exemple, en executant la phase d'apprentissage sur un ensemble de logs contenant les traces d'exécution de plusieurs navigateurs web, l'automate est ensuite capable de reconnaître la présence d'un navigateur web (qui ne faisait pas partie de la phase d'apprentissage) parmi un ensemble quelconque de traces d'exécution. En suivant cette philosophie, l'automate devient alors

capable de détecter sur d'autres applications une attaque similaire bien que celle-ci soit encore inconnue au moment de l'apprentissage.

2.5.2 Approche comportementale (Anomaly Detection)

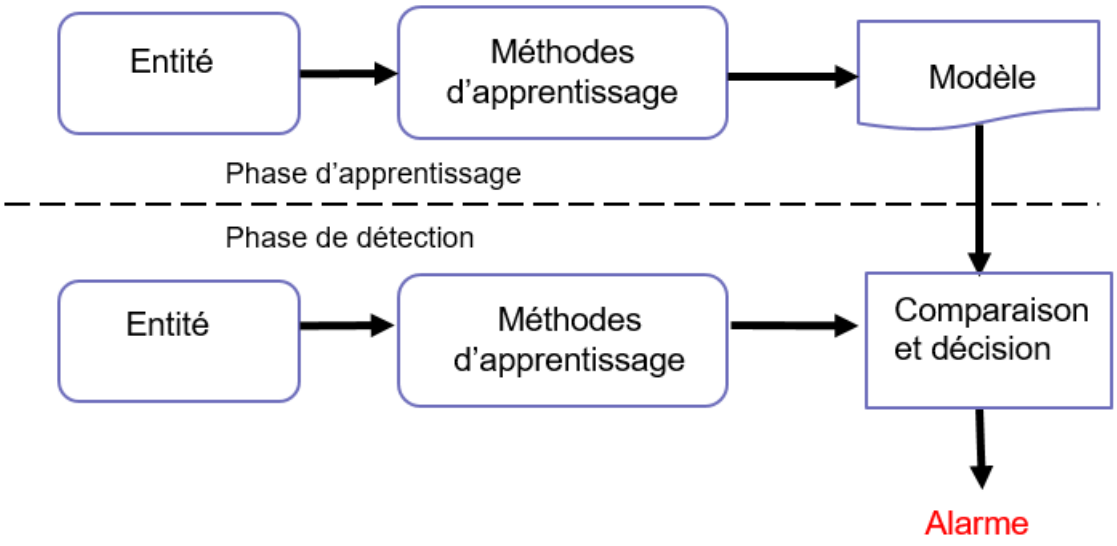


Figure 2.2: *Approche comportementale*

Cette approche repose sur une phase d'apprentissage et une phase de détection (Figure 2.2). Durant la première phase, on vise la création d'un modèle de référence représentant le comportement de l'entité surveillée (trafic réseau, utilisateur, ressource materiel,...) en situation de fonctionnement normal. Dans la deuxième phase, ce modèle est utilisé afin de pouvoir mettre en évidence d'éventuelles déviations comportementales.

Une déviation suffisamment grande (notion de seuil) par rapport à ce modèle de comportement normal conduit à une levée d'alerte. Cette approche se repose sur le fait que chaque comportement n'appartenant pas au modèle de comportement normal est considéré comme une intrusion.

Les approches comportementales se caractérisent par la détection des attaques fraichement conçues et encore inconnues au moment de la modélisation. En effet, le modèle de détection est construit non pas dans le but de caractériser les attaques à l'origine des intrusions mais dans le but de caractériser les déviations comportementales engendrées par les intrusions. De ce fait, une intrusion inconnue au moment de la création du modèle de référence est tout de même détectée tant que celle-ci produit une déviation observable par rapport au modèle de référence.

Un algorithme d'apprentissage est souvent exploité par l'approche comportementale pour construire le modèle de référence d'une part. D'autre part, cet algorithme évalue la différence entre le comportement de référence et celui observé durant la phase de détection. Le plus souvent, ces deux points sont liés : l'évaluation de la déviation est assimilée au type de modélisation sur lequel se base le mécanisme de détection. La précision du comportement de référence, c'est-à-dire sa

propension à présenter des faux-négatifs ou à émettre des faux-positifs, va donc dépendre de la méthode choisie pour la phase de construction du modèle [43].

Dans les travaux antérieurs, plusieurs approches ont été proposées pour construire ce modèle de référence. Les sous sections suivantes abordent quelques unes.

Le système expert

Ce mécanisme a la capacité de répondre à une question concernant plusieurs faits. Cet outil essaye de simuler la manière de penser d'un expert du domaine auquel se rapporte la question. Afin d'achever ce but, le mécanisme prend en considération un ensemble de règles spécifiquement écrites pour une question donnée. Ces règles sont ensuite vérifiées sur le jeu de faits fourni à l'aide d'un moteur d'inférence. Il faut vérifier si le jeu de faits fourni respecte ou non les règles pour répondre à la question. Sans oublier qu'à partir d'un jeu de faits initials, pour une question donnée on peut déduire automatiquement un jeu de règles. Ce type d'approche peut être utilisé pour analyser automatiquement de l'information. Il peut par exemple etre trés utile dans la classification des comportements chez les utilisateurs d'un système informatique [160]. Dans ces travaux, les sondes du système informatique sont utilisées pour collecter un ensemble d'informations sur les comportements des utilisateurs normaux. Puis, l'ensemble de ces informations est utilisé comme jeu de donnée initial pour la déduction des règles. Enfin, durant la phase de détection, le système expert ainsi obtenu est utilisé sur les informations fournies par les sondes du système informatique. Ce dernier permet ainsi de détecter les comportements inhabituels d'un utilisateur alors jugé comme malveillant.

Le modèle statistique

Ce modèle peut être utilisé pour décrire mathématiquement un mécanisme observé. Généralement, les observations ne permettent que d'obtenir une description approximative. Pour cela, la valeur de certaines observations sont considérées comme des variables aléatoires. Pour chacune de ses observations, un modèle statistique est utilisé pour décrire l'ensemble de distributions de la variable aléatoire correspondante.

Un modèle de comportement normal de ce système peut sans probleme etre établi si on travaille avec les observations effectuées par les sondes d'un système informatique durant une période de fonctionnement supposé normal. Une fois le modèle de référence ainsi établi, on a la capacité de détecter une anomalie en mesurant la distance entre ce modèle et de nouvelles observations provenant des sondes du système si on applique cette approche aux observations effectuées par les sondes d'un système informatique durant une période de fonctionnement supposé normal. Une fois le modèle de référence ainsi établi, on a la capacité de détecter une anomalie en mesurant la distance entre ce modèle et de nouvelles observations provenant des sondes du système [43].

Cette approche statistique a été appliquée notamment à la détection de comportements anormaux chez les utilisateurs d'un système informatique. Ces mécanismes de détection ciblent des

paramètres du système aussi variés que les tentatives d'authentification, la durée des sessions, le temps processeur utilisé, etc.

les modèles de Markov sont considérés comme des modèles statiques. Les travaux en détection d'intrusion qui reposent sur ses modèles [59] ont montré que ce type de modélisation permet d'obtenir un taux de détection satisfaisants. Notons également que des travaux ont cherché à utiliser ce type de modélisation pour détecter spécifiquement des élévations illégales du niveau de privilèges [36].

Les algorithmes génétiques

Ils s'agissent d'algorithmes évolutionnistes dont le but est d'obtenir, pour un problème qui ne possède pas de méthode exacte, une solution approchée. Ce type d'algorithme repose sur la notion de sélection naturelle. Pour cela, une population initiale est générée aléatoirement. Puis, pour chaque génération, une métrique permet de déterminer quel sous-ensemble de la population actuelle sera utilisé pour engendrer la génération suivante. La solution approchée est obtenue par améliorations successives. Ce type de mécanisme a été utilisé pour apprendre le comportement des utilisateurs d'un système informatique [11]. L'utilisation d'algorithmes génétiques dans ce contexte permet de prédire le comportement d'un utilisateur à partir d'un ensemble de ses actions passées. Par exemple, dans [11], l'historique des commandes exécutées par l'utilisateur est utilisé pour caractériser son comportement passé.

Les Réseaux de neurones artificiels

Ces methodes permettent de traiter de l'information selon une modélisation mathématique qui s'inspire du fonctionnement des réseaux de neurones biologiques. Ils sont composés de modèles simplifiés du neurone biologique appelés neurones formels. Ces derniers sont conçus comme des automates dotés d'une fonction de transfert qui calcule une valeur de sortie en fonction de ses entrées. Ils sont organisés en couche et s'interconnectent selon une topologie variable. Les paramètres des fonctions de transfert ainsi que la topologie d'interconnexion sont évolutifs, ils sont modifiables selon un procédé d'apprentissage. Ce type de modélisation peut être utilisé entre autre pour classifier automatiquement de l'information. En effet, si le jeu de données utilisé pour l'apprentissage est suffisamment grand, ces derniers présentent une capacité de généralisation. Cela nécessite toutefois de posséder a priori des informations déjà classifiées pour la phase d'apprentissage. De plus, si le réseau de neurones artificiels présente plusieurs entrées ou bien des connexions récurrentes, celui-ci a la capacité de traiter les informations en tant que série temporelle. Ce type de mécanisme a justement été utilisé pour modéliser le comportement des utilisateurs d'un système informatique [42]. Après un entraînement par les utilisateurs normaux du système, le réseau de neurones artificiels est capable de détecter la présence d'un utilisateur malveillant.

L'approche immunologique

Forrest [56] est parmi les pionniers à exploiter l'approche immunologique permettant de simuler les processus sur une machine. Sa méthode consiste à décrire le comportement normal via une séquence finie d'appels systèmes. Les séquences appelées N-gram servent de base pour comparer les appels systèmes des processus lors d'une phase de surveillance. Cette comparaison énumère les différences entre les paires dans une fenêtre de taille k (tide) [56] ou utilise des règles de r bits contiguës (stide) [69]. Wespi, Dacier et Debar [170] traitent une situation plus général en considerant les événements d'audit. Dans le but de modéliser l'état normal du système, ils produisent des séquences d'événements de tailles variables. Par la suite, un motif est choisi dans le cas ou ce dernier est suivi directement par d'autre motifs, sinon on ajoute 1 au score d'anomalie et une alarme est levée quand le score est plus grand que le seuil toléré.

la représentation des N-gram sous forme de graphes acycliques orientés (DAG) est optimisée par Marceau [112] pour réduire la base de profils définie par Forrest. De plus, cette optimisation permet d'utiliser le mécanisme de fenêtre glissante pour comparer les motifs. Kosoresow [82] analyse les caractéristiques des traces des appels systèmes et observe que les différences entre les motifs apparaissent dans des régions de tailles fixes. Lorsque la trace est divisée en trois parties, on arrive à créer de nouvelles séquences de motifs représentées en des machines à états finis. Cette approche réduit le nombre de séquences.

Warrender et Forest comparent dans [169] quatre approches immunologiques : la séquence simple d'événements (stide), la séquence d'événements menue des fréquences d'apparition (tstide), la génération automatique des règles inductives via RIPPER et le modèle caché de Markov (HMM). Ils concluent qu'en moyenne la modélisation HMM présente des meilleures performances. Mais il ne s'agit pas d'une supériorité absolue puisque les résultats des expériences dépendent des programmes testés.

le principal inconvénient de la modélisation HMM réside dans sa lenteur. Afin de résoudre cette carence, l'article [35] s'intéresse uniquement aux événements qui sont en relation avec le changement de privilège. Il réduit ainsi le nombre d'appels systèmes audités de l'ordre de 75% (de 267 à 80). Par ailleurs et afin de diminuer les faux positifs très répandus dans le cadre d'une détection d'intrusions comportementale, l'auteur combine plusieurs HMM et utilise diverses mesures dont les informations utiles sont raffinées via une méthode d'auto-organisation de données (SOM).

Le graphe de contrôle

Ce graphe peut être utilisé pour modéliser le comportement d'un système à partir du moment où celui-ci peut être décrit à l'aide d'un automate. Les nœuds du graphe représentent alors les différents états possibles du système et les arcs l'ensemble des transitions autorisées entre ces états. Le graphe est ensuite utilisé durant la phase de détection pour contrôler que le programme est toujours dans un état valide et que celui a été atteint au travers d'un chemin autorisé. Ce type d'approche a par exemple été appliquée sur des journaux conservant l'ensemble des connexions à un système d'information particulier [124]. Dans ces travaux il s'agit d'organiser sous forme de

graphes les données contenues dans les journaux de connexions.

Ce type d'approche a également été appliquée dans le cadre des systèmes diversifiés [109, 110]. Dans ces travaux, les graphes de flux d'information produits par plusieurs serveurs web différents qui répondent à une même requête sont comparés. Par ailleurs, la difficulté de comparaison des graphes rend l'application de l'approche impossible. En effet, il faut être capable de masquer le non-déterminisme propre à ce type d'information mais aussi être capable de masquer les différences entres les différents serveurs web. Par exemple, la séquence d'appel système nécessaire à l'envoi des données aux clients peut différer entre les serveurs web si ces derniers s'exécutent sur des systèmes d'exploitation différents. Généralement, l'avantage de ce mécanisme réside dans le traitement efficace des évènements se produisant rarement sur le système mais qui néanmoins sont valides vis-à-vis du comportement de l'entité surveillée. De nombreuses autres modélisations comportementales qui reposent sur l'utilisation d'un graphe de contrôle sont mises en œuvre par des mécanismes de détection de niveau applicatif [43].

2.5.3 Limites de l'approche comportementale

Malgré la proposition de plusieurs techniques afin de créer un modéle robuste, l'IDS comportemental contemporain souffre encore d'importants limites se résumant à ce qui suit:

1. La conception manuelle basée sur les connaissances des experts humains. Pour toutes les techniques utilisées dans les systèmes de détection d'intrusion, l'intervention de l'expert humain est indispensable dans les différentes étapes du développement de ces systèmes de détection d'intrusion. Le rôle de cet expert consiste généralement à spécifier les informations à auditer, établir le profil du comportement normal, spécifier l'encodage utilisé, suivre et mettre à jour les systèmes de détection d'intrusion. L'intervention des experts humains dans les différentes étapes de développement des systèmes de détection d'intrusion rend ces systèmes très associés aux experts et leurs performances dépendent de la compréhension des experts du système d'information et des attaques.

2. L'étroite dépendance de l'environnement cible.

   Vu que les systèmes de détection d'intrusion sont développés en étroite relation avec l'environnement cible, et que pendant ce développement on utilise généralement les spécificités de cet environnement cible rend la portabilité de ces systèmes de détection d'intrusion même pour des environnements similaire très difficile, voir impossible.

3. La difficulté liée à l'évaluation

   Pour évaluer les performances des systèmes de détection d'intrusion on utilise des données d'évaluation avec differents formats. Ce manque d'une base commune d'évaluation, d'un format standard pour présenter les traces d'audit représentent un obstacle qui nous empêche une vraie évaluation des différentes méthodes et une comparaison permettant de cerner les avantages et les inconvénients des IDS.

4. Les limites de performance.

   Face à la grande croissance du trafic réseau en volume, les systèmes de détection d'intrusion ont montré beaucoup de limites : la diminution importante de taux de détection, une consommation fulgurante de temps au niveau de la phase d'apprentissage et la difficulté de determination d'un seuil adéquat pour la prise de la décision (comportement normal ou pas).

2.5.4 Détection par agents mobiles

Chaque programme autonome pouvant se déplacer de son propre chef, de machine en machine sur un réseau hétérogène dans l'objectif de détecter et de combattre les intrusions est considéré comme un agent mobile. Ce dernier doit se caractériser par la capacité d'adaptation à son environnement, la communication avec d'autres agents, le déplacement et la protection. Pour ce dernier point, une des fonctions de l'agent doit être l'identification et l'authentification pour donner l'emplacement et l'identité de celui qui l'a lancé.

Les IDS basés sur les agents mobiles ou les Mobile Agents Intrusions Detection System (MAIDS) ont été introduit par [75] et mis en œuvre dans [12, 67]. D'autres articles plus récents ont adopté ce type d'IDS tel que [25, 142, 143].

Parmi les caractéristiques des (MAIDS) on cite :

1. La détection multi-point : celle ci est faite en analysant les flux d'audit de plusieurs hôtes pour détecter des attaques distribuées ou autres stratégies d'attaques d'un réseau dans sa globalité. Il est rarement possible de transporter tous les flux d'audit à un IDS central, et même dans ce cas, la détection est difficile. Les agents mobiles peuvent apporter le calcul distribué, pour le fait qu'on transporte l'analyseur au flux d'audit et non le flux d'audit à l'analyseur. Les agents pourraient détecter ces attaques, les corréler, et découvrir les stratégies d'attaques distribuées.

2. Une architecture résistante aux attaques : On utilise cette architecture pour obtenir plus de performances et pour centraliser le contrôle. Parmi les avantages de cette conception, on cite la non-redondance des lignes de communication. Les agents mobiles peuvent recuperer une branche de l'IDS si elle est coupée ou désactivé totalement. En general, l'architecture distribuée et hiérarchique offert par les agents mobiles permet a ces derniers de collaborer afin de ramener une fonctionnalité perdue ou detecter une activité suspecte.

3. La diversité génétique : On peut considérer les IDS à base d'agents mobiles comme une communauté d'entités autonomes. Chaque agent peut être programmé différemment ou manipuler de différents types de données. Cependant, s'ils ne partagent pas les mêmes données, leurs tests peuvent devenir prévisibles. Si chaque agent d'une même classe avait une façon différente de détecter la même chose, il serait autrement plus difficile de prévoir quoi que ce soit. Une manière de faire les choses serait de décrire ce qu'il faut détecter dans un langage standard et de laisser chaque instance de la classe trouver une manière

de le détecter. Les agents avec un faible taux de détection pourraient essayer de muter en introduisant de légères modifications dans leur façon de détecter l'attaque [19].

2.6 Installation des systèmes de détection d'intrusions

Lors du déploiement d'un IDS, il faut le configurer correctement en fonction de l'OS, des applications et du matériel utilisés, et il faut prendre en consideration l'impossibilité d'analyse de tout le trafic réseau. Afin de réduire la charge, Il est souvent préférable de placer l'IDS après le pare-feu du côté interne. Ainsi, seul les flux acceptés par le pare-feu seront analysés.

Le choix matériel a également une grande importance. Puisqu'un IDS doit pouvoir analyser le trafic réseau quelque soit sa destination, il devra recevoir tous les paquets. Mais le matériel réseau pose parfois des problèmes : L'utilisation des commutateurs simples (switchs) rend la conversation avec l'IDS impossible du fait que le switch commute les paquets directement au destinataire. Il est dès lors impossible d'installer une sonde qui analysera le trafic global. De plus, malgré que l'utilisation des concentrateurs (hubs) rende la conversation avec l'IDS possible car les concentrateurs répètent les paquets en les émettant à toutes les machines connectées, ces équipements sont peu fiables et sont donc à éviter. La solution sera donc l'utilisation des switches professionnels qui copie le trafic et l'envoie sur un port spécifié (où sera placé le N-IDS) : SPANport (Switch Port Analyzer). Un autre problème doit être pris en considération : il est impossible pour un IDS de décrypter les flux lorsque ces derniers sont cryptés (ex : par SSL, qui est le cas pour les VPN). Il faut donc utiliser un proxy SSL pour traiter ce problème.

Un autre point ne doit pas être oublié qui est la sécurisation du senseur et des logs d'alertes. En effet, un pirate pourrait très bien rendre l'IDS complètement inefficace, en visant ces deux composants. Pour sécuriser les sondes et les fichiers d'alertes, il est par exemple possible de mettre en place un réseau de management très contrôlé, avec son propre pare-feu et plusieurs mesures devront être prises pour assurer son fonctionnement tels la mise à jour régulière du système d'exploitation du senseur, la mis en place d'un système d'authentification robuste pour renforcer la sécurité, le changement régulier des mots de passe [23].

2.7 Exemples d'IDSs

Depuis le milieu des années 80 plusieurs IDS, mettant en œuvre de nombreux travaux s'inspirant du modèle d'Anderson, ont été commercialisés. IDES (Intrusion Detection Expert System) était le premier IDS hybride regroupant l'approche comportementale et celle par scénarios, utilisant à la fois les méthodes statistiques et les systèmes experts. Le prototype a été ensuite amélioré pour aboutir à NIDES (Next-generation IDES). NIDES et d'autres outils développés durant les mêmes années ont montré la possibilité d'utiliser les différents comportements d'utilisateurs sur une machine UNIX, pour détecter un éventuel essai d'intrusion. Cependant, si le comportement de l'utilisateur est trop riche, sa modélisation devient difficile et l'approche comportementale trouve sa limite.

Dans les années 90, l'approche par scénario a été fortement adopté par la plupart des IDS commerciaux pour plusieurs raisons : la simplicité de déploiement, la pertinence théorique, ainsi que la possibilité d'ajout, et de retrait des signatures.

les H-IDS ont été les premiers IDS employés, ils ont été exploités par les serveurs de calcul ayant une architecture UNIX avec des utilisateurs essentiellement locaux. Le serveur lui-même procurait les informations d'audit. De son cote, le N-IDS a connu un essor considérable depuis l'apparition d'Internet et l'insuffisance des sources d'audit locales pour détecter les nouvelles attaques réseau. Actuellement, la plupart des IDS commerciaux sont des N-IDS. Le tableau 2.1 montre quelques outils commerciaux et libres, ainsi que les approches de détection qu'ils utilisent.

Table 2.1: *Quelques outils d'IDS commerciaux et libres*

Type d'IDS	**Nom**	**Approche**
H-IDS	Asax (Audit UNIX) Gassata (Audit UNIX) Sutekh	Par Scénarios
N-IDS	BlackICE BRO Dragon ETrust (ex Session Wall) NetProwler NetRanger NFR Shadow Snort	Par Scénarios
	GrIDS	Comportementale
	Emerald	Comportementale+ Par Scénarios
IDS hybrides	Centrax (Audit NT) CyberCop Monitor (UNIX / NT) Intruder Alert (Audit UNIX) Real Secure (Audit NT)	Par Scénarios

2.8 Conclusion

Nous avons présenté tout au long de ce chapitre les qualités requises des systèmes de détection d'intrusions. Afin de remplir ces objectifs, diverses méthodes de détection d'intrusions ont été proposées. Elles se basent principalement sur trois types de détection : l'approche comportementale, l'approche par scenario et l'utilisation des agents mobiles. Nous avons expliqué ces trois principes de détection et avons souligné leurs limites. Le chapitre suivant abordera l'IDS proposé pour résoudre certains problèmes soulevés par l'approche comportementale.

CHAPITRE

3 Système de Détection d'Intrusions basé sur l'extraction des caractéristiques

> *"To find signals in data, we must learn to reduce the noise - not just the noise that resides in the data, but also the noise that resides in us. It is nearly impossible for noisy minds to perceive anything but noise in data"*
>
> STEPHEN FEW

3.1 Introduction

Nous avons présenté dans le chapitre précédent les difficultés principales que rencontrent les systèmes de détection d'intrusion comportementale. En général, ces solutions visent l'intégration des nouvelles méthodes de statistique et d'intelligence artificielle. Parmi les incovenients de ce type d'IDS, on souligne les limites de performances. En fait, les données du trafic peuvent être difficiles à analyser en raison de leur grande taille. Ce qui peut mener a une deffaillance de l'IDS.

Etre capable de réduire efficacement la taille des données est l'un des principaux défis de la sécurité réseau qui a été soulevée par de nombreux chercheurs[30, 80, 89].

Afin de pallier cette carence nous proposons dans ces travaux une approche de type NIDS reposant sur des algorithmes d'extraction de caractéristiques. Notre objectif est la minimisation de temps de réponse de l'IDS et des taux de faux positives d'une part. L'augmentation des taux de détection tout en identifiant les attaques inconnues d'autre part.

3.2 Architecture globale de l'IDS

Notre proposition d'IDS est basée sur l'architecture 3.1. L'entité etudiée sera le trafic réseau TCP/IP et l'IDS aura besoin d'une phase d'apprentissage et une phase de detection pour être évalué. Durant la première phase on génère un trafic réseau d'apprentissage et on lui applique un

algorithme d'extraction de caractéristiques. Par la suite, le nouveau trafic servira de modèle. Dans la phase de détection, on utilise un autre trafic réseau de test, on le libère de ces caractéristiques inutiles et on mesure sa déviation par rapport au modèle grâce à un algorithme de classification. Selon cette déviation, le trafic réseau de test sera déclaré intrusif ou non.

Dans les sections suivantes, on va décrire d'une manière détaillée chaque composant de cette architecture. A savoir, le type du trafic réseau, les phases d'apprentissage et les algorithmes d'extraction , ainsi que la phase de detection et les algorithmes de classification.

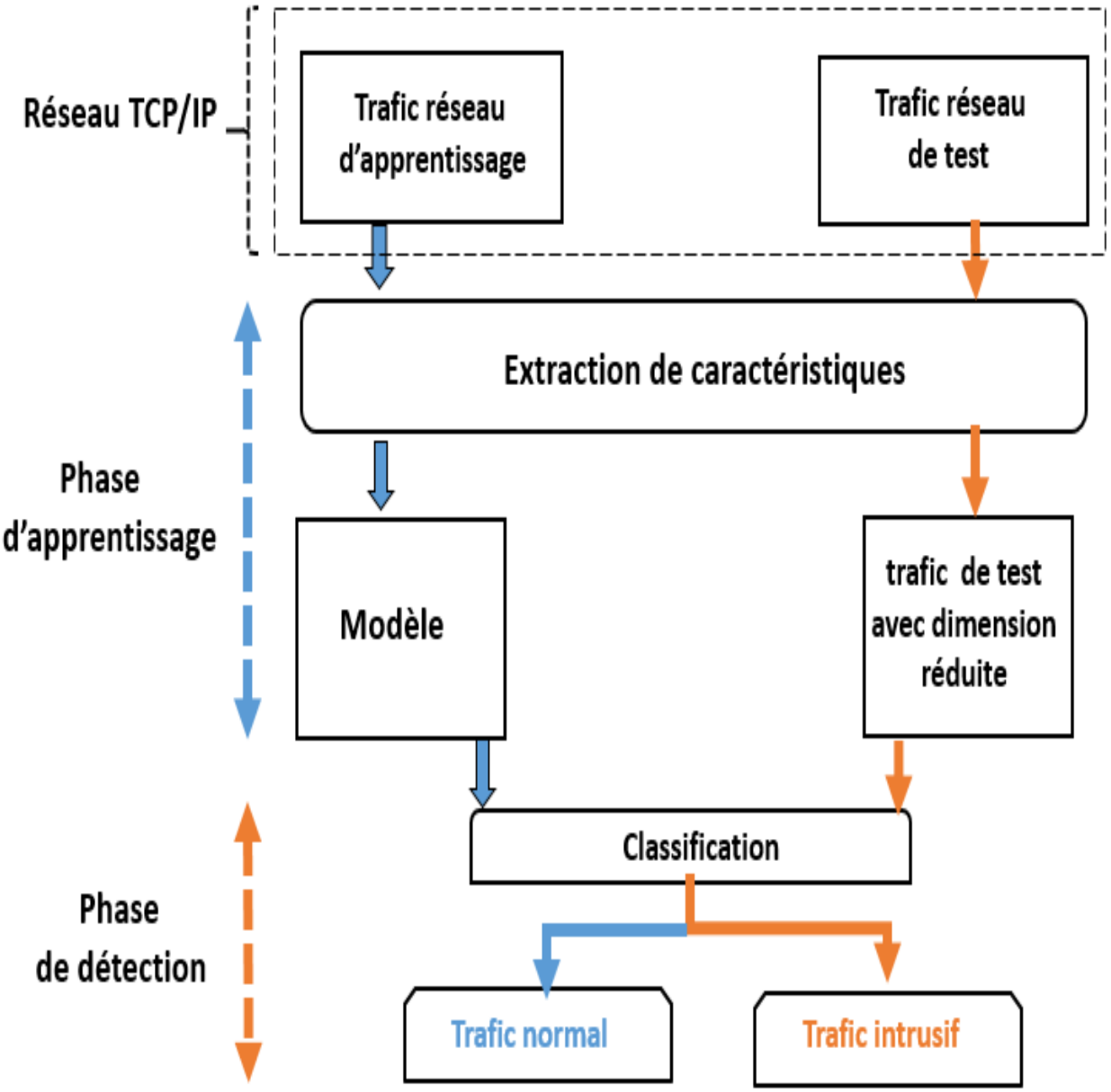


Figure 3.1: *Approche globale de l'IDS*

3.3 Le trafic réseau TCP/IP

Dans la fin des années soixante aux Etats Unis, un projet de réseau expérimental appelé ARPANET a été lancé par l'agence gouvernementale DARPA. Il fut construit dans le but d'étudier les technologies de communications, sans prendre en consideration le coté commerciale. Par la suite, plusieurs techniques de communication par modems se sont basé sur ce réseau pour implémenter des services de messagerie.

En 1975, le réseau passe officiellement du stade expérimental au stade opérationnel. Malheuresuement, le reseau manque d'uniformité.

Afin de pallier ce manque, en 1983, ARPANET adopte le protocole TCP/IP. Celui ci envahit les réseaux locaux eux-mêmes, car il est plus facile d'utiliser en interne et en externe.

Grace au TCP/IP, Internet a vu le jour et demeure maintenant un espace de communication qui englobe la planète tout entière. Des millions de sites partout sur la surface du globe y sont connectés.

Parmi les caractéristiques qui ont rendu le protocole TCP/IP populaire et indispensable on cite:

1. Le code source du protocole a été développé sans prendre en compte un environnement particulier ou une structure commerciale quelconque. De la vient la possibilité du transport du code source sur n'importe quel type de plate-forme.

2. Ce protocole ne prend en compte le support physique du réseau. Cela permet à TCP/IP d'etre supporté par des technologies aussi variés qu'une ligne série, un cable coaxial Ethernet, une liaison louée, un réseau token-ring, une liaison radio (satellites, "wireless" 802.11a/b/g), une liaison FDDI 600Mbits, une liaison par rayon laser, infrarouge, xDSL, ATM, fibre optique,..,etc.

3. Tous les utilisateurs de TCP/IP ont un mode d'adressage commun quelle que soit la plateforme qu'ils emploient. Si l'unicité de l'adresse est respectée, les communications aboutissent même si les hôtes sont aux antipodes.

4. Les protocoles de hauts niveaux sont standardisés ce qui permet des développements largement répandus et inter-opérables sur tous types de machines.

3.3.1 Modèle en couches

Pour que chaque machine puisse adopter le modèle TCP/IP, on a decomposé le système de protocoles TCP/IP en plusieurs modules, chacun exécute une tâche précise. De plus, ces modules effectuent ces tâches d'une manière ordonnée, on a donc un système stratifié, c'est la raison pour laquelle on parle de modèle en couches.

On a tendance d'utiliser le terme couche pour évoquer le fait que les données qui transitent sur le réseau traversent plusieurs niveaux de protocoles. Par consequent, les données (paquets d'informations) qui circulent sur le réseau sont traitées successivement par chaque couche, qui vient rajouter un élément d'information (appelé en-tête) puis sont transmises à la couche suivante.

Il existe une correspondance entre Le modèle TCP/IP et le modèle OSI (figure 3.2) qui a été mis au point par l'organisation internationale des standards (ISO, organisation internationale de normalisation) afin de normaliser les communications entre ordinateurs.

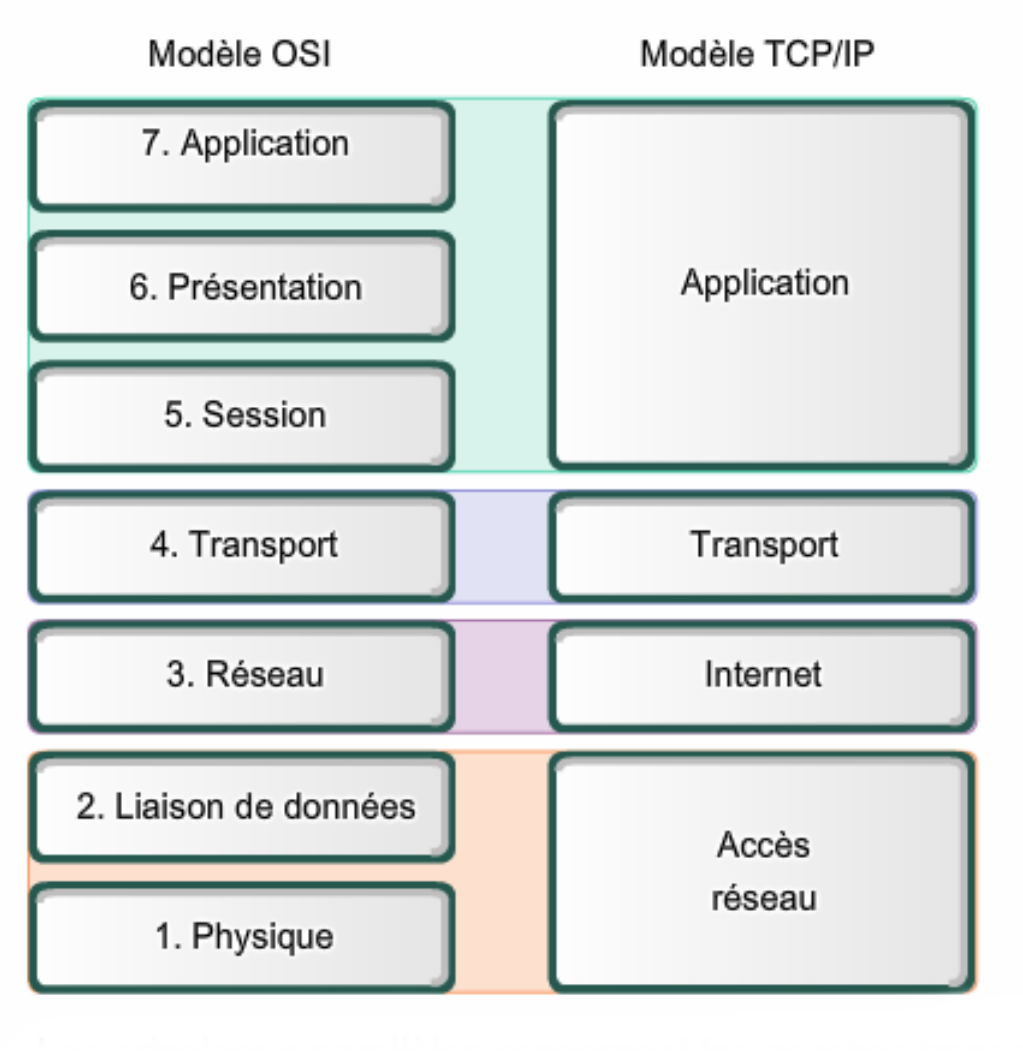


Figure 3.2: *Modèles OSI et TCP/IP*

Les rôles des différentes couches sont les suivants :

1. Couche Accès réseau : elle spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé.
2. Couche Internet : elle est chargée de fournir le paquet de données (datagramme)
3. Couche Transport : elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission.
4. Couche Application : elle englobe les applications standard du réseau (Telnet, SMTP, FTP, ...).

3.3.2 Encapsulation des données

Les données du trafic réseau TCP/IP sont composées de paquets, de connexions et de sessions. Un paquet est une unité de données qui est routé entre une source et une destination sur internet ou sur n'importe quel réseau.

Une connexion est une séquence unidirectionnelle de paquets entre une source et une destination donnée. Les données de session représentent la communication entre les ordinateurs, entre les hôtes, ou entre un ordinateur et un hôte. Habituellement, une telle communication implique l'échange de beaucoup de connexions. Les informations de session comprennent les informations sur la source du paquet et sur le port de destination, l'adresse IP et les types de services.

Les communications entre ordinateurs sont effectuées par la transmission de chiffres binaires. Pour que les ordinateurs émetteurs et récepteurs sachent ce que ces chiffres représentent, les chiffres doivent être regroupés sous une format logique. La dénomination de ce groupement change par rapport a chaque couche. Le paquet de données est appelé message au niveau de la couche Application. Le message est ensuite encapsulé sous forme de segment dans la couche Transport. Le segment une fois encapsulé dans la couche Internet prend le nom de datagramme. Enfin, on parle de trame au niveau de la couche Accès réseau. La figure 3.3 montre le processus d'encapsulation.

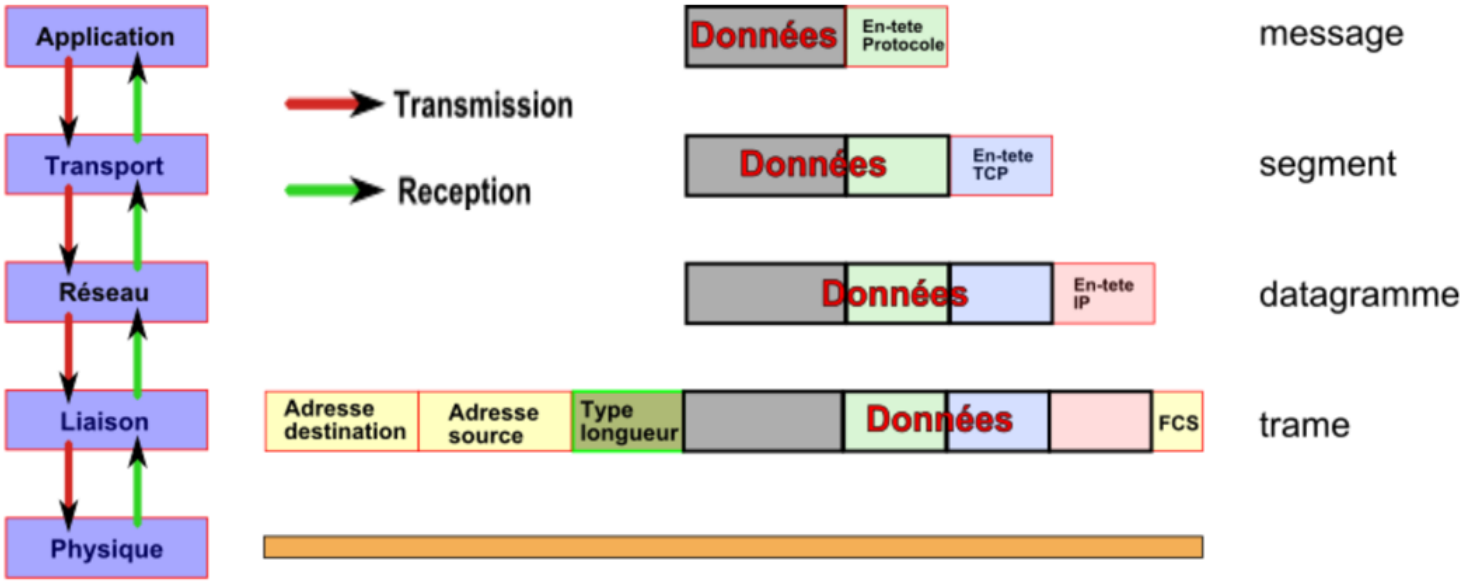


Figure 3.3: *Processus d'encapsulation*

Lors d'une transmission, les données traversent chacune des couches au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un en-tête, ensemble d'informations qui garantit la transmission. Au niveau de la machine réceptrice, lors du passage dans chaque couche, l'en-tête est lu, puis supprimé. Ainsi, à la réception, le message est dans son état originel.

3.3.3 Exemple de connexion TCP/IP

Chacune des lignes illustrées dans la figure 3.4 représente une connexion de type TCP/IP. Celles-ci constituent deux sessions. Les 2 premieres connexions représentent la premiere session tandis que les 2 dernieres forment la deuxième session. La première ligne indique que le trafic provient de 172.16.30.247 (numéro de port 80), à l'hôte 216.98.200.250 (numéro de port 63647). La deuxième ligne montre le trafic allant du deuxième hôte au premier. La troisième ligne montre le trafic provenant de 172.16.30.247 (numéro de port 80) à 216.98.200.250 (numéro de port 63648) et la dernière ligne affiche la connexion au sens inverse.

```
2005/04/28 19:14:01 172.16.30.247.80 -> 216.98.200.250.63647 6(SYN|ACK) 3 144
2005/04/28 19:14:01 216.98.200.250.63647 -> 172.16.30.247.80 6(SYN) 1 48
2005/04/28 19:14:01 172.16.30.247.80 -> 216.98.200.250.63648 6(SYN|ACK) 3 144
2005/04/28 19:14:01 216.98.200.250.63648 -> 172.16.30.247.80 6(SYN) 1 48
```

Figure 3.4: *Une section d'un trafic réseau*

3.3.4 Bases de données publiquement disponibles

Bien que nous ayons besoin de collecter des données réseau TCP/IP pour développer un NIDS comportementale il est grandement et fortement souhaitable d'avoir des données accessibles au public pour la recherche et l'évaluation de divers approches.

Pour tester et évaluer les IDS, les bases de données KDDcup99 et NSL-KDD ont été fortement utilisées. Ci-après, on décrit ces bases de données.

Description de KDDcup99

En 1998, les laboratoires de MIT Lincoln ont organisé un programme d'évaluation des systèmes de détection d'intrusion DARPA dans le but d'examiner et d'évaluer les recherches dans la détection d'intrusion. Pour cela, ils ont installé un environnement pour acquérir des connexions TCP/IP pendant neufs semaines dans un réseau local (LAN) simulant un LAN typique de l'Armée de l'Air des États-Unis. Le réseau a été actionné mais en lui injectant de multiples attaques. Pour chaque connexion TCP/IP, 41 caractéristiques quantitatives et qualitatives ont été extraites [146]. La compétition "KDD Intrusion Detection 1999" utilisait un sous ensemble de 494021 enregistrements ce qui représente 10% de la base de données globales. La base de données est publiquement accessible via [1]. Les principaux types d'attaques de l'ensemble de données KDD99 sont :

1. L'attaque d'exploration « Probing Attack »

   L'attaquant de cette classe commence par un sondage de la future victime, ce que l'on appelle scan. Ce sondage va balayer chaque port IP afin de connaître les services offerts par le système (OS, topologie du réseau, protections déployées,...). Une fois ce balayage

achevé, la machine de l'intrus (celui qui réalise l'intrusion) tente alors d'identifier le système d'exploitation utilisé par cette victime et d'exploiter les informations qu'elle a récoltées. Cette classe d'attaque est la plus étendue et elle requit une expertise technique minime. Les exemples de ce type d'attaque sont Ipsweep, Mscan, Nmap, Saint, Satan.

2. Attaques de Dénis de Services (DOS)

   Cette attaque porte bien son nom puisqu'elle aboutira à l'indisponibilité d'un service (application spécifique) ou d'une machine visée en occupant par exemple des ressources telle que la mémoire par de fausses requêtes. Il existe plusieurs types de déni de services, d'une part ceux qui exploitent les bugs d'une application et d'autre part ceux dus à une mauvaise implémentation d'un protocole ou à des faiblesse de celui-ci. Les principales attaques qu'on peut trouvées sont Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udpstorm.

3. L'attaque de passage d'un utilisateur à un super utilisateur « User to Root Attack (U2R) »

   L'objectif de cette classe d'attaques est d'obtenir les privilèges de l'administrateur système (Root) à partir d'un simple compte utilisateur par l'exploitation des vulnérabilités. Les exploits les plus connus sont les débordements réguliers des Buffers (buffer overflows) dus aux erreurs de programmation. Les principales attaques de ce type sont Eject, Ffbconfig, Fdformat, Loadmodule, Perl, Ps, Xterm.

4. L'attaque distance à local « Remote to Local Attack (R2L) »

   Dans cette classe, l'attaquant (machine distante) envoie des paquets vers une machine du réseau cible, après il exploite les vulnérabilités de cette machine afin d'avoir un accès illégal. Il y a plusieurs types d'attaque de R2L, les plus connues utilisent ou exploitent les bugs ou les mauvaises configuration des applications ou des systèmes. Les exemples de cette classe d'attaque sont Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock, Xsnoop.

Il faut noter que les données de test n'ont pas la même distribution de probabilité que les données d'apprentissage, et elles incluent des types d'attaques spécifiques qui n'existent pas dans les données d'apprentissage ce qui rend les données plus réalistes. Les ensembles de données contiennent un nombre total de 38 attaques, 24 types d'attaques dans les données d'apprentissage, et 14 autres types d'attaques existent dans les données de test seulement. Le nom et la description détaillée des types d'attaques sont répertoriés dans KDDcup99.

Le fichier "kddcup.data_10_percent" simule le trafic réseau utilisé dans la phase d'apprentissage, et le fichier "corrected" simule le trafic réseau dans la phase de test.

Description de NSL-KDD

L'ensemble de données NSL-KDD a été proposé pour résoudre certains problèmes inhérents de l'ensemble de données KDDcup99 qu'ils ont été mentionnés dans l'article de [154]. Bien que cette nouvelle version de données souffre encore de certains problèmes évoqués par McHugh [114] et

ne peut pas être un représentant parfait des véritables réseaux existants, cet ensemble de données peut encore être appliqué en tant qu'ensemble de données de référence efficace pour aider les chercheurs à comparer les différentes méthodes de détection d'intrusion. La base de données est publiquement accessible via [2].

Le nombre d'enregistrements dans les données d'apprentissage et de test du NSL-KDD sont raisonnable. Cet avantage rend abordable d'exécuter les expérimentations sur l'ensemble complet sans la nécessité de choisir au hasard une petite partie. Par conséquent, les résultats de l'évaluation des différents travaux de recherche seront cohérents et comparables. L'ensemble de données NSL-KDD présente les avantages suivants par rapport à l'ensemble original de données (KDDcup99):

1. Il n'inclut pas les enregistrements redondants dans les données d'apprentissage, de sorte que les classificateurs ne seront pas biaisés en faveur des enregistrements les plus fréquents.

2. Il n'y a pas des enregistrements redondants dans les ensembles de données de test proposées, par conséquent, les performances des classificateurs ne seront pas biaisées par les méthodes qui ont un meilleur taux de détection pour les enregistrements fréquents.

3. Le nombre d'enregistrements sélectionnés dans chaque groupe de niveau de difficulté est inversement proportionnel au pourcentage d'enregistrements dans l'ensemble original de données. En conséquence, les taux de classification des méthodes d'apprentissage automatique varient dans une plage plus large, ce qui lui rend plus efficace pour avoir une évaluation précise des différentes techniques d'apprentissage.

4. Le nombre d'enregistrements des données d'apprentissage et de test est raisonnable, ce qui rend abordable d'exécuter les expérimentations sur l'intégralité des données sans la nécessité de choisir au hasard une petite partie. Par conséquent, les résultats de l'évaluation des différents travaux de recherche seront cohérents et comparables.

3.4 Phase d'apprentissage

Au cours de la phase d'apprentissage, l'IDS manipulera seulement les caractéristiques pertinentes extraites du trafic réseau. En faisant cela, on se débarrasse des données inutiles qui mènent a une réduction de taux de détection et a une augmentation de faux positives. Les algorithmes d'extraction de caractéristiques ont pour but aussi la réduction du temps d'apprentissage de l'IDS.

Apres la fin de cette phase, on obtient un modèle représentant le comportement normal du trafic réseau. Ce modèle nous sera utile dans la phase de détection.

Dans ce qui suit on présente quelques algorithmes d'extraction de caractéristiques utilisées dans la littérature. Cette thèse s'intéresse en particulier à l'utilisation de l'analyse en composantes principales et l'analyse discriminante linéaire et à l'optimisation de ces derniers.

Dans le but de réduire la dimension des données de départ, plusieurs méthodes ont été proposées en vue de transformer le vecteur des données originales dans un autre espace, de faible dimension, sans pour autant éliminer les informations discriminatoires qui seront utilisées durant

l'étape de classification. La transformation des vecteurs de données peut être réalisée de manière linéaire ou non-linéaire.

Ainsi, les connexions réseau seront projetées dans un espace de plus faible dimension. Les nombreuses méthodes de projection existantes privilégient la bonne représentation des données suivant un point de vue. Par exemple, l'analyse en composantes principales ou le classical multidimensional scaling (MDS) permettent de maximiser la variance des données dans la représentation. La représentation exprime alors la forme générale du jeu de données. Les méthodes supervisées comme l'analyse discriminante projettent les données de façon à valoriser la séparation des classes.

les méthodes d'extraction de caractéristiques peuvent être classifiées en deux grandes catégories à savoir les méthodes linéaires et les méthodes non linéaires.

3.4.1 Méthodes linéaires

Analyse en composantes principales

L'analyse en composantes principales -Principal component analysis (PCA) [76] est une méthode d'extraction de caractéristiques non-supervisée permettant de réduire la dimension d'une matrice de données tout en maximisant leur variance. Elle consiste à transformer des caractéristiques liées entre elles en nouvelles caractéristiques décorrélées les unes des autres. Ces dernières sont des combinaisons linéaires des caractéristiques originelles. Afin d'obtenir ce resultat, PCA cherche une base de composantes principales à partir des vecteurs propres de la matrice de covariance des données. Parmi les travaux pionniers qui ont exploité cette methode dans la detection des intrusions, on cite [24, 166, 167]. Plus recemment, d'autres auteurs ont utilisé cette approche tel que [27, 74, 136].

Par ailleurs, cette méthode souffre de plusieurs limitations. On cite parmi elles, la sensibilité aux données aberrantes et la nature linéaire non discriminante.

Afin de contourner la première limite, Shyu *et al*. [144] proposent d'appliquer PCA sur les données surveillées et utilisent deux fonctions de détection. La première est appliquée sur l'espace composé par les premières composantes principales qui totalisent environ 50% de la variance initiale afin de détecter les données aberrantes. La deuxième étant sur les dernières composantes principales qui totalisent moins de 20% de la variance initiale afin de détecter les données qui possèdent une structure de corrélation différente de l'ensemble des données. Dans les deux cas, la distance de Mahalanobis est utilisée pour identifier les données aberrantes. A la différence de la distance euclidienne qui donne le même poids à toutes les variables utilisées dans le calcul, la distance de Mahalanobis considère la variance des variables et donne un poids à chacune dans le calcul. Cette distance est calculée par rapport à une moyenne définie par apprentissage et comparée à un seuil fixé en paramètre.

La travail de Lee *et al*. [93] se base sur la direction du premier vecteur propre et sur le fait que des données aberrantes peuvent changer cette direction. En effet, dupliquer des données de comportement légitime n'affecte pas la direction du premier vecteur propre. Cependant, dupliquer des données aberrantes modifie clairement cette direction. La duplication des données se fait par

un sur-échantillonnage effectué sur l'ensemble des données initiales. Le processus de détection se fait en comparant l'angle dessiné par les deux positions du premier vecteur propre avant et après le sur-échantillonnage, par rapport à un seuil prédéfini.

Pour remédier aux limitations de PCA, plusieurs variantes ont été aussi proposées dans d'autre domaines de recherche. En particulier dans la reconnaissance de visages. On cite le travail de Moghaddam *et al*. [120, 121] qui a introduit une PCA intégrant la théorie bayésienne des probabilités, [156] a introduit une PCA robuste aux données aberrantes, Hastie et al. [65] ont proposé une PCA non linéaire (principal curves). Wang [165] de son coté a proposé une solution pour sélectionner le nombre optimale des composantes principales. Un PCA incrémentiel résolvant les problèmes de calcul matriciel a été proposé par Artac [9]. Une autre variante qui minimise la variance dans les directions supposées contenir du bruit et maximise la variance totale a été introduite par Torre [157].

Analyse multidimensionnelle (MDS)

L'analyse multidimensionnelle –MultiDimensional Scaling (MDS)– comporte divers methodes qui visent à représenter chaque donnée dans un espace euclidien, habituellement bi ou tridimensionnel, de telle sorte que deux données semblables soient représentés par deux points proches l'un de l'autre, et un couple dissemblable par des points éloignés. Cette idée a évolué pour considérer des dissimilarités ou d'autres types de vraisemblances entre des points à la place des distances euclidiennes. Il y a une équivalence entre le MDS et PCA. Les coordonnées principales des données obtenues en MDS correspondent aux composantes principales des données trouvées par PCA. [21].

Pour achever la reduction de dimension, une fonction qui calcule la déviation entre les distances géodésiques mesurées dans l'espace initial de grande dimension, et les distances Euclidiennes mesurées dans l'espace d'arrivée de plus petite dimension est minimisée. Dans la litterature, plusieurs types et variantes d'algorithmes MDS ont été proposé, le plus utilisé est Sammon mapping qui est un algorithme de la famille "metric MDS" [140].

Analyse en composantes indépendantes

L'analyse en composantes indépendantes –Independent Component Analysis (ICA)– a été introduite afin de résoudre le problème de séparation de sources [38]. Le concept du problème est de retrouver des signaux qui ont été mélangés et bruités éventuellement, en tenant compte de toute information disponible sur les signaux d'intérêt et sur le processus de mélange. Les signaux d'intérêt sont appelés sources alors que les signaux mesurés sont appelés observations. Afin de ressortir des sources aussi indépendantes que possible à partir des observations. La méthode tire profit d'une décomposition d'une source aléatoire multivariable en une combinaison linéaire de sources indépendantes. l'ICA a été utilisé avec succès dans plusieurs domaines [81, 105, 178]. Cette approche a donné des bons resultats dans la détection des intrusions [126, 174].

Analyse de corrélation canonique

On utilise L'analyse de corrélation canonique –Canonical Correlation Analysis (CCA)– pour explorer les relations linéaires entre deux variables aléatoires ou bien entre deux ensembles de vecteurs [70, 78] .

L'approche consiste à mesurer et de caractériser la dépendance linéaire au cas ou elle existe entre deux groupes de variables mesurées sur les mêmes individus. La notion de dépendance est mesurée ici par le coefficient de corrélation maximal entre les combinaisons linéaires des variables du premier groupe et les combinaisons linéaires des variables du deuxième groupe. Ce coefficient est appelé facteur canonique.

Comme PCA, la méthode CCA permet de réduire la dimensionalité des données en ne prenant qu'un nombre limité de facteurs canoniques pour représenter efficacement les données originelles [152, 153].

Factorisation en matrices non négatives

La factorisation en matrices non négatives –Non negative Matrix Factorization (NMF)– est une approche générale qui se base sur la décomposition matricielle pour trouver un espace de dimension réduite sur lequel les nouvelles données sont représentées [91, 92]. Elle permet d'approximer toute matrice positive V de taille $n \times m$, par une décomposition de la forme $V \approx WH$, où W et H sont des matrices $(n \times k)$ et $(k \times m)$.

les vecteurs originales de dimension m composent la matrice V, les vecteurs correspondants ayant une dimension réduite $k < m$ appartiennent a W , et la matrice de passage H contient les vecteurs de base.

les contraintes de non-négativité (tous les élements sont positifs) que NMF impose à W et H font de cette dernière une méthode originale. La NMF a été appliquée à la détection d'intrusions dans quelques travaux comme [63, 168].

Analyse discriminante linéaire

L'analyse discriminante linaire –Linear Discriminant Analysis (LDA)– est un algorithme d'extraction de caractéristiques supervisé qui permet d'obtenir une séparation linéaire entre un ensemble de classes [58]. Afin d'achever cet objectif, LDA minimise la dispersion intra-classe tout en maximisant la dispersion inter-classe.

En d'autres termes, LDA trouve un ensemble de vecteurs de projection W qui maximise une matrice de dispersion inter-classe appelée S_b et minimise une matrice de dispersion intra-classe appelée S_w.

Cependant la méthode LDA classique souffre de plusieurs problèmes. Le premier est connu sous le nom de "Small Sample Size Problem" (SSS), il se produit lorsque la dimension des données d'apprentissage est très grande par rapport au nombre des données.

Par consequent, la matrice S_w peut devenir singulière et cela rend difficile le calcul des vecteurs de projections.

Plusieurs approches ont été proposées pour résoudre ce problème. Parmi les solutions on peut citer l'utilisation de la méthode LDA régularisée (R-LDA regularized LDA) [41] qui additionne une matrice diagonale $\alpha \times I$ à la matrice S_w pour eviter la singularité de celle-ci. Fisherface (PCA+LDA) [14] considère une étape préliminaire de réduction de dimensions avant d'appliquer LDA. La méthode de l'espace nul de S_w [32] exploite l'espace nul de S_w pour obtenir la transformation W. Un autre algorithme appelé Direct LDA (D-LDA) [138, 177] essaie de résoudre ce problème en inversant les étapes de diagonalisation simultanées. Un algorithme basé sur la décomposition QR qui elimine la singularité des matrices de dispersion a été introduit par Ye *et al.* [176]. Hansun Park [71] a utilisé la décomposition SVD généralisée pour résoudre le SSS. D'autres travaux exploitent le critère de marge maximale pondérée (WMMC) [94, 97]. Cette approche considère la différence pondérée entre la dispersion inter-classe et la dispersion intra-classe. Par la suite, une version robuste de WMMC utilisant la norme L1 invariante rotationnelle (R1-LDA) a été développée [96].

Un deuxième problème découle de la formulation de LDA. Cette dernière utilise d'une part la norme L2, d'autre part elle exploite la moyenne classique pour calculer les matrices de dispersion. Or l'utilisation de cette norme amplifie l'effet des valeurs aberrantes dans les données et mène a une projection erronée. Pour corriger cette carrence, des travaux qui considèrent les modèles sparsés de LDA "sparse LDA methods" [171]. D'autres articles ont proposé la régularisation supervisée basée sur un sous-espace robuste (SRRS), qui reconstruit les données originales en employant une représentation (LRR) [106] et applique simultanément LDA. Par ailleurs, l'effet de la norme L2 persiste encore. Pour contourner le problème, de nombreux chercheurs ont proposé des méthodes de LDA plus robustes intégrants la norme L1 [163, 182, 183].

La moyenne classique est sensible aussi au données aberrantes et peut falsifier le calcul [175]. Pour résoudre ce problème, des publications récentes ont proposé des moyennes plus robustes aux données aberrantes tel que [95, 181]. La première proposition se base sur le critère de marge maximale (MMC) et intègre le vecteur moyen maximum-minimum-médian au lieu de la moyenne classique pour construire la matrice de dispersion intra-classe S_w et la matrice de dispersion entre classes S_b. Les résultats expérimentaux sur les bases de données ORL et Yale montrent qu'une amélioration du modèle (MMC) est possible avec la technique proposée. La deuxième approche minimise l'inverse de la moyenne harmonique pondérée de la distance entre les classes. Ce qui s'avère plus robuste que la minimisation de l'inverse de la moyenne arithmétique classique. Des expériences approfondies sur différents types de données montrent que l'approche surpasse plusieurs autres en termes précision de la classification.

Un troisième probleme concerne le type de structure de données manipulée par LDA. Cette approche accorde plus d'attention à la structure globale des classes. En conséquence, les caractéristiques discriminantes produites sont souvent imprécises. Pour remédier à cette situation, des travaux antérieurs [31, 151, 164] ont exploité la structure locale pour obtenir des caractéristiques optimales. Néanmoins, dans ces travaux, il est nécessaire de faire une décomposition d'une

énorme matrice. Pour la détection d'intrusion ca sera une tâche longue et même infaisable.

3.4.2 Méthodes non linéaires

En général on distingue d'une part, les méthodes linéaires qui ne prennent pas en compte les non-linéarités des données traitées. Et d'autre part, les algorithmes dits non linéaires introduits par les chercheurs pour essayer de résoudre les problèmes rencontrés par les methodes linéaires.

Ces problèmes peuvent dégrader d'une façon considérable les performances d'un IDS en termes de taux de detection. Les algorithmes non linéaires les plus utilisés sont ceux basés sur les noyaux (kernel based methods) et sur l'apprentissage de variétés (Manifold learning).

Méthodes basées sur les noyaux

Ces techniques reposent sur une transformation non linéaire des données via des fonctions habituellement non linéaires, appelées fonctions noyaux. Cette transformation effectue un changement de base qui permet de projeter les données de l'espace d'entrée dans un nouvel espace où les relations entre les variables sont linéaires. Pratiquement, les produits scalaires des arguments sont calculés par les fonctions noyaux dans ce nouvel espace. La fonction qui calcule implicitement le produit scalaire dans l'espace des caractéristiques prend comme parametres ces arguments.

Une multitude de fonctions noyaux est employée dans la littérature. Chacune définit, avec ses paramètres, un espace de caractéristiques unique. Par ailleurs, les algorithmes qu'utilisent des fonctions noyaux sont conçus de telle manière que les coordonnées des projections ne soient plus nécessaires. Seule est nécessaire la relation entre données en termes de produits scalaires et de distances dans l'espace de caractéristiques.

Plusieurs algorithmes de reconnaissance des formes peuvent être développés de manière à intégrer des fonctions noyaux et ainsi étendre des algorithmes simples (linéaires) dans l'espace de caractéristiques à des algorithmes non-linéaires dans l'espace original.

Parmi les méthodes basées sur les noyaux on cite Kernel PCA (KPCA) [141, 180], kernel LDA et ses variantes [107, 118], kernel ICA [113], kernel CCA [173] et kernel NMF.

Pour améliorer la détection des intrusions, Kuang *et al.* [85] se sont beaucoup intéressé à l'utilisation de KPCA. Ils ont proposé d'utiliser KPCA avec la Machine à vecteurs de support (SVM) et l'algorithme génétique (GA) en se basant sur un noyau gaussien. SVM a été utilisé comme étape de classification et GA pour optimiser le paramètre de SVM. Par ailleurs, le temps d'apprentissage de l'approche s'est avéré très long. Pour remédier a cela, les mêmes auteurs en 2014 ont présenté un noyau gaussien plus performant appelé (N-RBF) [84]. Cette dernière publication a raccourci le temps d'apprentissage et améliorer la performance de SVM. En 2015, ils ont combiné KPCA avec une méthode robuste d'optimisation par essaims particulaires (ICPSO) [86] pour améliorer davantage l'efficacité de l'IDS.

Méthodes basées sur l'apprentissage de variétés

l'apprentissage par variété regroupe un ensemble de techniques de réduction de la dimensionnalité. Ces techniques transforment les ensembles de données X dont la dimensionnalité est D en un nouvel ensemble de données Y avec une dimensionnalité d, tout en conservant la géométrie des données autant que possible.

Une généralisation non-linéaire de l'algorithme MDS appelée Isomap [155] appartient a cette famille. L'algorithme Locally Linear Embedding (LLE) [137] est une technique qui est similaire à l'Isomap par la construction d'un graphe représentant les données. Contrairement à l'Isomap, il tente de préserver uniquement les propriétés locales des données. Dans la représentation de faible dimensionnalité, le LLE tente de conserver le poids de reconstruction dans les combinaisons linéaires aussi bien que possible. Une des limitations d'Isomap et de LLE est qu'ils ne sont pas adaptés pour traiter des ensemblesnon convexes.

D'une manière similaire à la technique LLE, l'algorithme Laplacian Eigenmaps [15] trouve une représentation de faible dimensionnalité des données en préservant les propriétés locales de la variété, par ailleurs, il possède un fondement théorique different: l'information de voisinage est récupérée à l'aide d'un graphe mais les coordonnées de faible dimension sont obtenues à partir de la notion du laplacien du graphe[37].

3.5 Phase de détection

Au cours de la phase de détection, les caractéristiques extraites dans la phase d'apprentissage sont réutilisées pour identifier de nouvelles connexions réseaux.

Dans cette partie, le système doit deviner la nature du nouveau trafic. Le système compare le nouvel trafic avec le modèle grâce à une phase de classification. Cette dernière décidera si le nouveau trafic est malicieux ou pas.

Parmi les algorithmes de classification ils existent le réseau de neurones artificiels (ANN), la machine à vecteurs de support (SVM), les arbres de décision, les réseaux bayésiens (BNS), le K plus proche voisin (KNN),...etc.

Dans ce qui suit on decrit quelques algorithmes de classification utilisés dans la littérature. Notre IDS utilise KNN et les arbres de décision en particulier.

3.5.1 La machine à vecteurs de support (SVM)

La machine à vecteurs de support (SVM) a été proposée par Vapnik en 1998 [161]. Son principe est illustré par la figure 3.5. On trace un vecteur d'entrée dans un espace de caractéristiques de grande dimension, afin d'obtenir l'hyperplan optimal de séparation pour cet espace. De plus, une limite de décision est déterminée par des vecteurs de soutien plutôt que des échantillons d'apprentissage entiers, par conséquent la machine à vecteurs de support est extrêmement robuste aux valeurs extrêmes. En particulier, SVM est conçu pour la classification binaire afin de séparer un ensemble de vecteurs d'apprentissage qui appartiennent à deux classes différentes. Il faut noter que les vecteurs

de support sont des échantillons d'apprentissage à proximité d'une frontière de décision. Le SVM fournit également un paramètre spécifié d'utilisation appelé facteur de pénalité. Ce paramètre permet aux utilisateurs de faire un compromis entre le nombre des échantillons mal classés et la largeur d'une frontière de décision. Cette méthode de classification a été employée dans divers travaux visant la détection d'intrusion. Dans le travail de Li *et al.* [99], SVM avec un noyau RBF a été utilisé pour classer l'ensemble des données KDDcup99 dans des catégories prédéfinies. Amiri *et al.* [7] ont utilisé un SVM basé sur la méthode des moindres carrés pour accélérer le processus de classification. En ce qui concerne l'approche comportementale, Hu et al. [72] ont utilisé un robuste SVM nommé (RSVM). Ce dernier emploie un hyperplan discriminant plus mince, et peut automatiquement déterminer le facteur de pénalité. Ce qui mène a un taux de détection de 75%.

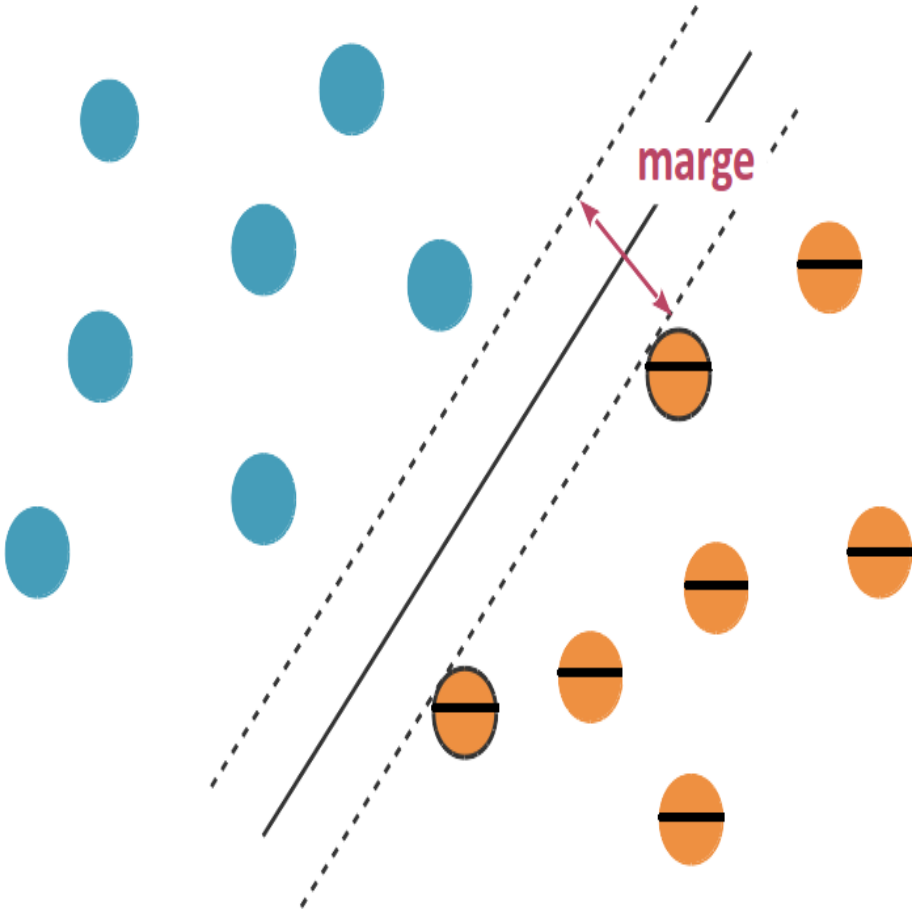


Figure 3.5: *Exemple de deux classes linéairement séparables par SVM*

3.5.2 Le réseau de neurones artificiels

Le réseau de neurones est un ensemble des unités de traitement de l'information qui a pour but d'imiter les neurones du cerveau humain [66]. Le Perceptron multicouche (MLP) est l'architecture de réseau de neurones la plus utilisée dans de nombreux problèmes de reconnaissance des formes. Comme le montre la figure 3.6, un réseau MLP se compose d'une couche d'entrée qui contient

un ensemble de nœuds sensoriels comme des nœuds d'entrées, une ou plusieurs couches cachées de nœuds de calcul, et une couche de sortie de nœuds de calcul. Chaque interconnexion est associée à une pondération scalaire qui est ajustée pendant la phase d'apprentissage. L'algorithme d'apprentissage de rétropropagation est généralement utilisé pour former un MLP. Au début, des poids aléatoires sont attribués. Ensuite, l'algorithme de rétropropagation effectue le réglage des poids, pour définir laquelle des représentations des unités cachées est plus efficace pour minimiser l'erreur de classification erronée. Ce classifieur a mené a des IDS plus performants [20, 29, 103].

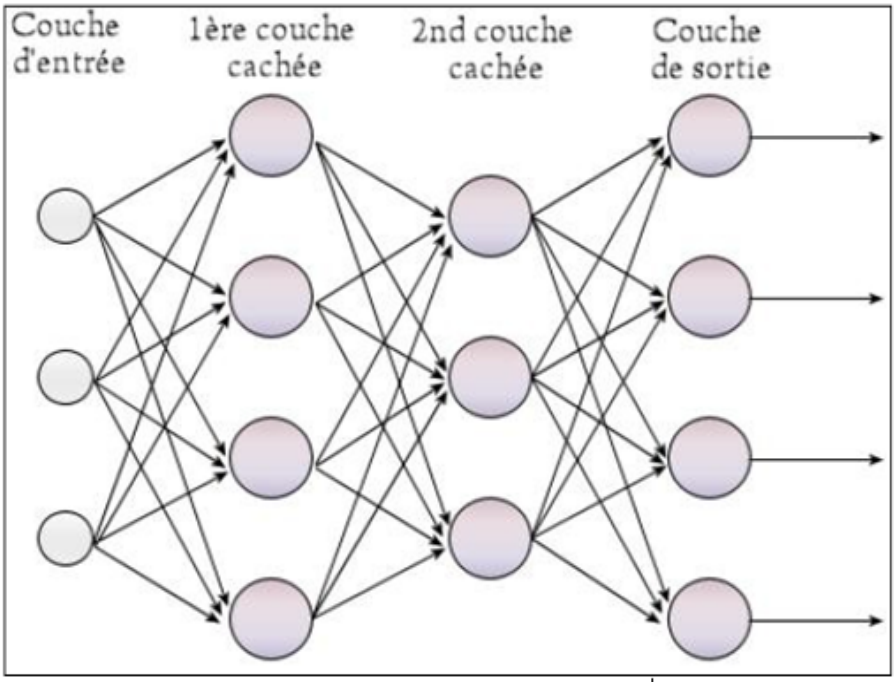


Figure 3.6: *Le perceptron multicouche*

3.5.3 Les réseaux bayésiens

Il existe de nombreux cas où nous connaissons les dépendances statistiques ou les relations causales entre les variables du système. Toutefois, il pourrait être difficile d'exprimer avec précision les relations probabilistes entre ces variables. En d'autres termes, la connaissance préalable du système est tout simplement représentée par le fait que certaines variables peuvent influencer les autres. Pour exploiter cette relation structurelle ou cette causale dépendance entre les variables aléatoires d'un problème, on peut utiliser un modèle de graphe probabiliste appelé réseau Bayésien Naïve (NB). Le modèle offre une réponse à des questions comme "Qu'elle est la probabilité de l'existence d'un certain type d'attaque, compte tenu de certains événements observés dans le système?" En utilisant la formule de probabilité conditionnelle. Comme le montre la figure 3.7, la structure d'un NB est généralement représentée par un graphe acyclique (DAG), où chaque nœud représente l'une des variables du système et chaque lien encode l'influence d'un nœud sur un autre [128]. Donc, s'il y a un lien entre le nœud A et le nœud B, alors A influe directement sur B.

Livadas *et al.* [108] se sont basé sur (NB) afin de determiner l'existence de botnet et ses origines dans le trafic Internet Relay Chat (IRC) . L'étude utilise les données TCP recueillies à partir de 18 endroits de l'Université de Dartmouth. La performance du réseau bayésien a produit 93% de précision avec 1,39% de faux positives.

Un détecteur d'intrusion DoS utilisant un réseau bayésien est décrit par Benferhat *et al.* [16]. Ils considerent un seul noeud parent représentant la variable cachée (classe normale ou malicieuse), et les variables observées sont des nœuds enfants. les nœuds enfants sont supposés être statistiquement indépendants. l'objectif principal de cette approche était d'effectuer une corrélation d'alerte avec une minimale utilisation des connaissances d'experts. Malheureusement, l'étude n'a rapporté aucun résultats.

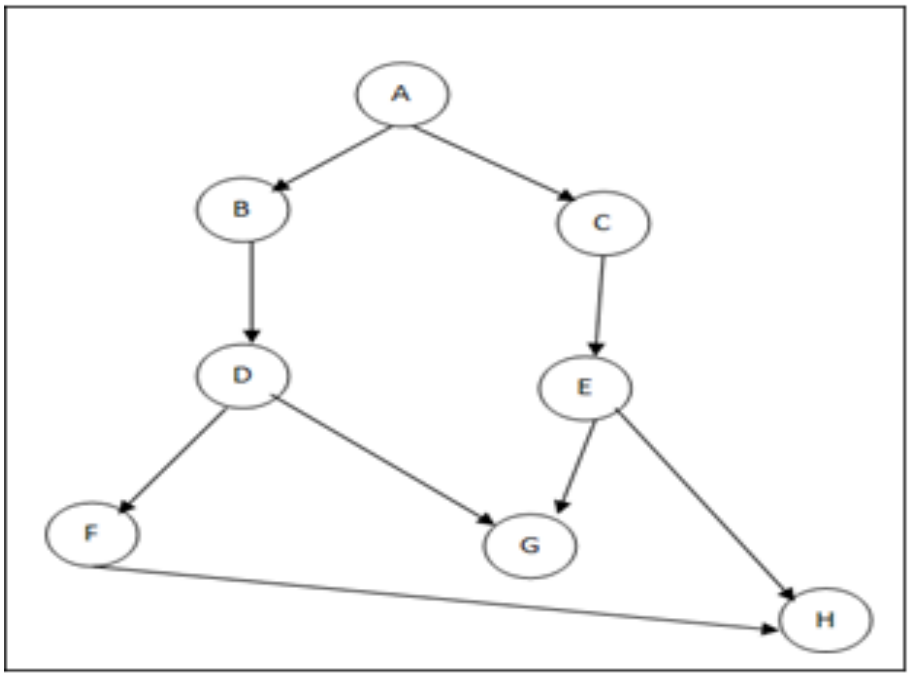


Figure 3.7: *Réseau bayésien naïve*

3.5.4 L'arbre de décision

Un arbre de décision classifie un échantillon à travers une série de décisions, dont la décision actuelle contribue à la décision ultérieure. Comme l'illustre la figure 3.8, la série de décisions est représentée sous forme d'une structure arborescente. La classification de l'échantillon se fait à partir du nœud racine à un souhaitable nœud feuille, où chaque nœud feuille représente une catégorie de classification. Les attributs des échantillons sont assignés à chaque nœud, et la valeur de chaque branche est correspondante aux attributs [139]. CART (Classification And Regression Trees) est un programme bien connu pour la construction des arbres de décision. Un arbre de décision avec des étiquettes discrètes de classe (symbolique) est appelé un arbre de classification, tandis qu'un arbre de décision avec une plage de valeurs continues (numérique) est appelé un arbre

de régression. Parmi les Les IDS implementant cette méthode de classifications on cite [18, 17, 83].

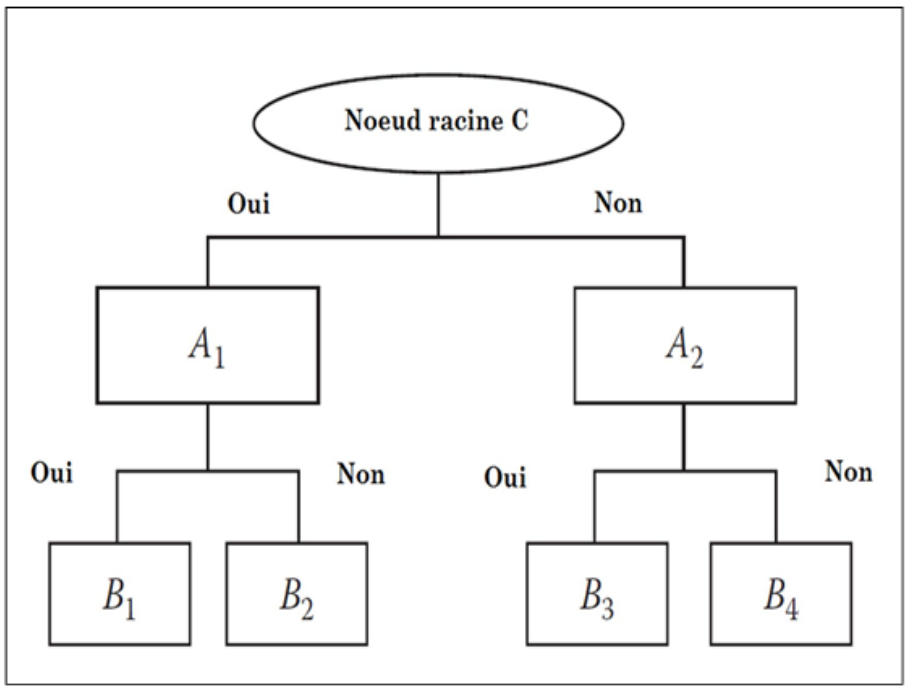


Figure 3.8: *L'arbre de décision*

3.5.5 Le K-plus proche voisin

Le K-plus proche voisin (K-NN) est l'un des methodes non paramétriques le plus simple et le plus traditionnellement utilisé pour classer les échantillons [111]. K-NN calcule les distances approximatives entre les différents points sur les vecteurs d'entrée, puis affecte le point non marqué à la classe de ses K-plus proches voisins. Dans le processus de création du classificateur K-NN, K est un paramètre très important et le changement des valeurs de K affectera les performances de notre classificateur. Si K est considérablement grand alors les voisins utilisés pour la prédiction vont prendre beaucoup de temps pour la classification. K- NN est appelé aussi algorithme d'apprentissage par l'exemple, et il est différent de l'approche d'apprentissage inductive. K-NN ne contient pas une étape d'apprentissage du modèle, il ne cherche que les exemples des vecteurs des entrées et il classifie les nouvelles instances. Par conséquent, K-NN apprend d'une manière "en ligne" les exemples et découvre le K plus proche voisin de la nouvelle instance. Parmi les travaux pionniers exploitant ce classifieur dans le domaine de la detection des intrusions [90, 98, 100].

3.6 L'architecture detaillée de l'IDS

Après la présentation d'une manière générale du type du trafic réseau manipulée, et des algorithmes d'extraction de caractéristiques et de classification. Cette thèse s'intéressera plus partic-

ulièrement au développement de nouvelles variantes de PCA/LDA pour résoudre ces problèmes décrits dans la section 3.4.1 d'une part, et d'optimiser la performance de l'IDS d'autre part. Ci-après, le model détaillé de l'IDS proposé qui sera adopté dans la suite de cette thèse.

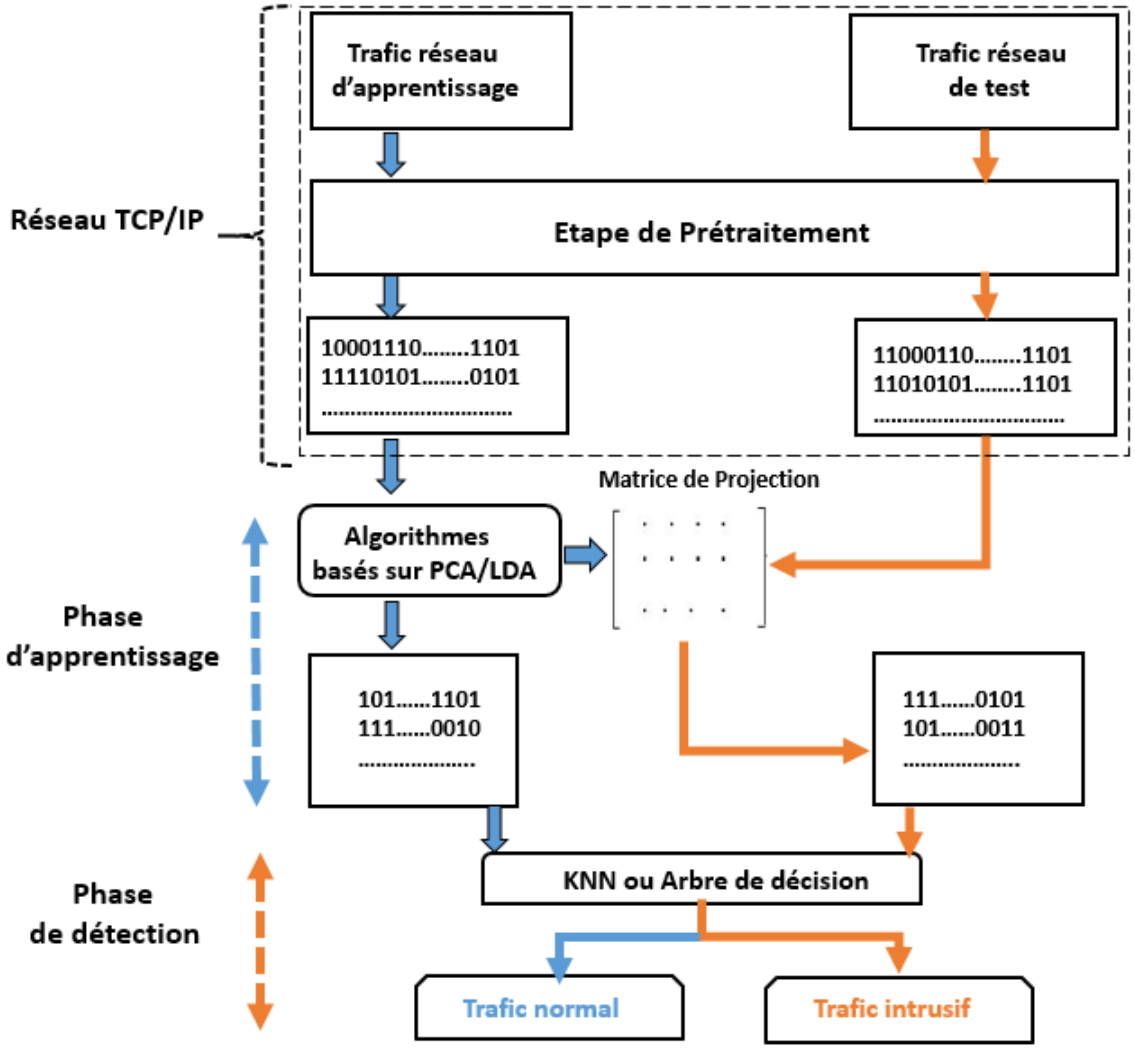


Figure 3.9: *Approche basée sur des nouvelles variantes de PCA/LDA*

L'architecture de L'IDS proposé dans la figure 3.9 représente une architecture plus détaillée que celle du 3.1.

L'IDS ne manipule que des données numeriques, donc il faut passer par une phase de conversion de caractéristique. KDDcup99 et NSL-KDD sont définis par des caractéristiques continues et discrètes, comme le montre le tableau suivant:

Table 3.1: *Caractéristiques de KDDcup99 et NSL-KDD*

#	Nom de la caractéristique	Type de la caractéristique
1	Duration	Continue
2	Protocol_type	Discrète

3	Service	Discrète
4	Flag	Discrète
5	Src_bytes	Continue
6	Dst_bytes	Continue
7	Land	Discrète
8	Wrong_fragment	Continue
9	Urgent	Continue
10	Hot	Continue
11	Num_failed_logins	Continue
12	Logged_in	Discrète
13	Num_compromised	Continue
14	Root_shell	Continue
15	Su_attempted	Continue
16	Num_root	Continue
17	Num_file_creations	Continue
18	Num_shells	Continue
19	Num_access_files	Continue
20	Num_outbound_cmds	Continue
21	Is_host_login	Discrète
22	Is_guest_login	Discrète
23	Count	Discrete
24	Srv_count	Continue
25	Serror_rate	Continue
26	Srv_serror_rate	Continue
27	Rerror_rate	Continue
28	Srv_rerror_rate	Continue
29	Same_srv_rate	Continue
30	Diff_srv_rate	Continue
31	Srv_diff_host_rate	Continue
32	Dst_host_count	Continue
33	Dst_host_srv_count	Continue
34	Dst_host_same_srv_rate	Continue
35	Dst_host_diff_srv_rate	Continue
36	Dst_host_same_src_port_rate	Continue
37	Dst host srv diff host rate	Continue
38	Dst_host_serror_rate	Continue
39	Dst_host_srv_serror_rate	Continue
40	Dst_host_rerror_rate	Continue

41	Dst_host_srv_rerror_rate	Continue

Nous avons transformé les caractéristiques discrètes en des valeurs continues en appliquant la transformation utilisée dans [24]. Décrivons brièvement ce processus: si une caractéristique discrète i prend k valeurs. i est convertie en k coordonnées composés de un et zéro. Par exemple, si nous considérons la caractéristique "type de protocole" qui peut être tcp, udp ou icmp.

Selon l'idée, il y aura trois coordonnées pour "type de protocole". Ainsi à tcp (resp. udp ou icmp) correspondent (1,0,0) (resp. (0,1,0) ou (0,0,1)). Avec cette technique, chaque connexion réseau sera représentée par un vecteur de 125 coordonnés. Un exemple de conversion est illustré par la figure 3.10.

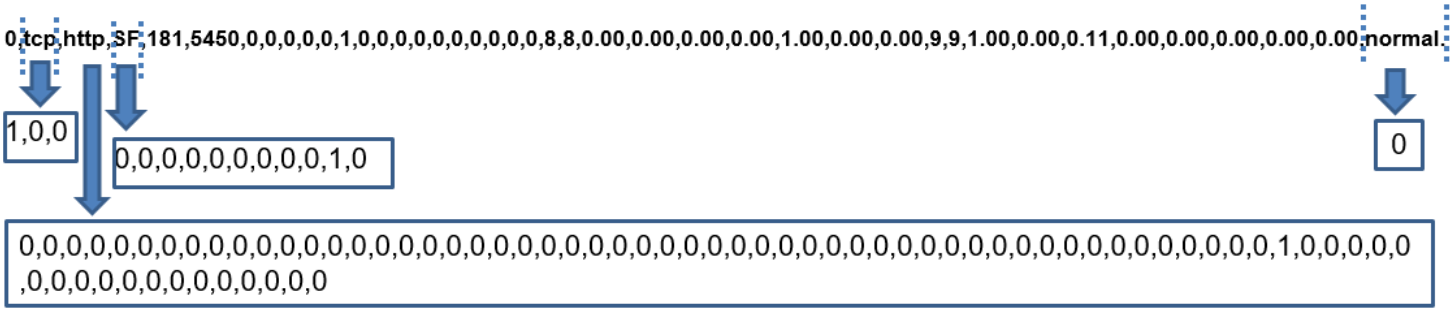


Figure 3.10: *Conversion vers un format numérique d'une connexion réseau appartenant à KDDcup99*

Au début, les connexions du trafic réseau d'apprentissage et de test vont être sélectionnées d'une manière aléatoire. Ensuite ces connexions seront prétraitées en suivant la procédure mentionnée dans la sous-section précedante afin d'obtenir un format numérique standard. Par la suite, une variante de PCA ou LDA est utilisée pour construire une matrice d'extraction de caractéristiques d'une part et obtenir des nouveaux vecteurs d'apprentissage ayants une petite dimension et libérés des caractéristiques inutiles d'autre part . Ces dernières représenteront le modèle de l'IDS. On projette après les connexions de test à l'aide de la matrice obtenue sur le même espace que celui obtenu à la phase d'apprentissage . A la fin de la phase de détection ou de classification. l'IDS aura recours à KNN ou l'arbre de décision pour décider si le trafic de test correspond à un trafic normale ou intrusive. Pour évaluer l'IDS, nous nous basons sur le taux de détection (DR) et le taux des faux positifs données par (2.2) et (2.1).

3.7 Conclusion

Dans un premier temps, ce chapitre a présenté l'architecture globale de l'IDS proposé. Il a décrit le type du trafic réseau manipulé par l'IDS. Ensuite, il a expliqué en quoi consistent la phase d'apprentissage et la phase de détection du système proposé. Dans la première phase on utilise un algorithme d'extraction de caractéristiques. Ce dernier peut être de nature linéaire ou non linéaire mais ayant un but commun. C'est la projection des données dans un nouveau sous-espace,

qui élimine les éléments ne portant pas une information aidant à classifier les intrusions. Dans la deuxième phase, on réutilise les caractéristiques extraites dans la phase d'apprentissage pour identifier de nouvelles connexions réseaux. Cette identification est faite à l'aide d'un algorithme de classification.

Dans un deuxième temps, le chapitre propose une architecture plus détaillé de l'IDS. Ce dernier s'intéresse en particulier à l'utilisation et à l'optimisation de deux fameux algorithmes d'extraction de caractéristiques. A savoir PCA et LDA. l'IDS utilise dans la phase de classification soit KNN ou l'arbre de décision.

CHAPITRE

4 Amélioration des Algorithmes d'Analyse en Composantes Principales (PCA)

"Brevity is the Soul of Wit"
WILLIAM SHAKESPEARE

4.1 Introduction

L'analyse en composantes principales -Principal component analysis (PCA)- est une méthode d'extraction de caractéristiques qui permet de réduire la dimension d'une matrice de données tout en maximisant leur variance. Elle transforme les données initiales de haute dimension en un ensemble de plus petite dimension composé de nouvelles variables qui sont des combinaisons linéaires des variables originelles.

Par ailleurs, cette méthode souffre de plusieurs limitations. L'une d'entre elles concerne sa nature linéaire. Vu que les connexions réseau peuvent avoir une structure non linéaire, le PCA classique ne sera pas très utile dans ce cas-là. Pour remédier à cette limitation, Kuang *et al.* [85] se sont beaucoup intéressé à l'utilisation de Kernel PCA (KPCA). Ils ont proposé d'utiliser KPCA avec la machine à vecteurs de support (SVM) et l'algorithme génétique (GA) en se basant sur un noyau gaussien. SVM a été utilisé comme étape de classification et GA pour optimiser le paramètre de SVM. Par ailleurs, le temps d'apprentissage de l'approche s'est avéré très long. Pour remédier a cela, les mêmes auteurs en 2014 ont présenté un noyau gaussien plus performant appelé (N-RBF) [84]. Cette dernière publication a raccourci le temps d'apprentissage et améliorer la performance de SVM. En 2015, ils ont combiné KPCA avec une méthode robuste d'optimisation par essaims particulaires (ICPSO) [86] pour améliorer davantage l'efficacité de l'IDS. Récemment, Chen *et al.* [33] ont proposé une méthode appelée local KPCA (LKPCA) qui est plus performante que le KPCA classique.

Toutefois, ces contributions utilisent souvent les mêmes noyaux, à savoir, le noyau gaussien et le noyau polynomial. Dans certains cas, ces noyaux peuvent s'avérer inefficaces en terme de taux

de detection. Afin de dépasser cette limite, nous proposons d'introduire de nouveaux noyaux dans la formulation de KPCA [49].

Une autre limitation de PCA concerne sa sensibilité aux données aberrantes (outliers). Ceci vient du fait que la formulation mathématique de l'approche utilise la norme euclidienne appelé (L2-norm). Cette norme amplifie l'effet des données aberrantes et mène à un taux de détection erroné. Pour résoudre ce problème, plusieurs variantes de PCA ont été développées [87, 88, 127, 172] et testé dans le domaine de reconnaissance faciale.

Ces méthodes proposées n'ont jamais été exploité dans la détection des intrusions. De ce fait, on propose dans un premier temps d'utiliser PCA Lp [87] dans le contexte de la détection des intrusions et de le comparer au PCA classique. Dans un deuxième temps, on met en avant une amélioration de PCA Lp qui consiste à introduire un algorithme itératif basé sur le gradient conjugué [50].

Malgré l'efficacité de ces variantes de PCA face aux données aberrantes, selon leurs formulations mathématiques, ils exploitent encore la moyenne utilisée dans PCA. Ce qui rend la formulation incohérente et peut mener à des résultats falsifiés. Pour remédier à cela, l'approche [123] a vu le jour. Cette méthode se base sur la décomposition en valeurs singulières (SVD) pour calculer la moyenne optimale. Toutefois, le processus employé s'avère très consommable en terme de temps CPU. Dans cette thèse, on propose l'approche QR-Optimal mean PCA(QR-OMPCA). Cette dernière utilise un processus rapide qui permet de calculer la moyenne optimale en utilisant la décomposition QR.

Chaque section décrit une approche proposée : PCA, KPCA, PCA-lp et QR-OMPCA. Par la suite, on valide ces approches par des simulations sur les bases de données KDDcup99 et NSL-KDD.

4.2 PCA et Kernel PCA

Dans cette section, nous présentons la formulation mathématique des méthodes PCA et KPCA.

4.2.1 PCA

L'analyse en composantes principales (PCA) est une technique mathématique qui transforme un certain nombre de variables corrélées en un certain nombre de variables non corrélées appelées composantes principales (PC). Généralement, le nombre de ces composantes principales est inférieur ou égal au nombre de variables d'origine. L'objectif principal de l'analyse en composantes principales est de réduire la dimension de l'ensemble de données initial, tout en conservant autant que possible la variance présente dans cet ensemble. Ceci est réalisé en ne prenant en compte que les premiers (PC), de sorte qu'ils conservent la plus grande partie de la variance présente dans toutes les variables d'origine. Supposons que nous avons des données d'apprentissage $X = [x_1, \ldots, x_M] \in \mathbb{R}^{n \times M}$, tel que n et M sont la dimension et le nombre des vecteurs respectivement.

Pour obtenir $n'(n' << n)$ composantes principales (PC), on se base sur la étapes suivantes [76] :

1. Calcul de la moyenne σ de l'ensemble :

$$\sigma = (\frac{1}{M}) \sum_{i=1}^{M} x_i \tag{4.1}$$

2. Soustraction de la moyenne σ de x_i pour obtenir ρ_i :

$$\rho_i = x_i - \sigma \tag{4.2}$$

3. Calcul de la matrice de covariance C sachant que :

$$C_{n\times n} = (\frac{1}{M}) \sum_{i=1}^{M} \rho_i \rho_i^T = AA^T \tag{4.3}$$

et

$$A_{n\times M} = (\frac{1}{\sqrt{M}}) \rho_i \tag{4.4}$$

4. Obtention des vecteurs propres U_k de la matrice C en utilisant :

$$CU_k = \lambda_k U_k \tag{4.5}$$

5. Trie des valeurs propres (et les vecteurs propres correspondants) dans l'ordre décroissant et choix des premiers vecteurs propres. Ces derniers sont les composantes principales (PC_i). Pratiquement, le nombre de PC choisi dépend de la précision explicitement exprimée par

$$\tau = \frac{\sum_{i=1}^{n'} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{4.6}$$

Le rapport τ définit le taux d'information conservé à partir des données d'origine par rapport aux valeurs propres n correspondantes. Finalement, la projection d'un nouveau vecteur x_{new} dans l'espace construit par les composantes principales peut être obtenue par :

$$t_i = PC_i^T x_{new} \tag{4.7}$$

4.2.2 Kernel PCA

Kernel PCA a été proposé par Schôlkopf et al. [141] comme extension non linéaire de PCA. Cette méthode utilise une fonction de mappage Φ qui transforme tous les vecteurs de données vers un espace F de grande dimension comme suit :

$$\Phi : x_i \in R^n \rightarrow \Phi(x_i) \in F$$

ou $\Phi(x_i)$ est un vecteur de F et $\sum_{i=1}^{M} \Phi(x_i) = 0$. Le mappage de x_i est notée comme $\Phi(x_i) = \Phi_i$ et la matrice de covariance dans l'espace F est construite de la manière suivante:

$$C = (\frac{1}{M}) \sum_{i=1}^{M} (\Phi_i - mean)(\Phi_i - mean)^T \tag{4.8}$$

Ou $mean = \sum_{i=1}^{M} \frac{\Phi_i}{M}$. La matrice de covariance C peut être diagonalisée de la manière suivante :

$$Cv = \lambda_i v \tag{4.9}$$

On remarque que chaque vecteur propre v de C peut être exprimé comme :

$$v = \sum_{i=1}^{M} (\alpha_i \Phi_i) \tag{4.10}$$

Pour obtenir les coefficients α_i, on definit une matrice K de taille $M \times M$. Les éléments de cette dernière sont déterminés comme suit :

$$K_{ij} = \Phi_i . \Phi_j = k(x_i, x_j) \tag{4.11}$$

Ou $k(x_i, x_j) =< \Phi_i, \Phi_j >$ est le produit scalaire de deux vecteurs de F. Si les données projetées $\Phi(x_i)$ n'ont pas une moyenne nulle, on peut travailler avec la matrice de Gram K' au lieu de K. La matrice K' est definie comme suit :

$$K' = K - 1_M K - K 1_M + 1_M K 1_M \tag{4.12}$$

Tel que $1_M = (1/M)_{M \times M}$. Afin de resoudre le problème des valeurs propres illustré par l'equation (4.9), on reformule l'équation de la même manière que celle définie en [141]

$$K' \alpha = M \lambda \alpha \tag{4.13}$$

Les vecteurs colonnes α_i sont les vecteurs propres orthonormaux de K' correspondants aux p plus grands valeurs propres positives $\lambda_1 \geq \lambda_2 \geq .. \geq \lambda_p$. Ainsi les vecteurs propres orthonormaux v_i de C peuvent être exprimés par :

$$v_i = (\frac{1}{\sqrt{\lambda_i}}) \Phi_i \alpha_i \tag{4.14}$$

La projection d'un nouveau vecteur x_{new} dans F se fait par :

$$t_{new} = (v_1, v_2, ..., v_p)^T \Phi(x_{new}) \tag{4.15}$$

qui peut s'ecrire comme suit :

$$t_{new} = v_i^T \Phi(x_{new}) = (\frac{1}{\sqrt{\lambda_i}}) \alpha_i^T k(x_i, x_{new}) \tag{4.16}$$

Il doit être noté que la matrice K peut être construite directement à partir des données d'apprentissage. Les fonctions à noyau (kernel) souvent utilisées sont:

⋆ Noyau Gaussien (Gaussian kernel) :

$$k(x,y) = e^{(\frac{-\|x-y\|^2}{2\times sigma^2})},\ sigma \in N \tag{4.17}$$

⋆ Noyau Polynomial (Polynomial kernel) :

$$k(x,y) = (x^T y + 1)^d,\ d \in N \tag{4.18}$$

On a proposé d'utiliser d'autres fonctions à noyau n'ayants pas eu assez d'attention dans la communauté scientifique. Ces fonctions sont:

⋆ Noyau à puissance (Power kernel) :

$$k(x,y) = \|x-y\|^d,\ d \geq 1 \tag{4.19}$$

⋆ Noyau à puissance rationnel (Rational Power kernel) :

$$k(x,y) = \|x-y\|^d,\ 0 < d < 1 \tag{4.20}$$

⋆ Noyau Logarithmique (Log kernel) :

$$k(x,y) = -log(\|x-y\|^d + 1),\ d \geq 1 \tag{4.21}$$

⋆ Noyau Sphérique (Spherical kernel) :

$$k(x,y) = 1 - \frac{3}{2}(\frac{\|x-y\|}{d}) + \frac{1}{2}(\frac{\|x-y\|}{d})^3\ ,d \geq 1 \tag{4.22}$$

4.2.3 Exemple illustratif de KPCA

Prenons le cas de deux classes qui ne sont pas linéairement séparables dans l'espace initial des données. Si on choisit une fonction de transformation $\Phi : x_i \in R^2 \rightarrow \Phi(x_i) \in R^3$. les données projetées $\Phi(x_i)$ peuvent être linéairement séparables dans R^3, ce qui correspond à une règle de décision non linéaire (ellipsoïde) dans l'espace initial des données (figure 4.1).

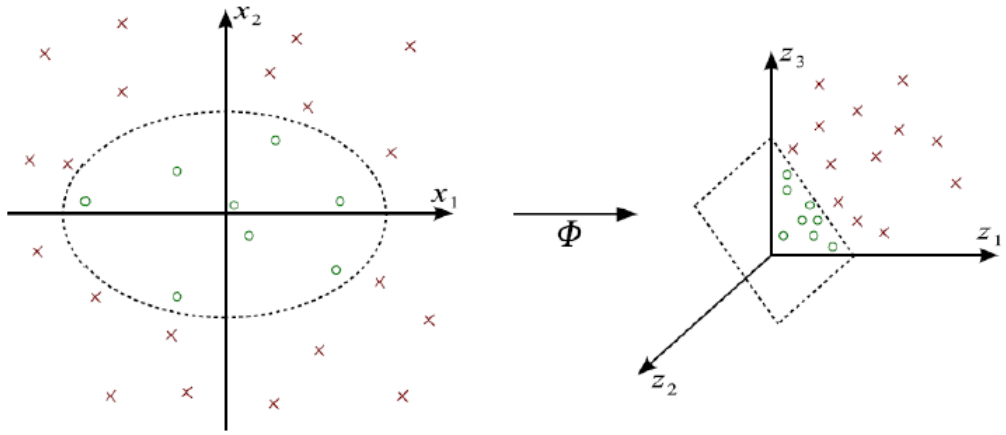


Figure 4.1: *Séparation linéaire après transformation de deux classes non linéairement séparables dans l'espace initial.*

Comme on peut le constater, il est impossible de trouver une droite séparant les deux classes dans l'espace initial à deux dimensions. Tandis que, si l'on considère la fonction :

$$\Phi : x_i \in R^2 \rightarrow \Phi(x_i) \in R^3$$

$$(x_1, x_2) \rightarrow (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

les données projetées dans le nouvel espace de redescription F engendré par la base (z_1, z_2, z_3) sont telles que les deux classes sont linéairement séparables.

4.2.4 Résultats expérimentaux

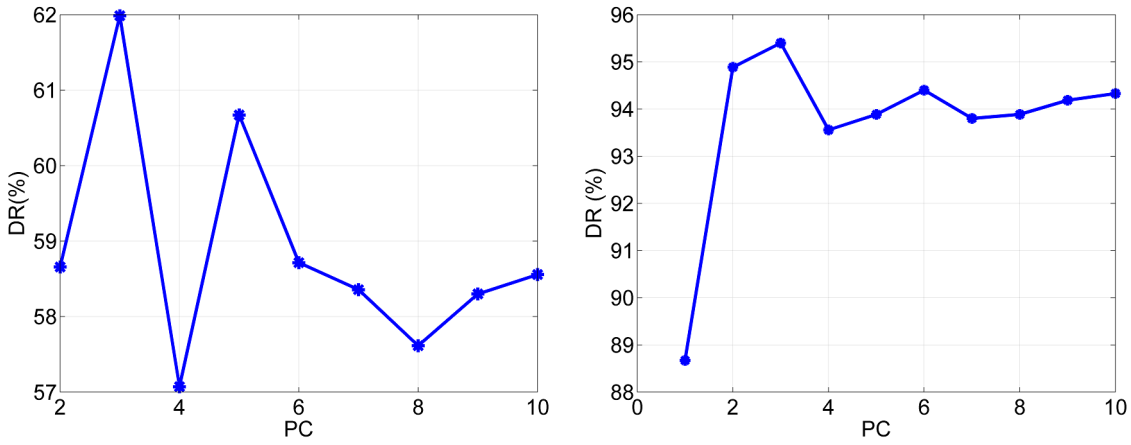
Pour KDDcup99, le protocole de test suivi est le suivant :

Dans la phase d'apprentissage, nous avons sélectionné au hasard 1000 connexions normales, 100 connexions de type DOS, 50 connexions de type U2R, 100 connexions de type R2L et 100 connexions de type PROBE à partir du "kddcup.data_10_percent". Durant la phase de test, on a sélectionné 100 connexions normales, 100 connexions DOS, 50 connexions U2R, 100 connexions R2L et 100 connexions PROBE à partir du "corrected".

En ce qui concerne NSL-KDD, on a repris le même protocole de test que celui utilisé avec KDDcup99. Sauf qu'au lieu de travailler avec deux sous bases de données, on a décidé de travailler avec tous les données de NSL-KDD.

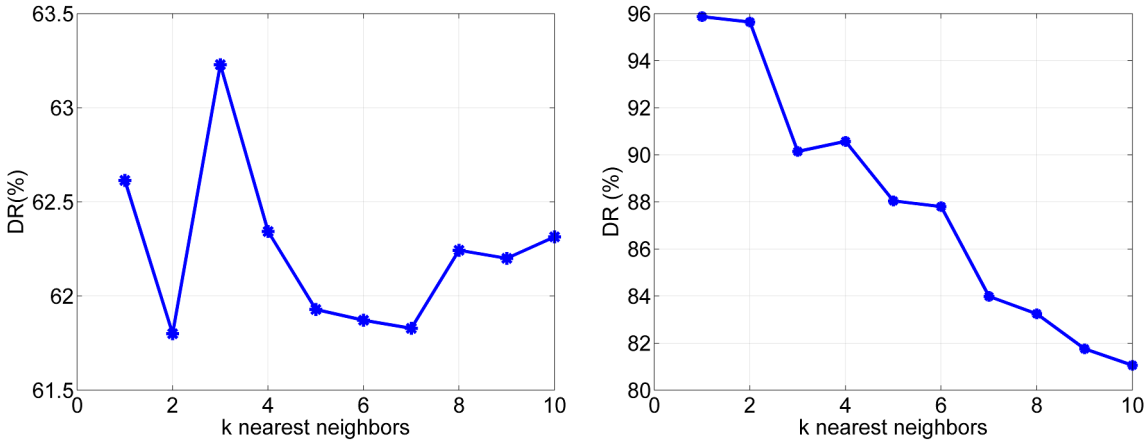
Le but de la première expérience est de trouver le nombre optimal de *PC* qui permet d'obtenir un meilleur taux de détection (DR). Pour cela, on commence par appliquer PCA sur les données d'apprentissage pour obtenir les composantes principales *PC*. Ensuite, on projette les données de test dans l'espace engendré par les différentes *PC*. Puis, on varie le nombre de ces derniers, et on visualise le comportement de (DR). On remarque à partir de la figure 4.2 que les 3 premières

composantes principales mènent a un DR maximale. Dans un autre calcul, ces *PC* donnent un taux d'inertie $\tau > 0,99$ (4.6). Ces deux résultats démontrent que seuls les trois premières *PC* méritent d'être retenu pour les deux bases de données.



(a): Expérience sur KDDcup99 *(b): Expérience sur NSL-KDD*

Figure 4.2: *Le taux de detection en fonction du nombre des composantes principales (PC).*



(a) *(b)*

Figure 4.3: *DR de PCA (%) en fonction du nombre des plus proches voisins k*

Une deuxième expérience tente de déterminer le nombre des *k* plus proches voisins qui donne le meilleur taux de détection. Pour cela, nous avons fixé le nombre de composantes principales à trois et nous avons varié le nombre *k* de 1 à 10. Pour KDDcup99, comme indiqué dans la figure 4.3, quand k = 3 le taux de détection est optimal. Alors que pour NSL-KDD, nous prenons seulement un seul voisin pour atteindre un taux maximal de DR. Dorénavant, toutes les prochaines expériences adopteront le *k* et le nombre de *PC* adéquats.

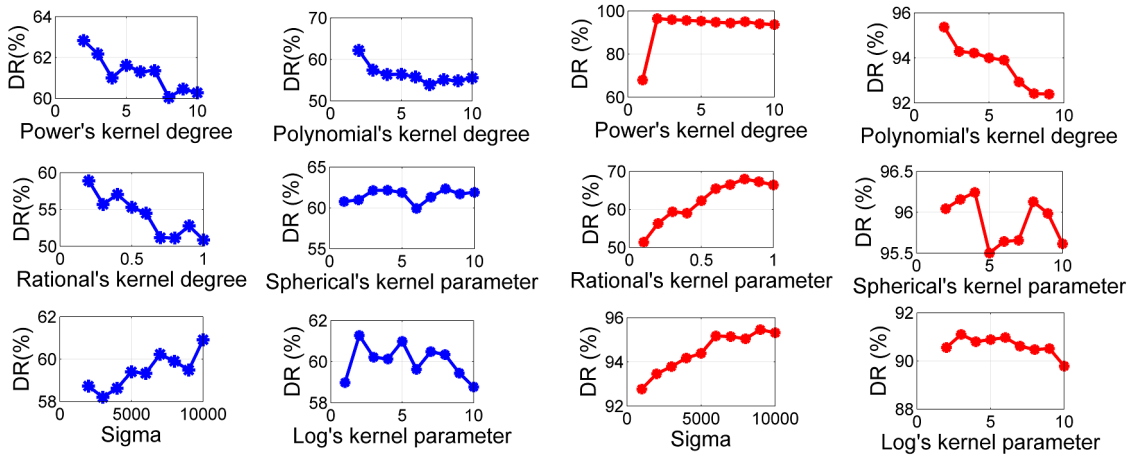
Le tableau 4.1 montre les taux de détections de chaque attaque. A partir de ce tableau, nous pouvons constater que pour KDDCup99, les deux catégories d'attaques DOS et PROBE sont dé-

tectés avec un taux de 95,13% et 69,8%. Contrairement aux attaques U2R et R2L qui sont détectées avec 8,13% et 3,5% respectivement. En ce qui concerne NSL-KDD, nous pouvons noter que DOS et les attaques PROBE sont fortement détectés avec un taux de 87,95% et 80,15%. De plus, U2R et R2L sont également bien identifiés avec les taux 98,80% et 93,85% respectivement.

Table 4.1: *Le taux de detection(%) de chaque attaque en utilisant PCA*

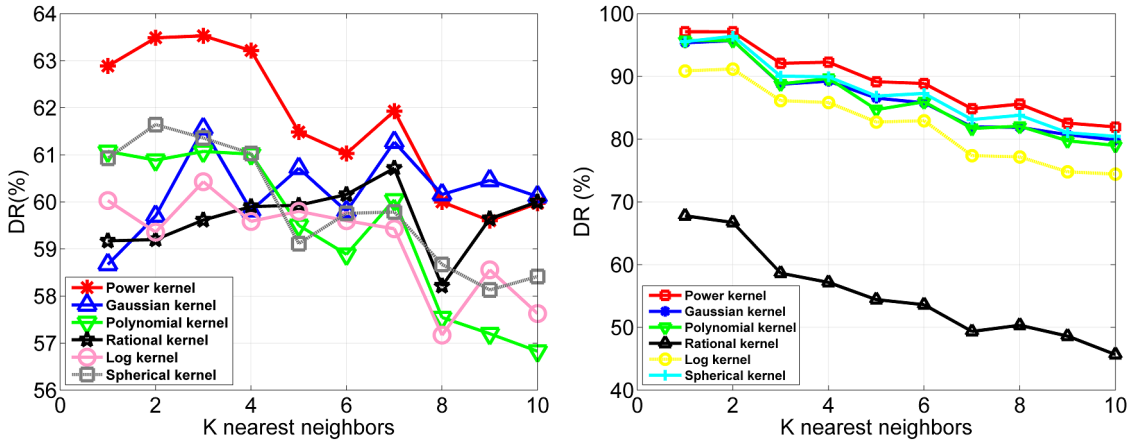
Base de données	DOS	U2R	R2L	PROBE
KDDcup99	95,1333	8,1333	3,5667	69,8667
NSL-KDD	87,95	98,80	93,85	80,15

Dans la troisième partie de nos expériences, nous évaluons l'efficacité de KPCA dans la détection des intrusions en suivant un processus bien déterminé. Pour commencer, nous exploitons six fonctions à noyaux décrits par les équations (4.17), (4.18), (4.19), (4.20), (4.21), (4.22). Pour chaque fonction noyau on cherche la valeur du paramètre qui contribue le plus à l'obtention d'un (DR) maximal. Afin d'achever cela, on varie les différents paramètres de chaque fonction et on observe le comportement du taux de détection. Comme c'est illustrée dans la figure 4.4, les résultats qui concernent KDDcup99 révèlent que les valeurs optimales de chaque paramètre sont les suivants : $d = 2$ pour le noyau polynomial, le noyau à puissance et le noyau logarithmique. $d = 0,2$ pour le noyau rationnel, $sigma = 10000$ pour le noyau gaussien et $d = 8$ pour le noyau sphérique. Les expériences sur NSL-KDD (figure 4.4) montrent que les paramètres qui mènent à une meilleur détection sont $d = 2$ pour le noyau polynomial, le noyau à puissance, $d = 3$ pour le noyau logarithmique, $d = 0.8$ pour le noyau rational, $sigma = 9000$ pour le noyau gaussien et $d = 4$ pour le noyau sphérique.



(a) *(b)*

Figure 4.4: *Effet des parametres des fonctions noyaux sur le DR.*



(a) *(b)*

Figure 4.5: *Comparaison des performances de différentes fonctions a noyau.*

La deuxième étape du processus a pour objectif l'identification de la fonction noyau la plus adapté a la détection des attaques. Pour atteindre cet objectif, les fonctions noyaux prendront comme valeurs de paramètres celles trouvées grâce à l'expérience précédente. Ensuite on compare les différentes fonctions en termes de DR. La figure 4.5 révèle que le noyau à puissance prend le dessus.

Une fois le noyau adéquat trouvé, on le compare à PCA. L'expérience sur kddcup99 révèle que KPCA basé sur le noyau a puissance dépasse PCA lorsque k est inferieur a 4 (figure 4.6). En regard de cela, PCA produit moins de faux positives (figure 4.6). L'expérience sur NSL-KDD démontre que le même noyau maintient sa supériorité face à PCA en termes de taux de détection (figure 4.6) et en réduction du taux des faux positives (figure 4.6).

Pour démontrer davantage l'efficacité de KPCA, on illustre à travers le tableau 4.2 le DR de chaque attaque (DOS, U2R, R2L and PROBE). On remarque d'une part que les deux catégories DOS et PROBE sont détectées avec les taux 96,13% et 73,8%. Des taux bien meilleurs que celles de PCA (95,13% and 69,8%). D'autre part on remarque que U2R et R2L sont mal détectées. En effet, le taux d'identification de U2R est seulement 9,93% (un peu mieux que celui de PCA 8,13%). On peut dire la même chose pour R2L (3,46%). Pour NSL-KDD, les DR des attaques individuelles de KPCA restent supérieurs par rapport à celles de PCA.

Table 4.2: *DR(%) individuelle de chaque attaque en utilisant KPCA*

Base de données	DOS	U2R	R2L	PROBE
KDDcup99	96,133	9,9333	3,4667	73,2667
NSL-KDD	93	99	95,30	89.05

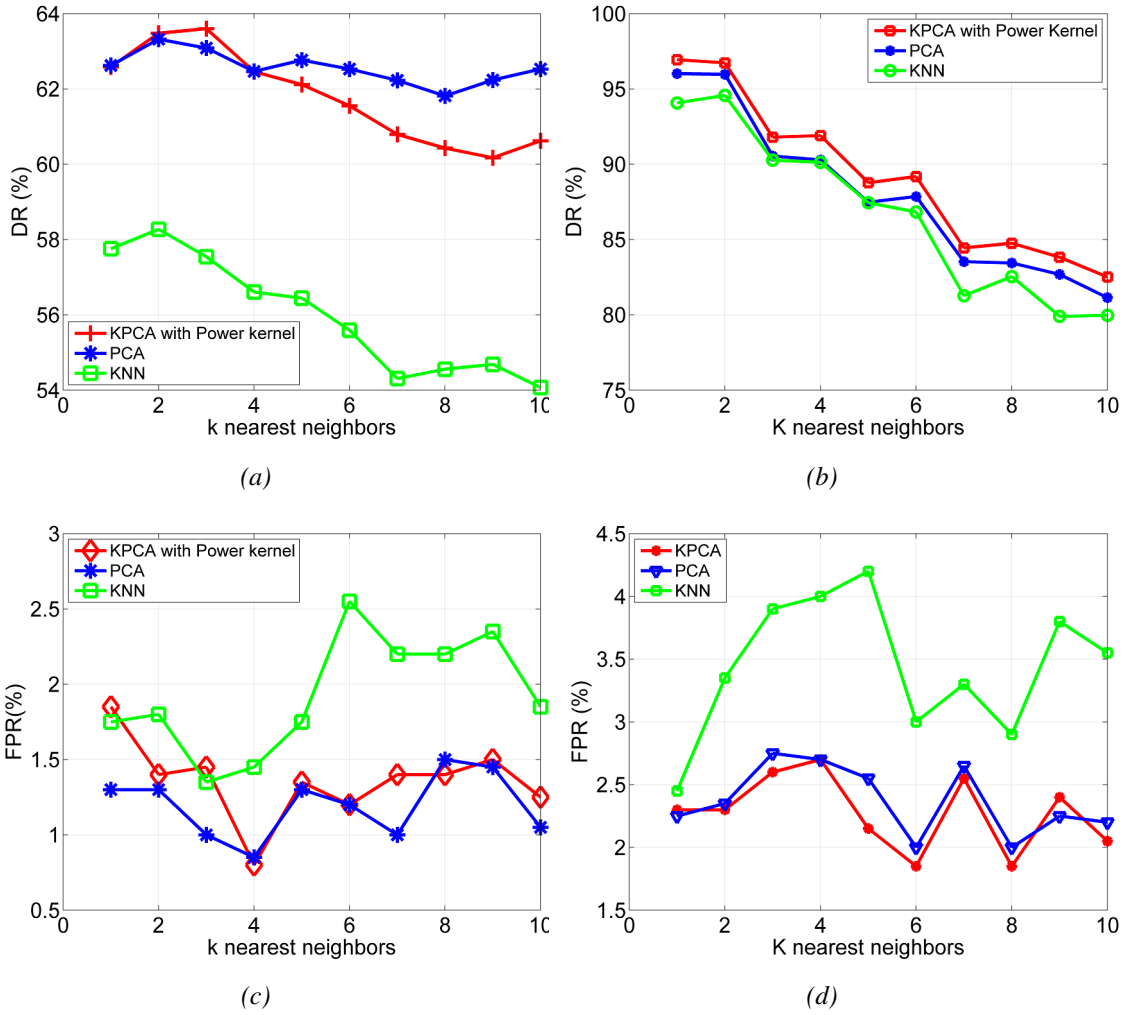


Figure 4.6: *DR(%) et FPR (%) de KPCA, PCA et KNN vs. k*

A ce stade de nos expériences, au lieu d'utiliser KNN comme classifieur, nous employons l'arbre de décision. Ceci dit, nous reprenons les deux expériences précédentes et cherchons le noyau optimal. La figure 4.7 montre que les noyaux sphérique et a puissance dépassent les autres noyaux pour les deux bases de données.

Le tableau 4.3 montre le taux de détection de chaque attaque. On peut conclure que KPCA avec le noyau adéquat détecte DOS et PROBE mieux que PCA.

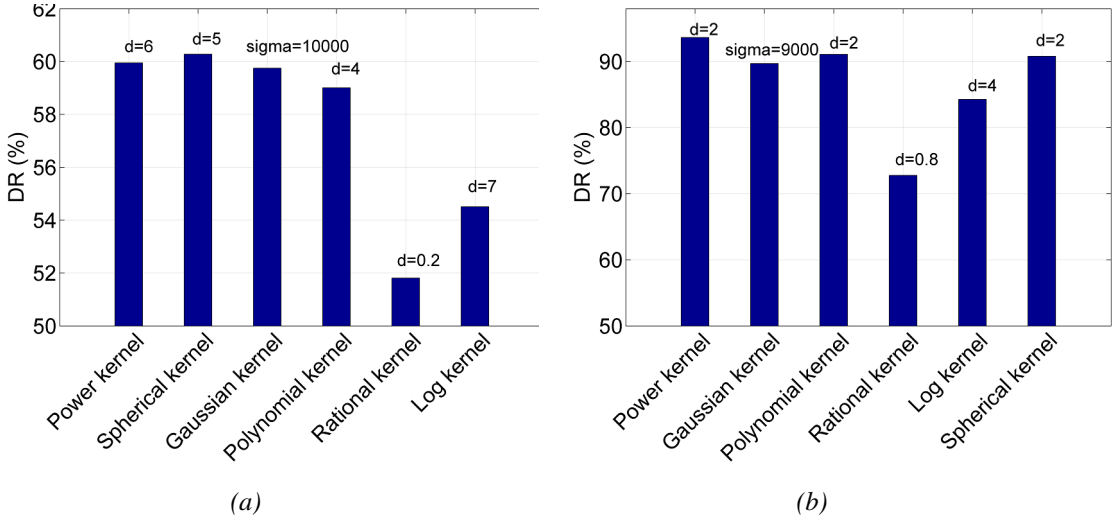


Figure 4.7: *Comparaison de différentes fonctions a noyau en utilisant l'arbre de décision*

Table 4.3: *Taux de detection d'attaques (%) en employant PCA et KPCA avec l'arbre de décision*

Base de données	The method	DOS	U2R	R2L	PROBE
KDDcup99	PCA	81,7	21,6	2,3	70,5
	KPCA	93,7	13,7	3,15	78
NSL-KDD	PCA	90,35	93,6	87,2	85,15
	KPCA	90,2	92,6	87,25	85,45

La dernière expérience illustrée par la figure 4.8 confirme les hypothèses suivantes : en utilisant l'arbre de décision comme classifieur, KPCA basée sur le noyau sphérique dépasse PCA en termes de DR quand le nombre de (PC) est compris entre 1 et 10. Néanmoins, PCA reste meilleur au niveau de production des faux positives. Ceci concerne KDDcup99. En se basant sur NSL-KDD, on constate que le KPCA basé sur le noyau à puissance donne à peu près les mêmes résultats que PCA.

En conclusion, nous avons proposé d'introduire deux nouvelles fonctions à noyau, jamais utilisé dans la littérature. A savoir, le noyau a puissance et le noyau sphérique. Différentes expériences sur KDDcup99 et NSL-KDD ont prouvé que le KPCA basé sur le noyau à puissance dépasse les autres noyaux classiques. En particulier, lors de l'emploi de KNN dans la phase de classification. En plus, on a noté une amélioration en termes de détection de certains attaques comme DOS et PROBE par rapport à PCA. Lors de l'utilisation de l'arbre de décision comme classifieur, on trouve que le KPCA basé sur le noyau sphérique prend le dessus.

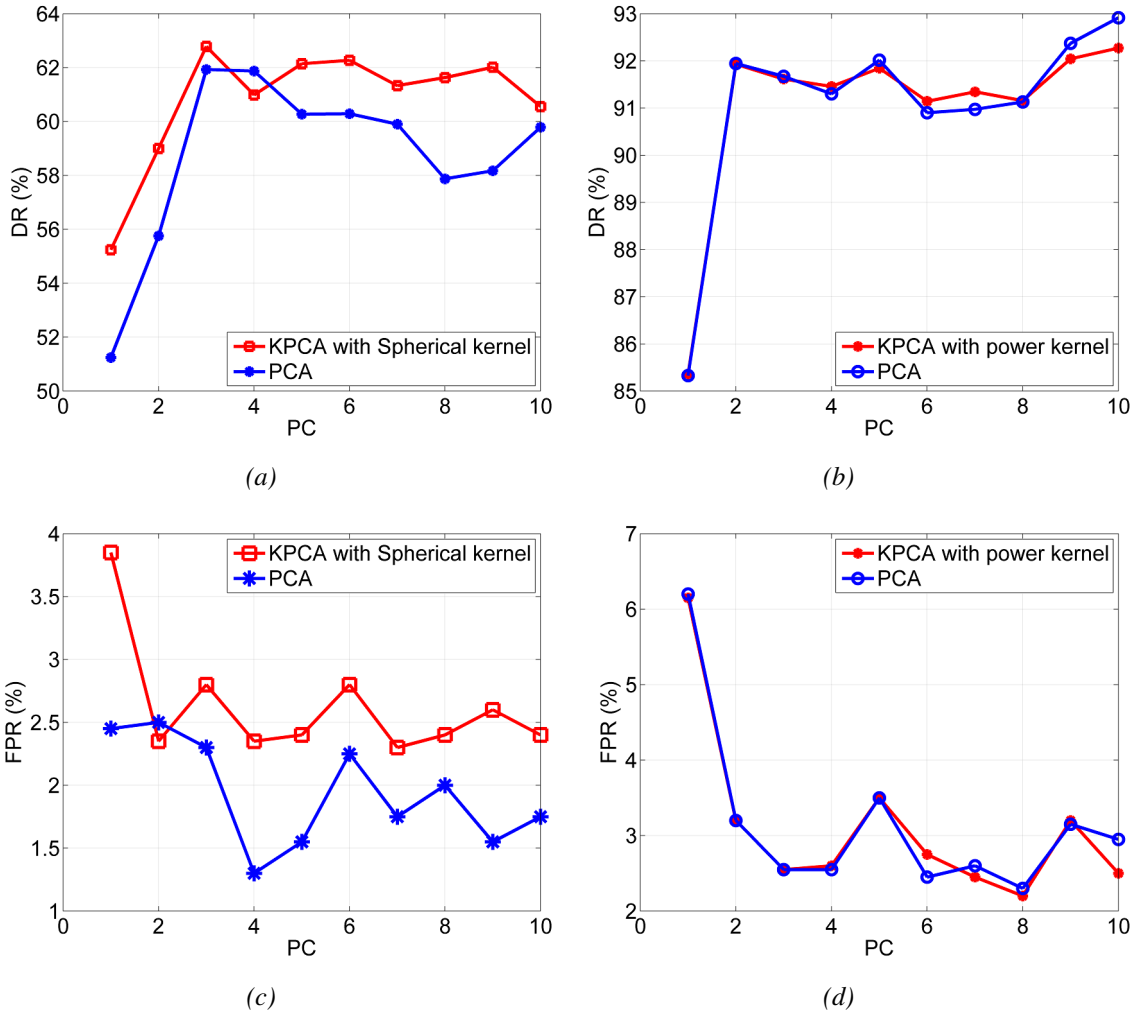


Figure 4.8: *DR(%) et FPR(%) de KPCA et PCA sous différentes dimensions en utilisant l'arbre de décision*

4.3 PCA-Lp basé sur le gradient conjugué

Dans cette section, pour traiter la dernière limitation, on propose dans un premier temps d'utiliser PCA-Lp [87] dans le contexte de la détection des intrusion et la comparer au PCA classique. Dans un deuxième temps, on met en avant une amélioration de PCA Lp qui consiste à introduire un algorithme itératif basé sur le gradient conjugué.

4.3.1 Préliminaires

Une autre formulation de PCA classique consiste à trouver n' vecteurs de projections orthonormaux $W \in \mathbb{R}^{n \times n'}$, tel que $W^T W = I_{n'}$, et la fonction suivante soit maximisée.

$$F_2(W) = \frac{1}{2}\sum_{i=1}^{M} ||W^T x_i||_2^2 = \frac{1}{2} tr(W^T S W) \tag{4.23}$$

Noter que $I_{n'}$ est la matrice identité de taille $n' \times n'$, $||.||_2$ est la norme L2 (L2-norm) d'un vecteur, $S = XX^T$ représente la matrice de dispersion, et tr(.) est l'opérateur de trace. La solution globale

de (4.23) est obtenue en résolvant $SW = W\Lambda$, ou Λ est une matrice diagonale contenant les valeurs propres de S.

L'equation (4.23) est dominée par les vecteurs ayants une grande norme. Ce fait rend le processus d'obtention des vecteurs de projections orthonormaux imprécis. Afin de contourner cette limite, on propose d'utiliser la fonction objective basée sur la norme Lp (Lp-norm) au lieu de la norme L2 (L2-norm) introduite par [87].

$$F_p(W) = \frac{1}{p}\sum_{i=1}^{M} ||W^T x_i||_p^p = \frac{1}{p}\sum_{i=1}^{M}\sum_{j=1}^{n'} |w_j^T x_i|^p \tag{4.24}$$

$W = [w_1, \ldots, w_{n'}] \in \mathbb{R}^{n\times n'}$ est la matrice de projection. Il faut garder en esprit que la fonction objective (4.23) est identique a (4.24) lorsque $p = 2$. Par ailleurs, l'obtention des solutions W demeure une tâche difficile.

Solution pour (n'=1)

Pour rendre le problème (4.24) plus gérable, comme première étape on cherche juste un seul vecteur de projection (n'=1). Ensuite (4.24) devient:

$$F_p(w) = \frac{1}{p}\sum_{i=1}^{M} |w^T x_i|^p \tag{4.25}$$

On résout ce problème d'optimisation en utilisant le gradient de $F_p(w)$. Cependant, on introduit une fonction de signe $s()$ qui remplace l'operateur de la valeur absolue. Cette fonction est définie de la manière suivante :

$$s(a) = \begin{cases} 1 & a \succ 0 \\ 0 & a = 0 \\ -1 & a \prec 0 \end{cases} \tag{4.26}$$

Ceci nous permet de réécrire $F_p(w)$ en (4.25) de la manière suivante:

$$F_p(w) = \frac{1}{p}\sum_{i=1}^{M} [s(a_i)a_i]^p, \quad a_i = w^T x_i \tag{4.27}$$

Le calcul du gradient ∇_w de $F_p(w)$ en respectant w donne :

$$\begin{aligned} \underset{w}{\nabla} &= \frac{dF_p(w)}{dw} = \sum_{i=1}^{M} \frac{dF_p(w)}{da_i}\frac{da_i}{dw} \\ &= \sum_{i=1}^{M} [s(a_i)a_i]^{p-1}\left[s'(a_i)a_i + s(a_i)\right] \\ &= \sum_{i=1}^{M} s'(a_i)s^{p-1}(a_i)a_i^p x_i + \sum_{i=1}^{M} s^p(a_i)a_i^{p-1}x_i \\ &= 2\sum_{i=1}^{M} \delta(a_i)s^{p-1}(a_i)a_i^p x_i + \sum_{i=1}^{M} s(a_i)\,|a_i|^{p-1}x_i \end{aligned} \tag{4.28}$$

Tel que $\delta()$ représente la distribution de Dirac et $s'()$ exprime la derivée partielle de $s()$. Le premier terme égale zero si $a_i \neq 0$ pour tout x_i. Après on obtient :

$$\nabla_w = \sum_{i=1}^{M} s(w^T x_i) \left| w^T x_i \right|^{p-1} x_i \tag{4.29}$$

Un cas particulier prend place lorsque $p \leq 1$. Dans ce cas, le gradient n'est pas bien défini pour w s'il existe quelques x_i tel que $w^T x_i = 0$. Afin d'éviter cette situation, on déplace d'un cran w. Il faut noter que telle action n'est valable qu'avec un nombre de vecteurs limité. La méthode du gradient est utilisée pour obtenir une projection qui maximise la fonction objective (4.25). L'entière procédure d'optimisation est représentée par l'algorithme PCA-Lp(G) [87] suivant :

Algorithm 1 : L'algorithme PCA-Lp

1. Initialisation: $t \leftarrow 0$. Choisir $w(0)$ tel que $||w(0)||_2 = 1$. ε=0.001.
2. Test de singularité (appliqué seulement si $p \leq 1$)
   - Si $\exists i$, tel que $w^T(t)x_i = 0$, $w(t) \leftarrow (w(t) + \eta)/||w(t) + \eta||_2$. η est un vecteur aléatoire.
3. Calcul du gradient ∇_w en utilisant (4.29).
4. Mise à jour de w : $w(t+1) \leftarrow w(t) + \alpha \nabla_w$, tel que α représente le taux d'apprentissage (learning rate).
5. Normalisation.
   - $t \leftarrow t + 1$
   - $w(t) \leftarrow \frac{w(t)}{||w(t)||_2}$
6. Test de convergence.
   - Si $||w(t) - w(t-1)||_2 > \varepsilon$ aller à l'étape 2.
   - Sinon, $w^* \leftarrow w(t)$. Arrêter l'itération.

La méthode du gradient trouve une grande difficulté pour converger si α est grand, tandis qu'avec des valeurs plus petites, la convergence n'est pas suffisamment rapide. Nous avons proposé de donner à α la valeur $\frac{10}{M}$, tel que M est le nombre des vecteurs d'apprentissage.

Solution pour $(n' > 1)$

Dans cette sous section, l'algorithme PCA-Lp est généralisé pour extraire plusieurs caractéristiques $(n' > 1)$. La méthode proposée est illustrée par Algorithme 2.

L'une des principales faiblesses du gradient simple (SG) est sa lenteur, ce fait vient de la stratégie de recherche implémentée dans cette méthode. Puisque le gradient suit une recherche en ligne droite, il ne s'arrête pas tant que la ligne n'est pas parallèle à la ligne du contour de la

Algorithm 2 : L'algorithme PCA-Lp ($n' > 1$)

1. Soit $w_0 = 0$ et $X_0 = X$.
2. Pour i = 1..n'
   - Considérer $X_i = (I_d - w_{i-1}w_{i-1}^T)X_{i-1}$
   - Appliquer PCA-Lp sur X_i.
3. Avoir comme sortie $W^* = [w_1, ..., w_{n'}]$

surface de la fonction (figure 4.9). La question se pose alors de savoir si nous pouvons arrêter et modifier la direction du gradient avant qu'elle ne devienne parallèle. Dans le gradient simple, il est difficile de faire un tel arrêt. Ce dernier postulat pose un problème important et entraîne une lente maximisation de la fonction.

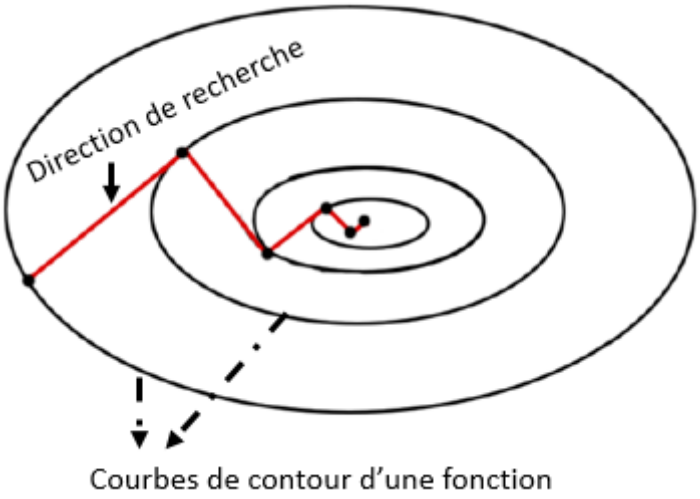


Figure 4.9: *Le contour de la surface d'une fonction avec les directions du gradient*

Pour traiter le problème du gradient simple, nous proposons d'utiliser le gradient conjugué (CG).

4.3.2 La solution proposée

Le gradient conjugué a vu le jour grâce à deux recherches importantes. La première est faite par Cornelius Lanczos et Magnus Hestenes à l'Institut de l'analyse numérique (National Applied Mathematics Laboratories of the United States National Bureau of Standards in Los Angeles). La seconde est réalisée par Eduard Stiefel chez (Eidg. Technische Hochschule Zurich). Dans le travail de Hestenes et Stiefel, l'algorithme est considéré comme une méthode pour résoudre des systèmes linéaires définis positifs et symétriques. En 1964, le champ d'application de cette méthode a été étendu aux problèmes non linéaires avec la recherche pionnière de Fletcher et Reeves [55].

Parmi les avantages les plus notables du gradient conjugué on cite la faible exigence de la mémoire et la vitesse de convergence. Ceci dit, notre algorithme (Algorithme 3) est inspiré par [55] et ressemble à la procédure du gradient simple.

Algorithm 3 : PCA-Lp basé sur le gradient conjugué

1. Initialisation: $t \leftarrow 0$. Considérer $w(0)$ tel que $||w(0)||_2 = 1$, $d(-1) = 0$, $b = 1$. d est un vecteur directeur. ε=0.001.
2. Test de singularité (appliqué seulement si $p \leq 1$)
   - Si $\exists i$, tel que $w^T(t)x_i = 0$, $w(t) \leftarrow (w(t) + \eta)/||w(t) + \eta||_2$. η est un vecteur aléatoire.
3. Calcul du gradient ∇_w en utilisant (4.29).
4. Calcul de β: $\beta = \frac{||\nabla_w||_2^2}{b}$
5. Obtention du vecteur directeur: $d(t) \leftarrow \nabla_w + \beta.d(t-1)$
6. Mettre à jour w : $w(t+1) \leftarrow w(t) + \alpha.d(t)$.
7. $b = ||\nabla_w||_2^2$
8. Normalisation.
   - $t \leftarrow t + 1$
   - $w(t) \leftarrow \frac{w(t)}{||w(t)||_2}$
9. Test de convergence.
   - Si $||w(t) - w(t-1)||_2 > \varepsilon$ aller a l'etape 2.
   - Sinon, $w^* \leftarrow w(t)$. Arrêt de l'iteration.

Deux paramètres critiques créent la différence par rapport au gradient simple et sont requis dans le processus de maximisation. Le premier est β et le second s'appelle le vecteur de recherche directeur $d(t)$. Pour le générer, il suffit de connaître trois vecteurs: le gradient actuel, le gradient précédent et le vecteur de recherche directeur precedent $d(t-1)$.

Chaque mise à jour du vecteur de projection w maximise $F_p(W)$, par conséquent choisir un $w(0)$ initial devient définitivement critique. Dans notre approche, nous correspondons le vecteur initial $w(0)$ au vecteur produit par le PCA classique. Nous pouvons aussi essayer d'autres techniques comme ré-exécuter l'algorithme CG plusieurs fois avec divers $w(0)$ et choisir le meilleur.

4.3.3 Résultats experimentaux

Cette section présente et analyse les résultats obtenus lors de l'utilisation de PCA-Lp qui exploite le gradient simple (SG), ensuite, on teste l'approche optimisée basée sur le gradient conjugué (CG).

Dans la première expérience, nous avons sélectionné aléatoirement comme données d'apprentissage 1000 connexions normal, 100 DOS, 50 U2R, 100 R2L, et 100 PROBE. Les données de test sont composées de 100 normal, 100 DOS, 50 U2R, 100 R2L, et 100 PROBE. La valeur de p prend 0.5, 1, 1.5.

La première expérience est semblable à celle illustrée à travers la figure 4.2. Elle cherche le nombre de (PC) qui mène à une détection optimale.

L'expérience sur KDDcup99 (les figures 4.10.a et 4.10.b) affirme clairement que trois composantes principales (PC) sont exactement le nombre que nous cherchons. L'expérience sur NSL-KDD retourne le même résultat sauf pour le cas de $(p = 1)$, où quatre PC sont nécessaires. Au passage, nous pouvons conclure que PCA Lp est beaucoup mieux que PCA classique spécialement lorsque p prend 1 et 1.5.

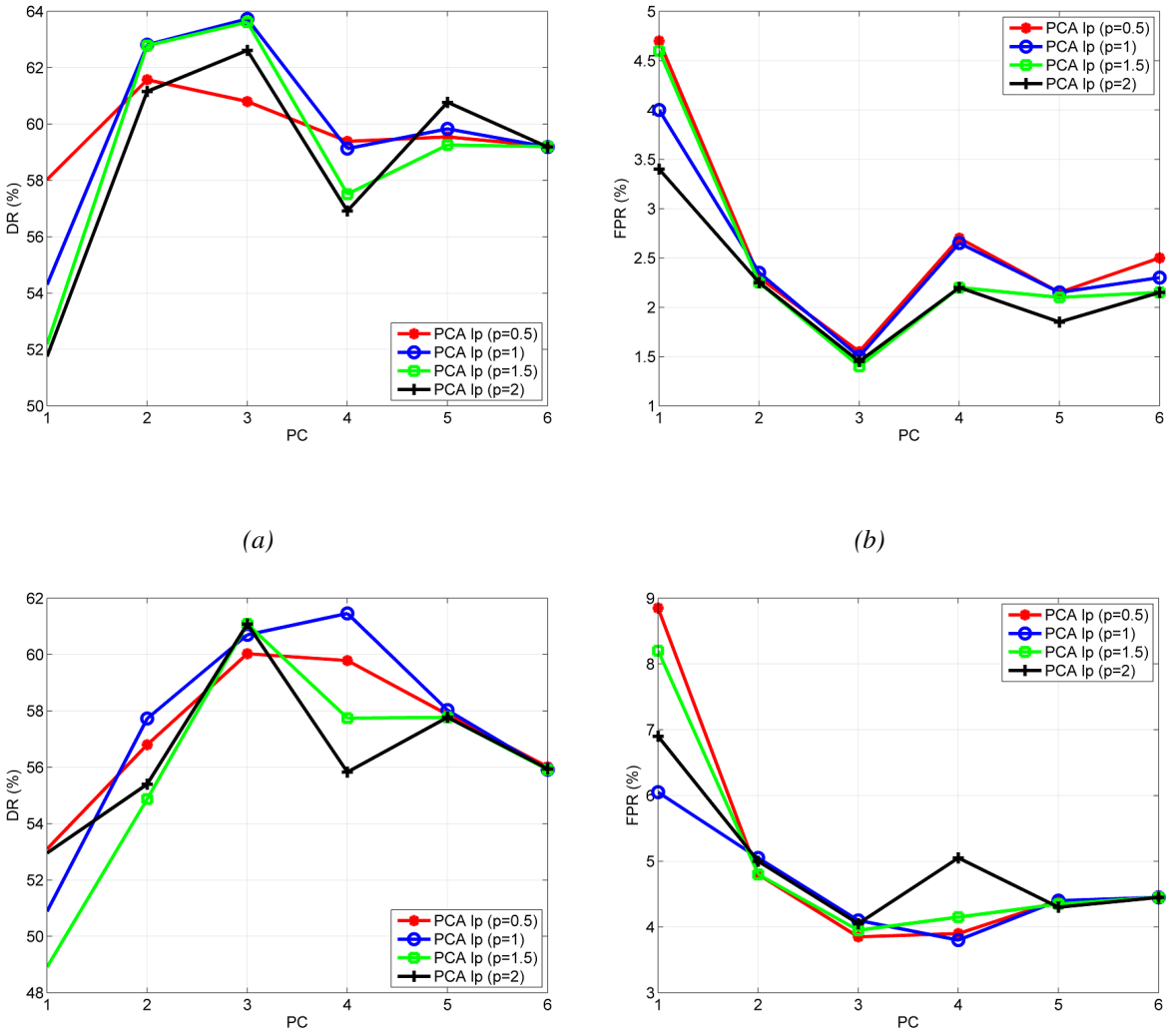
Nous avons aussi calculé le taux de détection de chaque type d'attaque (DOS, U2R, R2L, PROBE).

Table 4.4: *DR(%) individuelle de chaque attaque en utilisant PCA Lp*

Base de données	La methode utilisée	DOS	U2R	R2L	PROBE
KDDcup99	PCA L1	**93.65**	11.2	**4.2**	**75.5**
	PCA L1.5	93.15	11.1	4.2	70.05
	PCA L2	92.85	11	4.2	65.65
NSL-KDD	PCA L1	**74.6**	**11.5**	**15.8**	**56.3**
	PCA L1.5	73.4	11.4	15.8	53.4
	PCA L2	72.65	11.3	15.8	54.35

Selon le tableau 4.4, il est montré que l'IDS qui exploite PCA Lp détecte plus efficacement les attaques DOS et PROBE. les autres types d'attaques (U2R, R2L) sont identifiées avec un faible taux de détection quelque soit p. Ceci peut s'expliquer par le fait que KDDcup99 et NSL-KDD contiennent peu de connexions de ce type.

Dans une autre experience, nous varions le nombre de connexions d'apprentissage et nous observons le comportement du DR et FPR.



(a) *(b)*

(c) *(d)*

Figure 4.10: *PCA Lp appliquée sur les deux bases de données avec differentes valeurs de p*

Selon les tableaux 4.5 et 4.6, il est noté que les attaques DOS et PROBE sont encore bien détectées. Cependant, la faible identification des types U2R et R2L persiste. Les résultats obtenus confirment que PCA-Lp produit une meilleure détection des attaques et reduit efficacement le taux des fausses alarmes.

Table 4.5: *DR(%) vs. Données d'apprentissage de PCA-Lp concernant KDDcup99*

Données	PCA lp	Normal	DOS	U2R	R2L	PROBE	Temps CPU(s)
2650	$p=1$	98.75	**91.05**	**12.9**	**4.35**	**76.8**	1.05
	$p=1.5$	**98.80**	88.50	12.70	4.30	69.80	0.99
	$p=2$	98.70	89.05	12.60	4.30	64	0.06
3950	$p=1$	98.95	**93.7**	**12.10**	**4.2**	**77.15**	1.79
	$p=1.5$	99	89.75	11.8	4.15	69.10	1.40
	$p=2$	99.05	91.20	11.8	4.15	64.90	0.07
5250	$p=1$	**98.85**	**92.20**	**10.80**	**4.45**	**80**	2.20
	$p=1.5$	98.80	89.30	10.80	4.45	71.60	1.62
	$p=2$	98.75	89.20	10.80	4.45	65.35	0.09
6550	$p=1$	98.8	**92.95**	**9**	**3.8**	**80.3**	3.17
	$p=1.5$	**98.85**	89.75	9	3.75	75.50	2.77
	$p=2$	98.85	90.20	9	3.75	70.45	0.15
7850	$p=1$	**99.15**	**92.45**	**10.90**	**5.45**	**78.85**	3.76
	$p=1.5$	99.15	90.05	10.40	5.45	72.35	2.72
	$p=2$	99.15	90.15	10.30	5.45	66.95	0.15
9150	$p=1$	**99**	**91.95**	**9.9**	**5**	**79.55**	4.49
	$p=1.5$	98.95	88.25	9.7	4.95	70.75	3.67
	$p=2$	98.95	89.15	9.6	4.95	65.70	0.17
10450	$p=1$	**98.45**	**92.65**	**9.8**	**5.3**	**78.80**	5.02
	$p=1.5$	98.40	90.50	9.70	5.25	71.25	4.19
	$p=2$	98.35	91.10	9.50	5.20	65.40	0.19
11750	$p=1$	**98.85**	**94.30**	**9.80**	**4.80**	**79.80**	5.55
	$p=1.5$	98.85	93.15	9.30	4.75	73.60	4.60
	$p=2$	98.75	92.25	9.2	4.75	67.05	0.21

Malheureusement, nous avons noté que le temps d'execution consommé par PCA Lp basé sur le gradient simple (SG), peut être considéré comme sa principale faiblesse. Pour optimiser cela,

Table 4.6: *DR(%) vs. Données d'apprentissage de PCA-Lp concernant NSL-KDD*

Données	PCA lp	Normal	DOS	U2R	R2L	PROBE	Temps CPU(s)
2650	$p=1$	96.2	**68**	**10**	**14.25**	**58.6**	0.93
	$p=1.5$	96	67.2	9.5	14.2	52.15	0.78
	$p=2$	96.10	66.35	9.5	14.10	52.95	0.05
3950	$p=1$	96.4	**67.85**	**11.10**	**17.7**	**58.75**	1.51
	$p=1.5$	96.35	67.25	11.10	17.55	54.65	1.13
	$p=2$	96.4	66.20	11.10	17.5	53.95	0.075
5250	$p=1$	**95.95**	**68.30**	**11**	**17.7**	**60.9**	1.88
	$p=1.5$	95.95	68.10	10.70	17.4	58.10	1.4
	$p=2$	95.95	66.65	10.70	17.4	57.75	0.09
6550	$p=1$	**96.05**	69.65	**9.7**	**23.2**	**62.7**	2.46
	$p=1.5$	95.95	**69.75**	9.4	23.2	56.85	1.8
	$p=2$	95.95	68.55	9.4	23.15	55.25	0.13
7850	$p=1$	**95.6**	70.4	**9.4**	**19.05**	**63.15**	2.73
	$p=1.5$	95.55	**70.7**	9.1	18.1	59	1.8
	$p=2$	95.45	69.05	9	18.1	57.55	0.15
9150	$p=1$	**95.4**	**69.55**	**8.8**	**20.8**	**60.7**	3.75
	$p=1.5$	95.3	68.6	8.7	20.7	54.55	2.69
	$p=2$	95.25	67.4	8.4	20.7	53.05	0.20
10450	$p=1$	**95.9**	67.75	**7.3**	24.7	**63.2**	4.32
	$p=1.5$	95.7	**68.4**	7.3	**24.10**	58.9	2.42
	$p=2$	95.7	67.05	7.3	24.10	56.2	0.21
11750	$p=1$	**95.8**	67.5	**9.7**	25	**61**	4.40
	$p=1.5$	95.8	**68.3**	9.7	24.6	57	2.79
	$p=2$	95.8	66.7	9.6	24.6	55.9	0.23

nous proposons d'utiliser une nouvelle variante de PCA Lp qui repose sur le gradient conjugué (CG). Pour faire la difference entre les deux methodes, on utilise les notations: (SG Lp) pour l'ancienne solution, et (CG Lp) pour la solution proposée.

les figures 4.11.a et 4.11.b illustrent les résultats obtenus en considerant KDDcup99 comme base d'evaluation. Nous déduisons que l'utilisation du (CG Lp) maintient le taux de détection presque intact d'une part. D'autre part, il consomme moins de temps CPU.

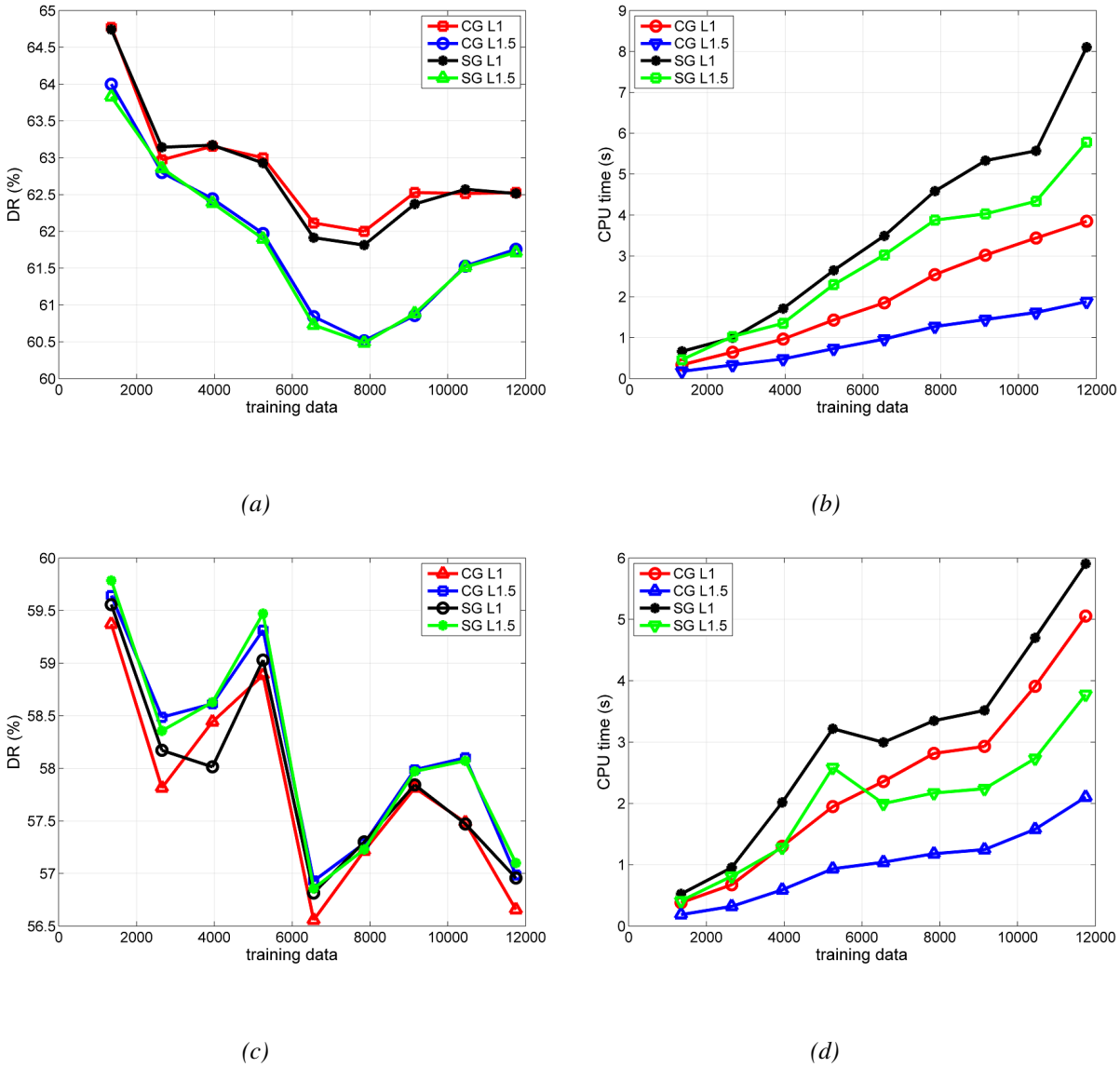


Figure 4.11: *(a): DR(%) vs. Données d'apprentissage pour KDDcup99*
(b): Temps CPU (s) vs. Données d'apprentissage pour KDDcup99
(c): DR(%) vs. Données d'apprentissage pour NSL-KDD
(d): Temps CPU (s) vs. Données d'apprentissage pour NSL-KDD

Pour plus de détails, le tableau 4.7 nous montre que la détection d'attaques individuelles pour $p = 1$ est presque similaire. Sauf pour ceux qui concernent DOS et PROBE. Lorsque le nombre de données d'apprentissage depasse (9150) nous observons que le gradient conjugué prend le dessus. Dans le cas où nous utilisons NSL-KDD, comme mentionné dans les figures 4.11.c et 4.11.d, nous ne remarquons aucune amélioration en terme d'identification d'attaque générale par rapport à SG Lp. Néanmoins, CG Lp consomme moins de temps.

Table 4.7: *DR(%) de chaque attaque appartenant à la base base KDDcup99 vs. Données d'apprentissage*

Données d'apprentissage	Type du gradient	Normal	DOS	U2R	R2L	PROBE
2650	CG L1	98.8	94.10	13.10	4	72.8
	SG L1	98.80	94.10	13.10	4	73.05
3950	CG L1	99.3	93.05	10.9	5	75.95
	SG L1	99.25	93.25	10.9	5	77.9
5250	CG L1	99.3	93.6	10.9	5	78.15
	SG L1	99.25	92.9	10.9	5	79.50
6550	CG L1	98.75	92.95	9.8	3.8	77.95
	SG L1	98.75	92.95	9.8	3.8	78
7850	CG L1	98.8	93.10	11.8	4.55	78.10
	SG L1	98.8	92.55	11.8	4.55	77.95
9150	CG L1	98.95	92.10	9.6	4.9	78.50
	SG L1	98.95	91.45	9.6	4.9	78.20
10450	CG L1	98.65	93.55	11.50	5.55	80.4
	SG L1	98.65	92.85	11.50	5.55	80.4
11750	CG L1	98.95	93.05	8.6	4.45	78.7
	SG L1	98.95	93.10	8.6	4.45	78.6

Table 4.8: *DR(%) de chaque attaque appartenant à la base base NSL-KDD vs. Données d'apprentissage*

Données d'apprentissage	Type du gradient	Normal	DOS	U2R	R2L	PROBE
2650	CG L1.5	95.95	69.65	11.40	13.2	54.35
	SG L1.5	95.95	69.60	11.40	13.2	54.25
3950	CG L1.5	96	68.9	10.30	13.45	52.9
	SG L1.5	96	68.8	11.40	13.45	52.25
5250	CG L1.5	95.95	68.6	10.40	20.2	52.85
	SG L1.5	95.95	68.55	10.40	20.2	52.45
6550	CG L1.5	96.90	69.25	9.3	17.2	54.95
	SG L1.5	96.90	69.2	9.3	17.2	54.9
7850	CG L1.5	95.55	68	10	20.4	58.25
	SG L1.5	95.55	67.9	10	20.4	57.75
9150	CG L1.5	95.85	69.05	9.4	22.25	57.80
	SG L1.5	95.85	69.35	9.4	22.25	58.05
10450	CG L1.5	96.30	67.85	9.2	21.4	58.90
	SG L1.5	96.30	68.15	9.2	21.4	58.65
11750	CG L1.5	95.75	68.6	8.4	23.50	59.70
	SG L1.5	95.75	68.8	8.4	23.50	59.40

En ce qui concerne la détection individuelle d'attaques. Le tableau 4.8 nous apprend que lorsque ($p = 1,5$), l'approche proposée surpasse légèrement l'ancienne solution au niveau de l'identification des attaques DOS et PROBE.

En conclusion, pour remedier au probleme des données aberrantes existantes dans un trafic

réseau, nous avons proposé un nouveau PCA-Lp Basée sur le gradient conjugué. cette technique tente de trouver des projections qui maximisent la covariance totale en utilisant la norme Lp au lieu de la norme L2. Les résultats expérimentaux ont montré que notre approche est plus rapide que PCA-Lp qui repose sur le gradient simple.

Malgré l'efficacité de l'approche face aux données aberrantes, selon sa formulation mathématique, pour centrer les données, la méthode exploite encore la moyenne utilisée dans PCA (4.1). Ce qui rend la formulation incomplete et peut mener à des résultats falsifiés. Afin de traiter ceci, la prochaine section aborde une nouvelle alternative.

4.4 QR-OMPCA

4.4.1 Optimal Mean PCA (OMPCA)

Une autre formulation de l'analyse en composantes principales (PCA) consiste a chercher la matrice des composantes principales W qui minimise l'entité suivante [159] :

$$\min_{rang(Z)=k} ||X-Z||_2^2 \tag{4.30}$$

Tel que $Z = WW^TX$.

Cette formulation suppose que X est centrée en utilisant la moyenne classique (4.1). Par ailleurs, l'emploi de ce type de moyenne dans PCA et ces variantes amplifie l'effet des données aberrantes. Pour remedier a ceci, l'article [123] introduit le processus du calcul de la moyenne optimale a l'interieur de la fonction objective. Ainsi, l'équation (4.30) sera reformulée de la manière suivante :

$$\min_{b,rang(Z)=k} ||X-b1^T-Z||_2 \tag{4.31}$$

Tel que 1 est un vecteur colonne ayant des uns comme éléments. b est une variable à optimiser. L'équation (4.31) peut être réécrite comme une somme:

$$\min_{b,W\in R^{n\times k},W^TW=I} \sum_{i=1}^{M} ||X-b-W(v^i)^T||_2 \tag{4.32}$$

En incluant $v^i = (x_i-b)^TW$ dans l'équation ci-dessus, on aura :

$$\min_{b,W\in R^{n\times k},W^TW=I} \sum_{i=1}^{M} ||(I-WW^T)(x_i-b)||_2. \tag{4.33}$$

L'équation peut être rapprochée par :

$$\min_{b,W\in R^{n\times k},W^TW=I} \sum_{i=1}^{M} ||d_{ii}(I-WW^T)(x_i-b)||_2^2. \tag{4.34}$$

Où d_{ii} représentent les éléments diagonaux d'une matrice pondérée D. Après plusieurs étapes décrites dans [123], le problème devient :

$$\max_{W\in R^{n\times k},W^TW=I} Tr(W^TXH_dX^TW). \tag{4.35}$$

où $H_d = D - \frac{D11^TD}{1^TD1}$.

Pour resoudre l'equation (4.35), on utilise un algorithme iteratif . A chaque iteration, on applique SVD sur:

$$Y = X(D^{1/2} - \frac{D11^TD^{1/2}}{1^TD1}). \tag{4.36}$$

Avec $XH_dX^T = YY^T$.

Le resumé de l'approche OMPCA est décrit ci-dessous :

Algorithm 4 : L'algorithme Optimal mean PCA (OMPCA)

1. Initialiser D en tant que matrice d'identité.
2. Modifier W en prenant juste les k vecteurs propres de Y correspondants aux k plus grandes valeurs propres.
3. Modifier b en $b = \frac{XD1}{1^TD1}$.
4. Modifier D en $d_{ii} = \frac{1}{2||d_{ii}(I-WW^T)(x_i-b)||_2}$.
5. Si W ne converge pas, aller a l'étape 2. Si non, retourner W et arrêter l'iteration.

Toutefois, ce processus s'avère long. Cela est dû principalement à l'utilisation du SVD dans l'etape 2. En effet, lorsque la dimension est plus grande que le nombre des échantillons d'apprentissage ($n >> M$) [60], le SVD nécessite environ $14nM^2 - 2M^3$ flops. Pour reduire cette compléxité on propose La méthode QR-OMPCA qui se base sur la décomposition QR. Ainsi, les vecteurs propres de YY^T seront extraits d'une manière numériquement stable. En plus, sa complexité est inférieure à celle du SVD.

4.4.2 La méthode QR-Optimal Mean PCA (QR-OMPCA)

Soit r le rang de $YY^T \in R^{n\times n}$, ou $1 \leq r \leq M$. La matrice Y peut être décomposée en une matrice orthogonale $Q \in R^{n\times r}$ et une matrice triangulaire supérieure $R_1 \in R^{r\times M}$ en utilisant la decomposiion QR comme suit :

$$Y = QR_1. \tag{4.37}$$

Alors :

$$YY^T = QR_1R_1^TQ^T. \tag{4.38}$$

La matrice R_1^T peut être décomposée en utilisant SVD de la manière suivante :

$$R_1^T = U_1D_1V^T. \tag{4.39}$$

ou $U_1 \in R^{M\times r}$ et $V \in R^{r\times r}$ sont des matrices orthogonales. $D_1 \in R^{r\times r}$ est une matrice diagonale.

Si on inclut (4.39) dans (4.38) , on obtient :

$$YY^T = QVD_2V^TQ^T. \tag{4.40}$$

Tel que $D_2 = D_1^2$. Puisque $(QV)^T(QV) = I$, on conclut que (QV) est une matrice orthogonale. Cette dernière diagonalise aussi YY^T. Ces deux faits confirment que QV est matrice des vecteurs propres de YY^T ainsi que D_2 est composée de ces valeurs propres.

Par consequent, la matrice de projection s'ecrit sous forme de $W = QV_k$ tel que V_k sont les k vecteurs propres correspondants aux k elements diagonales de D_2.

Ceci dit, la méthode QR-OMPCA sera décrite par l'algorithme 5:

Algorithm 5 : l'algorithme QR-OMPCA

1. Initialiser D en tant que la matrice identité.
2. Calculer Q et R_1 en utilisant la décomposition QR économique de Y.
3. Calculer D_1 et V à l'aide de la décomposition SVD économique de R_1^T.
4. Obtention de V_k les k vecteurs propres de YY^T.
5. $W = QV_k$.
6. Modifier b en $b = \frac{XD_1}{1^T D_1}$.
7. Modifier D en $d_{ii} = \frac{1}{2||d_{ii}(I-WW^T)(x_i-b)||_2}$.
8. Si W ne converge pas aller a l'étape 2. Si non, retourner W et arrêter l'itération.

Pour comparer la complexité de calcul de QR-OMPCA et OMPCA, il suffit de comparer la complexité au niveau de l'étape 2 de l'algorithme 4 avec la complexité des étapes 2 à 5 de l'algorithme 5. La décomposition QR économique de Y nécessite $2nM^2$ flops (si la méthode QR Gram-Schmidt modifiée est utilisée) ou $2nM^2 - 2M^3/3$ flops (si la méthode QR de Givens est utilisée) [60]. Le SVD économique appliqué sur $R1^T$ dans l'équation (4.39) nécessite $4Mr^2 + 8r^3$ flops [60]. Enfin, le produit de Q et V_k nécessite $2nrk$ flops. L'estimation totale de QR-OMPCA est $2nM^2 + 2nrk$ flops, ce qui est inférieur à celui du SVD-OMPCA.

4.4.3 Experiences

Dans cette section, nous effectuons de nombreuses simulations pour évaluer la performance de l'approche proposée en utilisant le classificateur K-NN. Dans la figure 4.12, nous comparons PCA, R1-PCA[46] et QR-OMPCA en termes de DR et de FPR. Puisque ces méthodes peuvent partager la dimensionnalité commune réduite, elles peuvent être comparées sous la même dimension. Dans la simulation, nous avons choisi de travailler sur 4 dimensions $(n' = 4)$. Ensuite, nous augmentons la taille des données d'apprentissage et illustrons son effet sur les comportements DR et FPR. Nous changeons la taille des échantillons d'entraînement de la manière suivante : Nous avons commencé avec 1350 échantillons d'apprentissage composés de 1000 échantillons normaux, 100 attaques DOS, 50 U2R, 100 R2L et 100 attaques PROBE. Après cela, nous tra-

vaillons sur 2000 échantillons normaux, 200 DOS, 50 U2R, 100 R2L et 200 PROBE,...etc. U2R et R2L restent inchangés à cause de la rareté de ce type d'attaques. Nous selectionons un ensemble de données de test avec 100 échantillons normaux, 100 DOS, 50 U2R, 100 R2L et 100 attaques PROBE. Il convient de noter que les échantillons ont été sélectionnés au hasard parmi les ensembles de données.

La figure 4.12.a montre que QR-OMPCA surpasse R1-PCA et PCA dans la plupart du temps en terme de DR. La figure 4.12.b affirme que l'approche proposée et R1-PCA produisent le FPR le plus bas. Ce sont les résultats qui concernent KDDcup99.

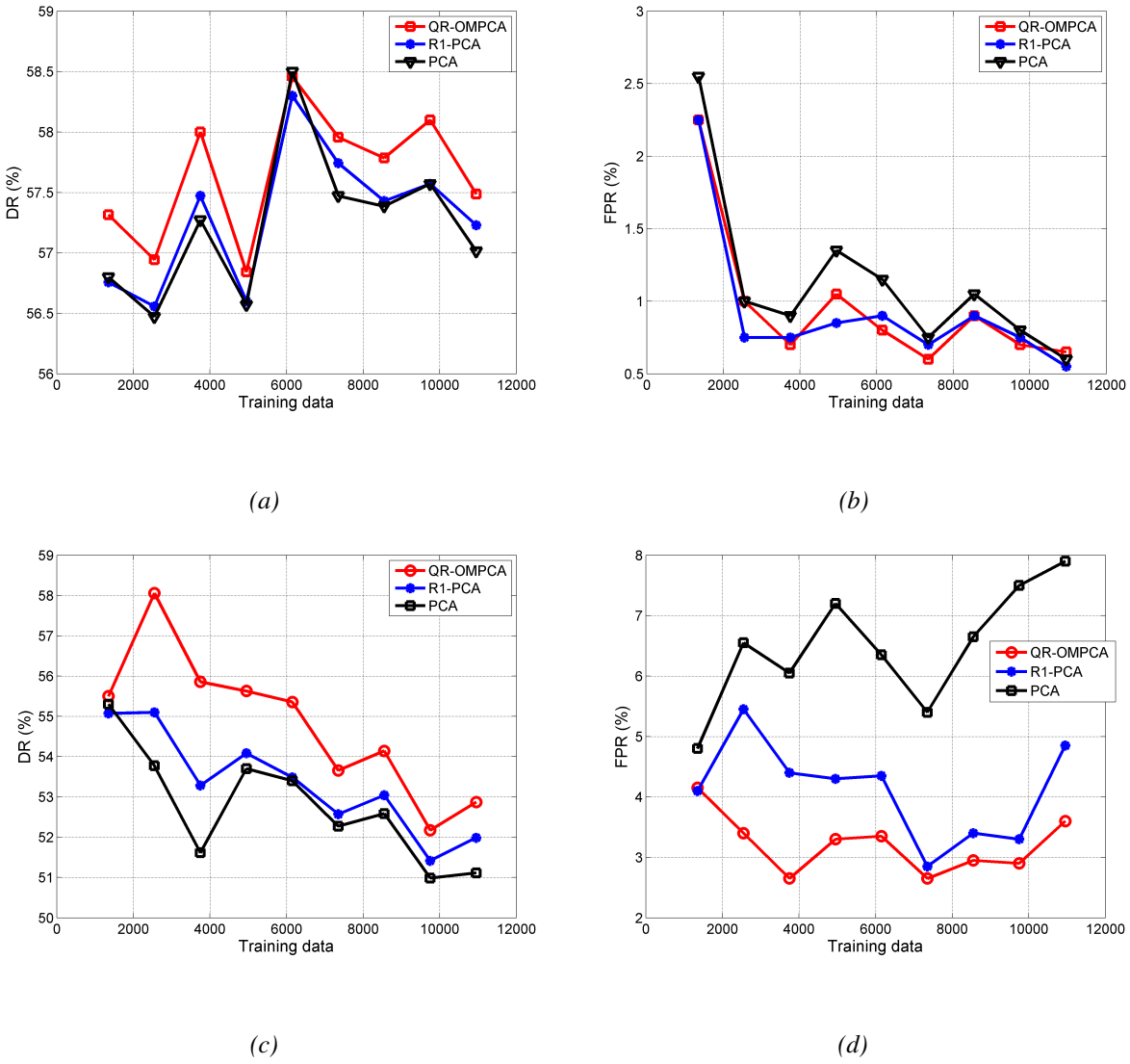


Figure 4.12: *(a) : DR (%) vs. Nombre de données d'apprentissage pour KDDcup99*
(b) : FPR (%) vs. Nombre de données d'apprentissage pour KDDcup99
(c) : DR (%) vs. Nombre de données d'apprentissage pour NSL-KDD
(d) : FPR (%) vs. Nombre de données d'apprentissage pour NSL-KDD

Lorsque nous reproduisons la même expérience sur NSL-KDD (figure 4.12.c et figure 4.12.d),

l'approche proposée surpasse R1-PCA et PCA d'une manière permanente. Néanmoins, nous observons que l'augmentation des données d'apprentissage conduit à une diminution de DR. L'explication de ce fait réside dans la nature de ces données. Le trafic d'apprentissage contient un nombre important de connexions normales. Par conséquent, créer un modèle qui détecte les attaques devient vraiment difficile une fois le nombre d'échantillons d'apprentissage atteint une valeur élevée. C'est ce qui cause la détérioration de DR. La figure 4.12.d confirme que QR-OMPCA donne un minimum de FPR. Ce résultat trivial est dû à la résistance importante de QR-OMPCA face aux données aberrantes.

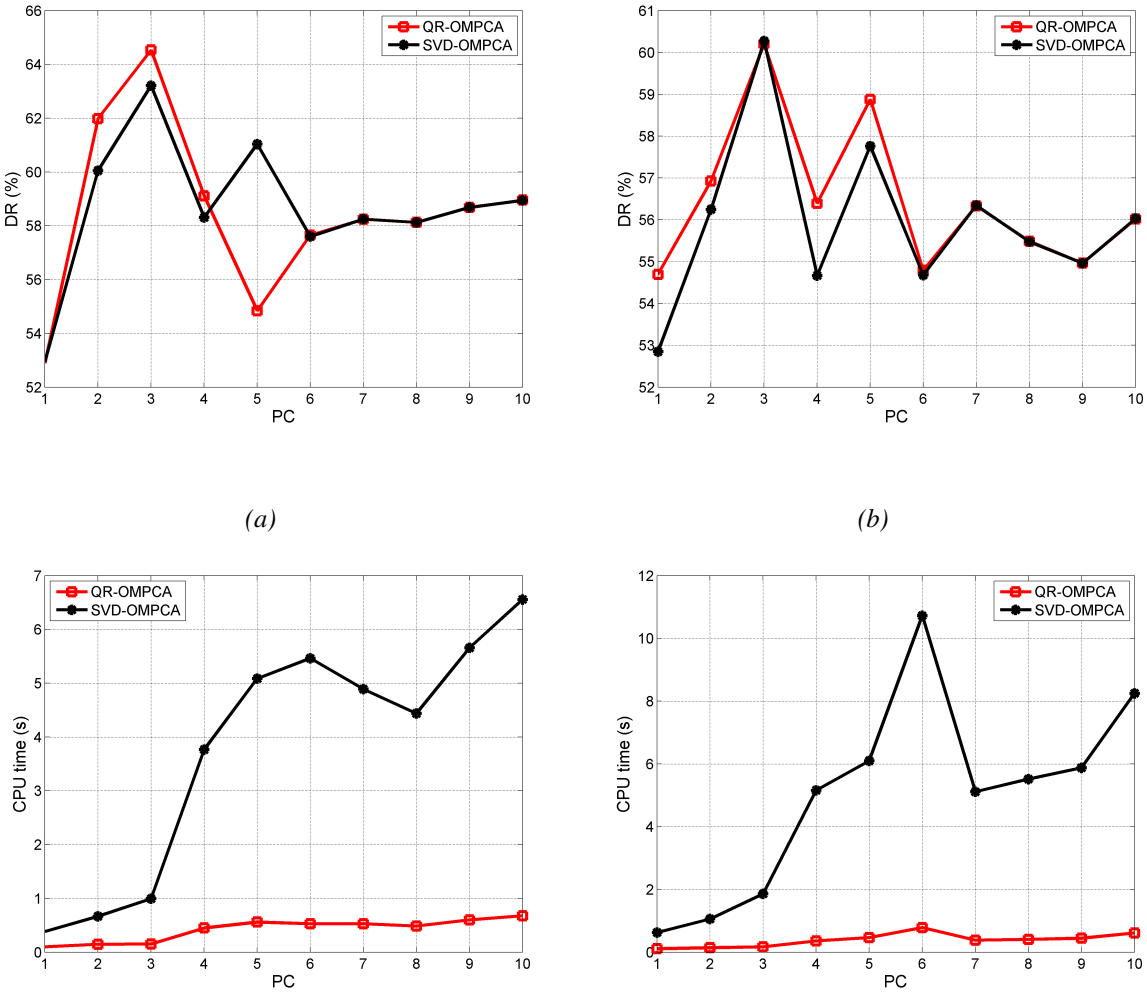


Figure 4.13: *(a): DR(%) vs. Nombre de composantes principales concernant KDDcup99*
(b): DR(%) vs. Nombre de composantes principales concernant NSL-KDD
(c): Temps CPU (s) vs. Nombre de composantes principales concernant KDDcup99
(d): Temps CPU (s) vs. Nombre de composantes principales concernant NSL-KDD

Dans l'expérience suivante illustrée par la figure 4.13, nous comparons QR-OMPCA avec

SVD-OMPCA en termes de taux de détection et de temps CPU consommé. Pour cela, nous générons des données d'apprentissage composées de 1000 échantillons normaux, 100 attaques DOS, 50 attaques U2R, 100 R2L et 100 attaques PROBE. Ensuite, nous varions le nombre de composants principaux *PC* de 1 à 10 et nous visualisons le temps CPU consommé. La figure montre le DR des deux méthodes lorsqu'il est appliqué sur les deux bases de données. On peut voir à partir de ces figures que la méthode proposée est plus précise que la SVD-OMPCA lorsque *PC* est inférieur à 4. Une fois ce dernier dépassé, les résultats deviennent similaires. Les figures 4.13.c et 4.13.d représentent la relation entre le nombre de composants principaux et le temps CPU consommé. D'une part, nous observons que l'augmentation du nombre de *PC* conduit à une consommation de temps plus élevé. D'un autre côté, les resultats montrent que la méthode proposée est plus rapide que SVD-OMPCA. La rapidité de QR-OMPCA provient principalement de l'exploitation de la décomposition QR. Cette dernière décompose la matrice Y en matrice orthogonale et en matrice triangulaire supérieure. Après cela, la décomposition SVD peut être appliquée uniquement à la matrice triangulaire supérieure. Ceci consomme moins de temps par rapport a celui consommé en utilisant SVD directement sur la matrice rectangulaire Y.

Pour résumer, dans cette section, nous avons proposé une variante de PCA, appelée QR-OMPCA. Cette méthode incorpore un rapide calcul de la moyenne optimale dû à l'utilisation de la décomposition QR. L'intégration de QR-OMPCA dans le système de détection d'intrusion (IDS) rend ce dernier plus efficace et résistant face aux données aberrantes. En plus, il accélère le processus de l'IDS. De tels faits ont été vérifiés par de nombreuses expériences sur KDDcup99 et NSL-KDD. En même temps, les résultats montrent que QR-OMPCA surpasse R1-PCA, PCA et OMPCA.

4.5 Conclusion

Dans ce chapitre nous avons proposé trois variantes de PCA, afin de remédier aux problèmes de la méthode d'une part et d'optimiser la précision de l'IDS proposé d'autre part. Dans un premier temps, nous avons pu résoudre le problème de la non linéarité de PCA en proposant l'utilisation de KPCA avec des nouvelles fonctions a noyau. A savoir, le noyau à puissance et le noyau sphérique. L'idée de base de KPCA est fondée sur l'utilisation de l'astuce du noyau "kernel trick" pour transformer les données d'entrée dans un espace de caractéristiques implicite. Puis, ces données sont traitées dans cet espace pour produire des caractéristiques non linéaires qui facilitent la classification des connexions.

Le problème des données aberrantes existantes dans un trafic réseau a été contourné en utilisant un nouveau PCA-Lp Basée sur le gradient conjugué au lieu du gradient simple utilisé dans la littérature. Cette technique tente de trouver des projections qui maximisent la covariance totale en utilisant la norme Lp $(p < 2)$ au lieu de la norme L2. Parmi les avantages les plus notables du gradient conjugué on cite la faible exigence de la mémoire et la vitesse de convergence.

La troisième contribution appelée QR-OMPCA partage la même philosophie que celle de PCA-Lp. Elle cherche un modèle plus robuste face aux données aberrantes. Elle essaye de corriger

une mauvaise formulation de PCA-Lp(p=1) et R1-PCA en incluant un processus de calcul de la moyenne optimale. Cette approche est caractérisée par l'utilisation de la décomposition QR au lieu du SVD employée dans un travail antérieur. Par conséquent, l'intégration de QR-OMPCA dans le système de détection d'intrusion (IDS) rend ce dernier plus efficace et plus rapide.

Enfin, et pour valider nos algorithmes, nous avons procédé à des tests sur les bases de données KDDcup99 et NSL-KDD. Nous avons comparé nos approches avec PCA, R1-PCA et PCA-Lp (p=1) et nous avons obtenu des résultats satisfaisants en terme de taux de détection et de faux positifs.

CHAPITRE

5 Amélioration des Algorithmes d'Analyse Linéaire Discriminante (LDA)

"If you believe that discrimination exists, it will."
ANTHONY J. D'ANGELO

5.1 Introduction

L'analyse discriminante linaire –Linear Discriminant Analysis (LDA)– est un algorithme d'extraction de caractéristiques supervisé qui permet d'obtenir une séparation linéaire entre un ensemble de classes [58]. Afin d'achever cet objectif, LDA minimise la dispersion intra-classe tout en maximisant la dispersion inter-classe.

En d'autres termes, LDA trouve un ensemble de vecteurs de projection W qui maximise une matrice de dispersion inter-classe appelée S_b et minimise une matrice de dispersion intra-classe appelée S_w.

Cependant la méthode LDA classique souffre de plusieurs problèmes. Le premier est connu sous le nom de "Small Sample Size Problem" (SSS), il se produit lorsque la dimension des données d'apprentissage est très grande par rapport au nombre des données.

Par consequent, la matrice S_w peut devenir singulière et cela rend difficile le calcul des vecteurs de projections.

Plusieurs approches ont été proposées pour résoudre ce problème. Parmi les solutions on peut citer l'utilisation de la méthode LDA régularisée (R-LDA regularized LDA) [41] qui additionne une matrice diagonale $\alpha \times I$ à la matrice S_w pour eviter la singularité de celle-ci. Fisherface (PCA+LDA) [14] considère une étape préliminaire de réduction de dimensions avant d'appliquer LDA. La méthode de l'espace nul de S_w [32] exploite l'espace nul de S_w pour obtenir la transformation W. Un autre algorithme appelé Direct LDA (D-LDA) [138, 177] essaie de résoudre ce problème en inversant les étapes de diagonalisation simultanées. Un algorithme basé sur la décomposition QR qui elimine la singularité des matrices de dispersion a été introduit par Ye *et*

al. [176]. Hansun Park [71] a utilisé la décomposition SVD généralisée pour résoudre le SSS. D'autres travaux exploitent le critère de marge maximale pondérée (WMMC) [94, 97]. Cette approche considère la différence pondérée entre la dispersion inter-classe et la dispersion intra-classe. Par la suite, une version robuste de WMMC utilisant la norme L1 invariante rotationnelle (R1-LDA) a été développée [96].

Un deuxième problème découle de la formulation de LDA. Cette dernière utilise d'une part la norme L2, d'autre part elle exploite la moyenne classique pour calculer les matrices de dispersion. Or l'utilisation de cette norme amplifie l'effet des valeurs aberrantes dans les données et mène a une projection erronée. Pour corriger cette carrence, des travaux qui considèrent les modèles sparsés de LDA "sparse LDA methods" [171]. D'autres articles ont proposé la régularisation supervisée basée sur un sous-espace robuste (SRRS), qui reconstruit les données originales en employant une représentation (LRR) [106] et applique simultanément LDA. Par ailleurs, l'effet de la norme L2 persiste encore. Pour contourner le problème, de nombreux chercheurs ont proposé des méthodes de LDA plus robustes intégrants la norme L1 [163, 182, 183].

La moyenne classique est sensible aussi au données aberrantes et peut falsifier le calcul [175]. Pour résoudre ce problème, des publications récentes ont proposé des moyennes plus robustes aux données aberrantes tel que [95, 181]. La première proposition se base sur le critère de marge maximale (MMC) et intègre le vecteur moyen maximum-minimum-médian au lieu de la moyenne classique pour construire la matrice de dispersion intra-classe S_w et la matrice de dispersion entre classes S_b. Les résultats expérimentaux sur les bases de données ORL et Yale montrent qu'une amélioration du modèle (MMC) est possible avec la technique proposée. La deuxième approche minimise l'inverse de la moyenne harmonique pondérée de la distance entre les classes. Ce qui s'avère plus robuste que la minimisation de l'inverse de la moyenne arithmétique classique. Des expériences approfondies sur différents types de données montrent que l'approche surpasse plusieurs autres en termes précision de la classification.

Un troisième problème concerne le type de structure de données manipulée par LDA. Cette approche accorde plus d'attention à la structure globale des classes. En conséquence, les caractéristiques discriminantes produites sont souvent imprécises. Pour remédier à cette situation, des travaux antérieurs [31, 151, 164] ont exploité la structure locale pour obtenir des caractéristiques optimales. Néanmoins, dans ces travaux, il est nécessaire de faire une décomposition d'une énorme matrice. Pour la détection d'intrusion ca sera une tâche longue et même infaisable.

Afin de contourner le deuxième et troisième problème, les contributions proposées dans cette thèse ont pour but l'introduction des nouvelles variantes : Median NN-LDA [51], geomean LDA et R1-PCA+median LDA [185]. Ces méthodes ont pour but l'optimisation de LDA en proposant différentes approches. La première méthode résout le problème d'inconsistance des données en se basant sur les éléments les plus proches du median de chaque classe. Par conséquent, cette méthode offre une meilleure séparabilité entre classes dans l'espace réduit. Ce qui facilite la détection des intrusions par la suite. Geomean LDA propose de remplacer la moyenne classique utilisée dans LDA par la moyenne géométrique. Cette dernière est plus robuste vis-à-vis des données aberrantes. La dernière méthode combine deux puissants algorithmes R1-PCA et median

LDA. En suivant cette approche, on tire profit des avantages de la norme R1 et du median. Ce qui offre une résistance remarquable face au données aberrantes et évite la singularité des matrices.

Dans la premiere section de ce chapitre, on présentera les formulations mathématiques de LDA et de quelques variantes proposées dans l'etat de l'art, ensuite, on décrit des nouvelles approches et on illustre leur effet observé dans la détection des intrusions.

5.2 L'analyse discriminante lineaire (LDA)

Considérons un ensemble de connexions partitionnées en k classes, chaque classe correspond à un type, soit normal ou malicieux. L'analyse discriminante linéaire (LDA) est une technique supervisée, qui a pour but de construire à partir de ces données, un sous-espace linéaire de l'espace initial des données, dans lequel les k classes sont les mieux séparées possible. Pour déterminer le type d'une connexion, il suffit de la projeter dans ce sous-espace et de déterminer à quelle classe elle est la plus proche.

D'un point de vue mathématique, l'analyse discriminante linéaire (LDA) est une technique supervisée, basée sur la maximisation d'un critère de séparabilité. Soit $X = [x_1, \ldots, x_M] \in \mathbb{R}^{n \times M}$ composée de M vecteurs, notre but est de trouver une transformation linéaire $W \in \mathbb{R}^{n \times l}$ qui transforme chaque vecteur x_i en un nouveau vecteur x_i^l dans l'espace réduit de dimension l comme suit :

$$x_i^l = W^T x_i \in \mathbb{R}^l (l < n)$$

La matrice de données X peut être réécrite comme $X = [X_1, \ldots, X_k]$ tel que k est le nombre de classes et $X_i \in \mathbb{R}^{n \times M_i}$ represente les données de la $i$ème classe, M_i est le nombre de connexion de la $i$ème classe et $\sum_{i=1}^{k} M_i = M$. LDA se base sur trois matrices : la matrice de dispersion intra-classe, la matrice de dispersion inter-classe, la matrice de dispersion totale. Ces matrices sont définies de la manière suivante :

$$S_w = \frac{1}{M} \sum_{i=1}^{k} \sum_{x \in X_i} (x - c_i)(x - c_i)^T \tag{5.1}$$

$$S_b = \frac{1}{M} \sum_{i=1}^{k} M_i (c_i - c)(c_i - c)^T \tag{5.2}$$

$$S_t = \frac{1}{M} \sum_{i=1}^{M} (x_i - c)(x_i - c)^T \tag{5.3}$$

c_i est la moyenne de la $i$ème classe, et c est la moyenne générale. Il est prouvé que $S_t = S_w + S_b$ [58]. A partir de (5.1) et (5.2) on conclut que:

$$trace(S_w) = \frac{1}{M} \sum_{i=1}^{k} \sum_{x \in X_i} ||x - c_i||^2 \tag{5.4}$$

$$trace(S_b) = \frac{1}{M}\sum_{i=1}^{k} M_i ||c_i - c||^2 \tag{5.5}$$

La trace de S_w nous donne une idée de la façon dont chaque échantillon est proche de la moyenne de sa classe. La trace de S_b estime le degré d'eloignement de chaque classe par rapport a la moyenne globale. Dans l'espace réduit transformé par W, les trois matrices de dispersion deviennent :

$$\begin{aligned} S_w^l &= W^T S_w W \\ S_b^l &= W^T S_b W \\ S_t^l &= W^T S_t W \end{aligned}$$

La matrice de projection optimale est obtenue en maximisant la fonction objectif :

$$W = \arg\max \frac{trace(S_b)}{trace(S_w)} = \arg\max \frac{|S_b^l|}{|S_w^l|} \tag{5.6}$$

Quand S_w est inversible, les solutions de (5.6) sont obtenues en performant la décomposition a valeurs propre suivante :

$$S_w^{-1} S_b w_i = \lambda_i w_i \tag{5.7}$$

Ou $W = [w_1, \ldots, w_l]$.

Néanmoins, dans le contexte de la détection des intrusions, il y a quelques données qui sont sous-représentées, au sens où la taille n des connexions d'apprentissage est supérieure à leur nombre ($n >> M$). La matrice S_w devient singulière dans ce cas et il sera difficile de calculer S_w^{-1}. C'est le problème de la singularité qui est plus connu sous le "Small Sample Size (SSS) problem".

Pour éviter la singularité de S_w, les sous sections suivantes décrivent les solutions les plus utilisées par les chercheurs dans l'état de l'art.

5.2.1 PCA+LDA

Afin d'éviter le problème de la singularité de la matrice de dispersion intra-classe S_w Belhumeur *et al.* [14] a introduit PCA+LDA. Elle consiste à appliquer une PCA en amont à LDA. En faisant ceci, on arrive a réduire la dimensionnalité du problème, de manière à ce que la nouvelle matrice de dispersion intra-classe soit inversible, et ceci en conservant au maximum la forme de la distribution initiale des données.

Pour cette méthode la matrice de projection optimale est calculée par [14] :

$$W = W_{pca} * W_{lda} \tag{5.8}$$

où W_{pca} est la matrice de projection de l'espace original vers le sous-espace PCA de taille l_1. Elle est définie par l'équation suivante :

$$W_{lda} = \arg\max_W = |W^T S_t W| \tag{5.9}$$

De plus, W_{lda} est la matrice de projection du sous-espace PCA vers le sous-espace LDA de taille l_2 obtenu en maximisant le rapport suivant :

$$W_{lda} = \arg\max_{W}\left(\frac{|W^T W_{pca}^T S_b W_{pca} W|}{|W^T W_{pca}^T S_w W_{pca} W|}\right) \tag{5.10}$$

La solution optimale de cette équation est constituée par les vecteurs propres de la matrice suivante : $(W_{pca}^T S_w W_{pca})^{-1}(W_{pca}^T S_b W_{pca})$.

5.2.2 Null space LDA

Cette méthode a été proposée pour la première fois par [32]. Les auteurs ont montré que l'espace nul de S_w contient une information discriminante si la projection de S_b est non nulle dans les directions ainsi définies. En termes mathématiques, au lieu de travailler avec (5.6), les auteurs maximisent la fonction objectif :

$$W = \arg\max_{W^T S_w W = 0} |W^T S_b W| \tag{5.11}$$

Le calcul de W requiert l'espace nul de S_w. Pour obtenir ce dernier, vu que la matrice S_w est très grande, il nous faut beaucoup de temps. Pour pallier cette carence, les travaux présentés par [73] proposent d'éliminer l'espace nul de S_t comme première étape, avant d'attaquer le problème (5.11). La procédure se base sur les étapes suivantes:

1. Supprimer l'espace nul de S_t. Ceci sera fait en formant une matrice U qui contient les vecteurs propres de S_t. Apres, l'espace nulle de S_t sera éliminé en projetant les données sur le sous-espace engendré par les vecteurs colonnes de U. Soient S'_w et S'_b les nouvelles matrices de dispersion obtenues après élimination de l'espace nul de S_t.

$$S'_w = U^T S_w U \tag{5.12}$$

$$S'_b = U^T S_b U \tag{5.13}$$

2. Calcul de l'espace nul G de S'_w grâce à la décomposition en valeurs propres de S'_w.

$$G^T S'_w G = (UG)^T S_w (UG) = 0 \tag{5.14}$$

   A partir de cette équation, nous concluons que UG est l'espace nul de S_w

3. Projeter S'_b sur l'espace nul de S_w et trouver S''_b .

$$S''_b = (UG)^T S'_b (UG). \tag{5.15}$$

4. Supprimer l'espace nul de S''_b avec une analyse en valeurs propre de S''_b. Pour achever cela, nous considérons V une matrice dont les colonnes sont les vecteurs propres de S''_b associés aux plus grandes valeurs propres.

$$(V)^T S''_b(V) = V^T (UG)^T S'_b(UGV) \tag{5.16}$$

$$= (UGV)^T S_b(UGV). \tag{5.17}$$

5. À partir de l'équation finale, nous concluons que la matrice de projection W s'ecrit sous forme de:

$$W = UGV. \tag{5.18}$$

5.2.3 Direct LDA

L'algorithme Direct LDA [177] permet le calcul de la matrice de projection W_{dlda} de la manière suivante:

Comme première étape on diagonalise S_b ensuite on élimine l'espace nul de S_b: $Y^T S_b Y = A > 0$. Avec Y la matrice des vecteurs propres de S_b. A est la matrice diagonale de taille $l \times l$ des valeurs propres non nulles.

Dans la deuxième étape on transforme S_w en $Q_w = A^{-1/2} Y^T S_w A^{-1/2}$ puis on diagonalise Q_w tel que $U^T Q_w U = B$.

A la fin on obtient la matrice de projection W_{dlda}, qui diagonalise simultanément le numérateur et le dénominateur du critère de Fisher. Elle est définie par : $W_{dlda} = YB^{-1/2}UB^{-1/2}$.

5.2.4 Pseudo LDA

Plusieurs travaux ont contribué a l'apparition d'approches proposant l'utilisation des techniques de l'algèbre linéaire qui se basent sur les décompositions matricielles pour venir a bout du problème de la singularité des matrices. Parmi les méthodes les plus employées, la méthode dite pseudo LDA.

Au lieu de travailler avec l'inverse S_w^{-1} de la matrice de dispersion intra-classe de l'algorithme LDA standard, Pseudo LDA utilise la pseudo-inverse S_w^+. Dans ce cas, la matrice W de projection est constituée par les vecteurs propres de $S_w^+ S_b$ associés aux plus grandes valeurs propres. Le calcul de S_w^+ passe par la décomposition en valeurs singulières (SVD) de la matrice S_w, définie comme suit :

$$S_w = UDV^T \tag{5.19}$$

Où, U et V sont deux matrices orthonormées de taille $n \times n$. D= diag$(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$ est la matrice diagonale contenant les valeurs singulières avec r= rang$(S_w) < n$. Si l'on note D^+= diag$(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_r, 0, \ldots, 0)$, alors la pseudo-inverse S_w^+ [60] s'écrit :

$$S_w^+ = VD^+U^T \tag{5.20}$$

5.3 Median NN-LDA

Des classes inconsistantes sont des classes dont on ne peut pas définir leurs vraies structures. Elles peuvent être non gaussiennes ou non linéairement séparables. Ce type de classes pose un problème pour LDA. Pour surmonter cet inconvénient, on a tendance a diviser la classe en deux distributions. Une distribution centrale ou locale qui définit d'une certaine manière la nature de la classe. Une distribution globale qui détermine les limites de la classe. Ces 2 distributions doivent être préservé lors de l'obtention des vecteurs de projection. Dans la littérature, (LFDA) a été proposé [150] ensuite sa version semi-supervisé [151] a eté introduite en 2010. Ces méthodes considèrent des matrices inter-classe et intra-classe locales. Pour les calculer, la première approche a recours a des matrices de pondérations $W^{(lb)}$ et $W^{(lw)}$. La version semi-supervisé calcule la matrice de pondération régularisée $W^{(rlb)}$. Ces matrices se basent sur un facteur β qui détermine les éléments appartenant à la même classe et celles qui n'appartiennent pas. Les auteurs affirment que la structure de données locale dans le la même classe tend à être préservée quand β est petit, mais il n'est plus préservé quand β est grand. Par ailleurs, pour calculer les matrices de pondérations il faut beaucoup de temps. Le travail [31] s'est basé sur la préservation de la relation de voisinage lors de la projection des classes dans un espace de dimension réduite. Les points voisins représentent la distribution locale alors que les autres constituent la distribution globale. Les auteurs commencent par construire des graphes de voisinage G et G'. Ensuite ils calculent les poids d'affinité pour définir le degré de connectivité de chaque élément de classe. Finalement, ils exécutent une décomposition en valeurs propres pour trouver la matrice de projection. Cette méthode s'est avérée efficace sur divers bases de données. Par ailleurs la construction des graphes n'est pas une tache évidente dans le cas des intrusions. LPMIP [164] cherche un compromis entre les structures globales et locales, qui est ajusté par un paramètre α. Malheureusement, il suit la même philosophie que celle utilisé dans les travaux antérieurs. Il est nécessaire de faire un grand calcul des matrices de pondération.

Afin de résoudre ce problème nous proposons une solution qui exploite la médiane de chaque classe. Cette section détaille l'approche.

5.3.1 Idée de base

La figure 5.1 illustre deux classes non consistantes et non gaussiennes. Pour surmonter le problème d'inconsistance et trouver des frontières locales entre ces deux classes, nous proposons d'exploiter la distribution locale de chaque classe. Pour cela, nous nous basons sur le concept du médiane. Dans la théorie des probabilités et les statistiques, la médiane est définie comme un vecteur qui sépare la moitié supérieure d'une distribution de probabilité de la moitié inférieure. C'est la valeur moyenne dans une distribution, au-dessus et au-dessous de laquelle se trouvent un nombre égal d'échantillons. À partir de cette hypothèse, nous observons que les échantillons qui sont proches de la médiane représentent la distribution centrale de chaque classe et correspondent logiquement à la distribution locale. D'un autre côté, nous pouvons assimiler les autres échantillons à la distribution globale, puisqu'ils existent naturellement sur les frontières de la classe et

facilitent la séparation des classes. Avec ce concept, nous dissocions les deux distributions. Par conséquent, nous résolvons la question de la cohérence de la distribution.

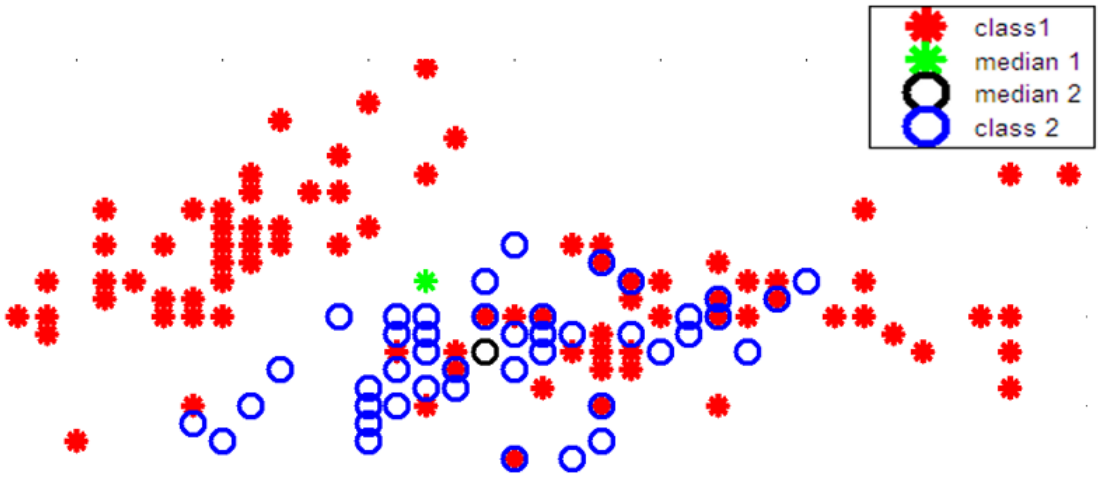


Figure 5.1: *Deux classes inconsistantes et non gaussiennes.*

Notre approche fonctionne également bien même si les données ne sont pas gaussiennes ou ont des frontières non linéaires. Puisqu'elle peut extraire les structures globales des données en déterminant les échantillons qui sont éloignés de la médiane. Le procédé peut obtenir un certain nombre de vecteurs discriminants linéaires locaux qui approchent la frontière non linéaire entre les classes. Le but de l'approche est de chercher une représentation similaire à la figure 5.2.

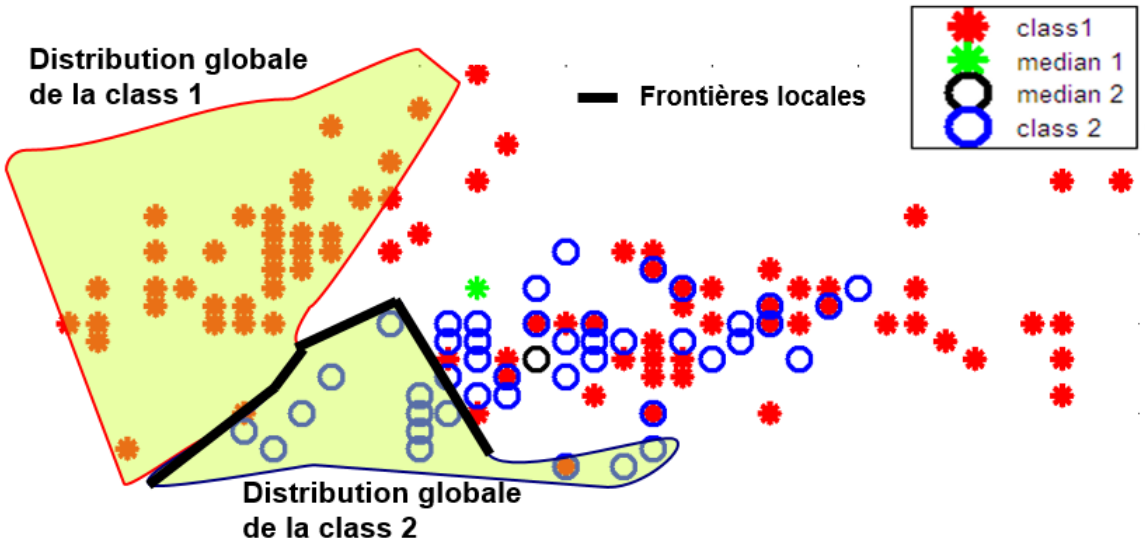


Figure 5.2: *Deux classes séparables à l'aide des frontières locales.*

5.3.2 Formulation mathématique

En termes mathématiques, X_i sera divisé en X_i^w et X_i^b.

Soit $X_i^w = [x_1, \ldots, x_p] \in \mathbb{R}^{n \times p}$ la matrice représentant les p plus proches voinsins du mediane de chaque classe.

Soit $X_i^b = [x_{p+1}, \ldots, x_{M_i}] \in \mathbb{R}^{n \times (M_i - p)}$ la matrice représentant les $M_i - p$ données les plus loins du mediane de chaque classe.

La distribution locale X_i^w sera exploitée par la nouvelle matrice de dispersion intra-classe S'_w, car celle ci mesure d'une certaine manière la compacité intra-classe. D'autre part, la distribution globale représentée par X_i^b est nécessaire pour calculer la nouvelle matrice de dispersion inter-classes S'_b et plus précisément la moyenne générale c.

Après les équations (5.1) et (5.2) seront réécrites de la manière suivante:

$$S'_w = \frac{1}{p} \sum_{i=1}^{k} \sum_{x \in X_i^w} (x - c_i^w)(x - c_i^w)^T \tag{5.21}$$

$$S'_b = \frac{1}{p} \sum_{i=1}^{k} (c_i^w - c)(c_i^w - c)^T \tag{5.22}$$

Où c_i^w est la moyenne de X_i^w, c_i^b est la moyenne de X_i^b et $c = \frac{1}{k} \sum_{i=1}^{k} (c_i^b)$ représente la moyenne générale.

Par conséquent, les équations (5.4) et (5.5) seront remplacées par:

$$trace(S'_w) = \frac{1}{p} \sum_{i=1}^{k} \sum_{x \in X_i^w} ||x - c_i^w||^2 \tag{5.23}$$

$$trace(S'_b) = \frac{1}{p} \sum_{i=1}^{k} n_i ||c_i^w - c||^2 \tag{5.24}$$

Nous obtenons les vecteurs discriminants en maximisant la fonction objective suivante:

$$W' = \arg\max \frac{trace(S'_b)}{trace(S'_w)} \tag{5.25}$$

La solution peut être atteinte en effectuant:

$$(S'_w)^{-1} S'_b w'_i = \lambda'_i w'_i \tag{5.26}$$

Où $W' = [w'_1, \ldots, w'_l]$.

Afin de traiter le problème de singularité, nous proposons d'appliquer (PCA) [76] comme étape de réduction de dimension intermédiaire.

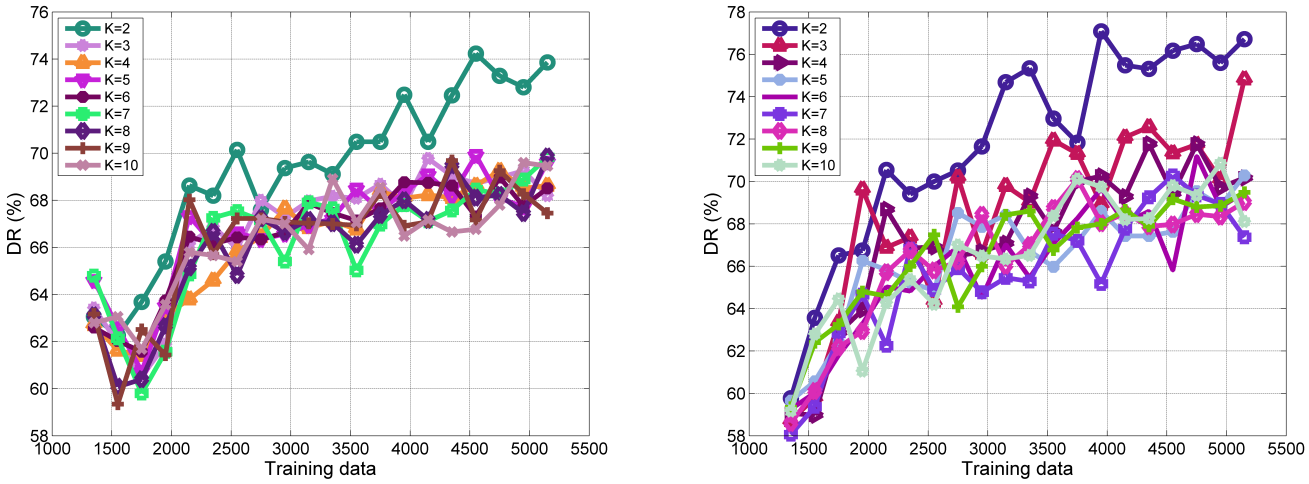
5.3.3 Tests Expérimentaux

Afin de démontrer l'efficacité de la méthode proposée, nous menons une série d'expériences et nous comparons également les performances du median NN-LDA avec LDA, Direct LDA , null space LDA, pseudo LDA.

Dans nos expériences, nous avons varié la taille des échantillons d'apprentissage et gardé les données de test intact avec la composition suivante (100 données normales, 100 données DOS, 50

données U2R, 100 données R2L et 100 PROBE). Pour réduire la variation du taux de détection (DR), nous considérons la moyenne de vingt essais.

La première expérience consiste à définir le nombre adéquat d'échantillons p qui représentent la structure locale de chaque classe. En théorie, c'est difficile à faire. Le p le plus approprié est affecté par plusieurs facteurs tels que le nombre total d'échantillons, le nombre de classes, la distribution des échantillons. Par conséquent, la valeur de p est déterminée empiriquement. Dans notre cas, nous considérons p comme $\frac{M_i}{K}$ et nous varions K de 2 à 10. La figure (5.3).a et La figure (5.3).b montrent que $p = \frac{M_i}{2}$ est la valeur qui produit le taux de détection (DR) le plus élevé pour KDDcup99 et NSL-KDD. Par conséquent, nous fixons p à cette valeur dans les prochaines expériences.



(a) *(b)*

Figure 5.3: *(a) : DR(%) sous differents K pour KDDcup99*
(b) : DR(%) sous differents K pour NSL-KDD

Dans la seconde expérience, nous comparons notre méthode aux algorithmes suivants : LDA, median LDA , nul space LDA, Direct LDA et pseudo LDA. Pour éviter le (SSS) problème, PCA est utilisée comme première étape dans les algorithmes LDA, median LDA et median NN-LDA. Par conséquent, ces algorithmes peuvent également être vus comme PCA + LDA, PCA + median LDA, PCA + median NN-LDA. Nous avons choisi de travailler avec 3 composantes principales *PC* dans la phase de PCA. Dans la deuxième phase nous avons choisi les 3 premières w_i. Le reste des algorithmes LDA exploitent les 4 premières w_i. Ceci dit, nous avons augmenté le nombre de données d'apprentissage et nous avons visualisé leur effet sur DR et FPR.

La figure (5.4) illustre les résultats trouvés. Selon (5.4).a et (5.4).c, nous observons que notre approche prend le dessus au fur et à mesure que les données d'apprentissage augmentent. La raison derrière ce phénomène semble être que plus il existe des données d'apprentissage, plus la structure locale autour de chaque médiane de classe peut être capturée. En outre, lorsque nous

augmentons le nombre d'échantillons d'apprentissage, les frontières de chaque classe deviennent plus structurées et séparables. Ce fait permet autant que possible à préserver la distribution globale. Les figures (5.4).b et (5.4).d illustrent l'effet des échantillons d'apprentissage sur le FPR. Il est clair que median NN-LDA produit le plus faible taux de faux positifs par rapport aux autres méthodes. Ceci prouve la grande capacité de notre approche à identifier les connexions normales indépendamment de la taille des données d'apprentissage.

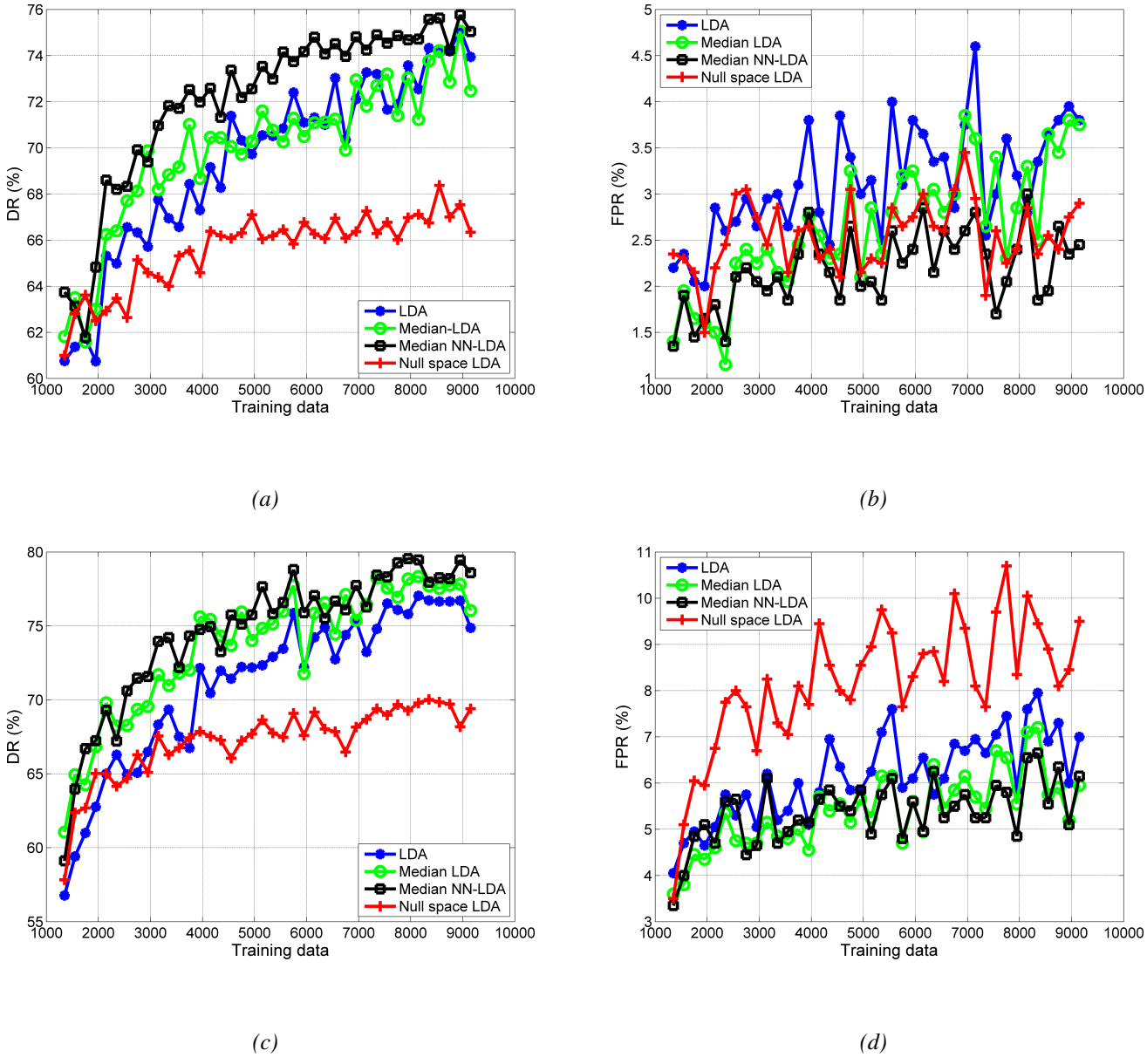


Figure 5.4: *(a) : DR(%) vs. Echantillons d'apprentissage pour KDDcup99*
(b) : FPR(%) vs. Echantillons d'apprentissage pour KDDcup99
(c) : Detection rate (%) vs. Echantillons d'apprentissage pour NSL-KDD
(d) : FPR(%) vs. Echantillons d'apprentissage pour NSL-KDD

Pour évaluer davantage les performances de notre approche, nous la comparons à d'autres

méthodes telles que Direct LDA et pseudo LDA. La figure (5.5) expose les résultats obtenus en utilisant KDDcup99 et NSL-KDD. Nous avons varié la taille des échantillons d'apprentissage de 1350 à 9150 ensuite on illustre les comportements DR et FPR.

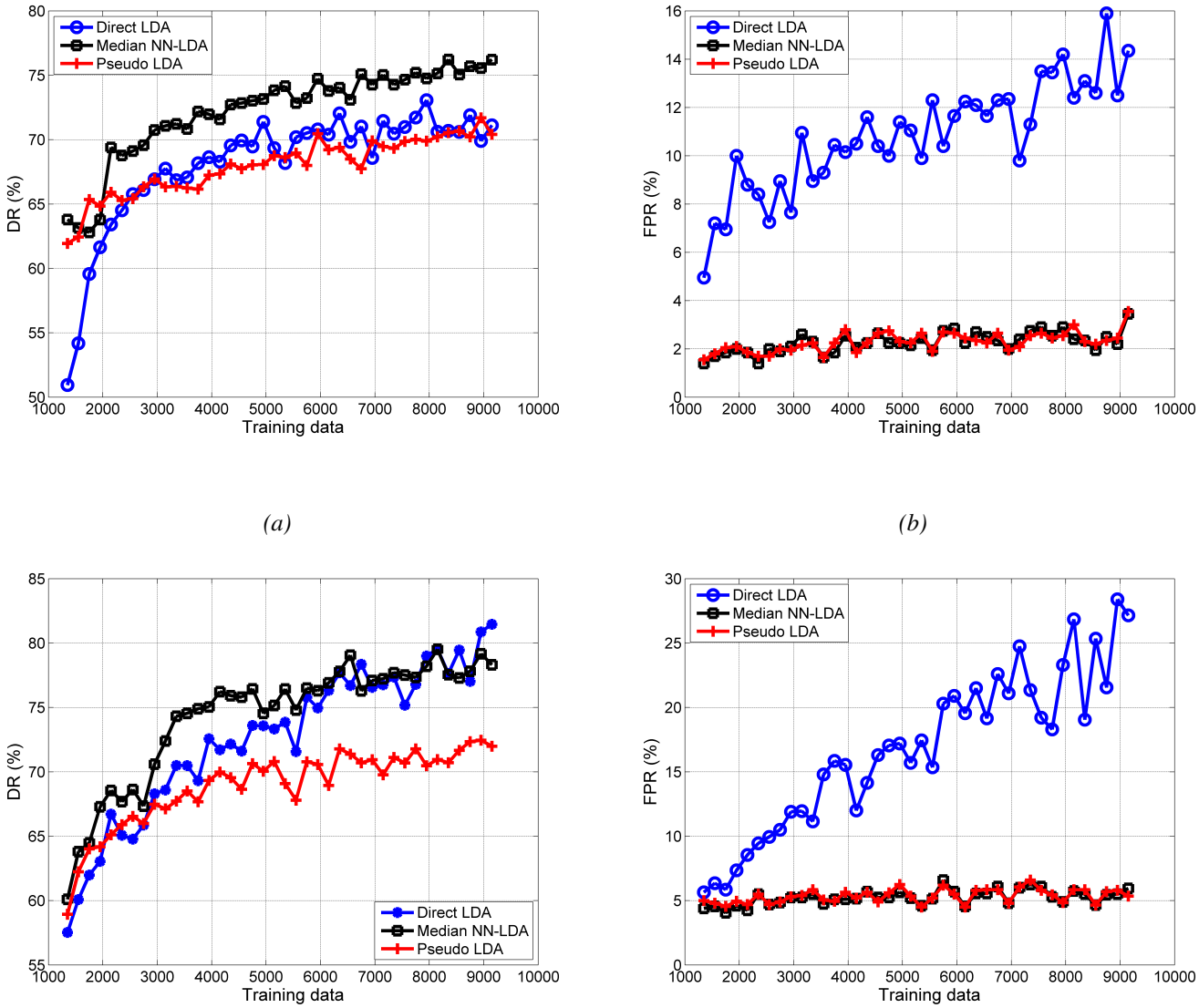


Figure 5.5: *Comparaison de median NN-LDA avec Direct LDA et Pseudo LDA*

En ce qui concerne la première base de données, nous observons à partir de la figure (5.5).a que median NN-LDA dépasse les deux approches une fois la taille des données est supérieure à 2000. D'autre part, la figure (5.5).b montre que Pseudo LDA et l'approche proposée produisent le minimum taux de faux positifs.

Dans le cas où nous utilisons NSL-KDD, la figure (5.5).c montre qu'en termes de DR, l'approche proposée surpasse Direct LDA et Pseudo LDA lorsque la taille des données est in-

férieure à 8000. Une fois cette valeur dépassée, Direct LDA commence a rivaliser avec median NN-LDA. Concernant le FPR, la figure (5.5).d affirme que notre approche donne toujours des résultats satisfaisants.

Pour conclure, afin de venir à bout du problème d'incohérence des données à l'intérieur de chaque classe. Une nouvelle méthode d'extraction de caractéristiques appelée Median NN-LDA est proposée. Dans cette approche, nous exploitons la médiane de chaque classe pour calculer les matrices de dispersion. Il y a deux avantages de cette approche. Elle préserve les distributions locales et globales et elle est peu sensible aux données non gaussiennes. Par conséquent, la méthode proposée est plus robuste que l'analyse discriminante linéaire traditionnelle.

5.4 Geometric Mean LDA

Dans les formulations mathématiques de PCA et LDA, les moyennes des classes jouent un rôle très important. Pour la première technique d'extraction de caractéristiques, la moyenne de classe contribue à définir la matrice de covariance. Pour LDA, les vecteurs moyens participent à la création des matrices de dispersions inter-classes et intra classes. Cependant, ces vecteurs sont estimés d'une manière classique. Comme il existe de nombreuses valeurs aberrantes et certaines classes qui contiennent peu d'échantillons d'apprentissage, il devient difficile de donner une estimation précise des vecteurs moyens de classe en utilisant les moyennes classiques. Afin de résoudre le problème de calcul de la moyenne optimale, de nombreux articles se sont penchées sur différentes approches. Par exemple, l'article [95] utilise le vecteur maximum -minimum-median pour estimer la moyenne. Le travail [181] propose un LDA basé sur la moyenne harmonique. Cette approche montre de bons résultats lorsqu'elle est appliquée à de nombreuses bases de données. Pour surmonter cette faiblesse dans le contexte de la détection d'intrusion, cet article propose d'utiliser la moyenne géométrique [145] pour estimer le vecteur moyen des classes. Ce dernier est moins sensible aux valeurs aberrantes. Ainsi, le modèle de LDA proposée devrait être plus robuste.

Dans ce qui suit, on formule théoriquement l'approche proposée et on la valide en se basant sur des test expérimentaux.

5.4.1 La moyenne géométrique

Dans la théorie des probabilités et des statistiques, la moyenne géométrique m_g d'un ensemble de M nombres positifs $x_1, x_2, ..., x_M$ est:

$$m_g = (x_1 \times x_2 \times ... \times x_M)^{\frac{1}{M}} \tag{5.27}$$

Comme la moyenne classique, la moyenne géométrique [145] peut être utilisée pour estimer la tendance centrale des données. En plus, elle est généralement considérée plus résistante aux valeurs aberrantes. Ceci, on peut le constater à travers cet exemple : supposons on a les données Data=[3.3, 3.0, 10, 3.1, 1, 3.2, 3.4] avec les valeurs aberrantes "1" et "10". On a m= 3.857 et m_g=3.186. On observe que 3.186 represente plus la tendance centrale ($\frac{3+3.1+3.2+3.3+3.4}{5} = 3.2$)

que 3.857. La moyenne géométrique d'une matrice non negative :

$$Z = [Z_1, Z_2, .., Z_M] = \begin{bmatrix} Z_{11} & Z_{21} & Z_{31} & \dots & Z_{M1} \\ Z_{12} & Z_{22} & Z_{32} & \dots & Z_{M2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Z_{1n} & Z_{2n} & Z_{3n} & \dots & Z_{Mn} \end{bmatrix}$$

est donnée par $m_g = (m_{g1}, m_{g2}, .., m_{gM})$ tel que m_{gi} est la moyenne géométrique des éléments de la ième colonne de la matrice de données Z.

5.4.2 Geometric Mean LDA

Dans le cas ou il existe peu de données d'apprentissage, la moyenne géométrique m_{gi} représente généralement la vraie tendance centrale, particulierement en présence des valeurs aberrantes. En plus, contrairement à de nombreuses méthodes d'élimination qui suppriment les valeurs aberrantes des données d'apprentissage. La moyenne géométrique peut en tirer des informations utiles. En se basant sur tous ces qualités , m_{gi} et m_g seront utilisés comme estimateurs des moyennes des classes au lieu de c_i et c. Pour éviter la singularité de la matrice de dispersion intra-classe, nous appliquerons PCA [76] sur X et obtiendrons la matrice X_{PCA} à l'aide de l'équation :

$$X_{PCA} = (W_1)^T X \tag{5.28}$$

Tel que W_1 est la matrice contenant les composantes principales (PCs) de X. Ensuite, au lieu de travailler avec X on opère sur $|X_{PCA}|$. On considére la valeur absolue de X_{PCA} afin de rendre le calcul de la moyenne géométrique (5.27) possible. Par la suite, on calcule les nouvelles matrices S_w^g et S_b^g avec les formules suivantes :

$$S_w^g = \frac{1}{M} \sum_{i=1}^{k} \sum_{x \in X_i} (x - m_{gi})(x - m_{gi})^T \tag{5.29}$$

$$S_b^g = \frac{1}{M} \sum_{i=1}^{k} (m_{gi} - m_g)(m_{gi} - m_g)^T \tag{5.30}$$

Le nouveau critère de Fisher est défini comme :

$$W' = \arg\max \frac{W'^T S_b^g W'}{W'^T S_w^g W'} \tag{5.31}$$

Les solutions de ce problème sont obtenues par la résolution de l'equation :

$$(S_w^g)^{-1}(S_b^g) w_i' = \lambda_i' w_i' \tag{5.32}$$

Tel que $W' = [w_1', \dots, w_l']$. La projection d'un vecteur x_{new} dans l'espace construit par notre approche se fait par :

$$t_i = (W')^T x_{new} \tag{5.33}$$

Vu qu'il utilise la moyenne géométrique, l'algorithme proposé sera nommé geomean LDA.

5.4.3 Tests Expérimentaux

Plusieurs expériences ont été réalisées pour démontrer l'efficacité de notre approche. Nous comparons geomean LDA à d'autres méthodes telles que LDA [58], Direct LDA [177], median LDA [175], espace vide LDA [32].

Nous décidons de varier le nombre d'échantillons d'apprentissage et de conserver les données de test inchangé avec la composition suivante (100 données normales, 100 données DOS, 50 données U2R, 100 données R2L et 100 PROBE). La manière dont nous modifions les échantillons d'apprentissage consiste d'une part à augmenter le nombre d'attaques DOS et PROBE, d'autre part, nous fixons des données d'apprentissage normales à 1000 échantillons. Les échantillons U2R et R2L sont fixés à 100. Afin d'obtenir un taux de détection (DR) et un FPR réalistes, l'opération de sélection de l'échantillon a été effectuée de manière aléatoire une trentaine de fois. Ensuite, DR et FPR prennent la moyenne des valeurs trouvées.

Table 5.1: *DR(%) de geomean LDA, LDA et median LDA sous différents espaces*

La méthode	PCs	LDs	KDDcup99	NSL-KDD
geomean LDA	3	3	60.60	60.38
	3	2	58.26	59.09
	3	1	48.52	52.39
LDA	3	3	61.68	54.35
	3	2	59.61	58.43
	3	1	50.45	42.70
median LDA	3	3	60.63	58.91
	3	2	59.71	56.44
	3	1	40.88	42.12

Le but de la première expérience est de trouver la dimension adéquate du sous-espace transformé par PCA, de sorte que les variantes de LDA puissent être appliquées et donner des résultats optimaux (DR élevé et FPR minimal). Pour atteindre cet objectif on a fixé les données à 1000 normale, 100 DOS, 50 U2R, 100 R2L, 100 PROBE. Ensuite, on a appliqué LDA dans des espaces

Table 5.2: *DR(%) de Direct LDA et Null space LDA sous différents espaces*

Base de test	La méthode	5 LDs	4 LDs	3 LDs
kddcup99	Null space LDA	59.48	60.86	59
	Direct LDA	42.85	46.28	60.28
NSL-KDD	Null space LDA	57.58	58.84	58.37
	Direct LDA	47.14	46.85	50.28

ayants différentes dimensions et récupéré ceux qui assurent un bon DR. Le tableau 5.1 montre les résultats de cette manipulation sur les deux bases de données. les LD désignent le nombre de vecteurs discriminants. Nous observons que le choix de trois PC et trois LD contribue significativement à l'obtention d'un DR élevé concernant median LDA et l'approche proposée. LDA traditionnelle excelle avec trois PC et deux LD sur NSL-KDD. Dans le même état d'esprit, nous recherchons le nombre de LD qui améliorent l'efficacité des autres modèles de LDA. D'après le tableau 5.2, nous notons que trois vecteurs discriminants assurent un bon DR pour direct LDA. Null space LDA a besoin de quatre vecteurs discriminants. Par la suite, les expériences suivantes exploitent ces derniers résultats.

Les figures 5.6.a et 5.6.b illustrent la comparaison des résultats de notre approche avec les variantes de LDA mentionnées. D'après la figure 5.6.a, nous observons que geomean LDA surpasse tous les variantes de LDA une fois les données dépassent 2000. La raison derrière ce phénomène semble être que plus il y a d'échantillons d'apprentissage plus l'effet des valeurs aberrantes est visible. Puisque les autres modèles de LDA à l'exception de median LDA utilisent la moyenne classique dans leurs formulations, ils seront plus sensibles aux valeurs aberrantes, ce qui diminuera leur efficacité. Median LDA est également résistante aux données aberrantes car elle estime la moyenne des classes par une mesure robuste qui est la médiane. Cependant, elle est moins puissante que geomean LDA dans ce cas. Une explication possible de ce fait peut résider dans la nature de la distribution des données. Nous nous attendons à ce que les connexions suivent une distribution log-normale, ce qui favorise la moyenne géométrique. Concernant le FPR, la figure 5.6.b affirme que la méthode proposée produit un taux de faux positifs inférieur à 2,7 %. Cela signifie que la méthode permet de faire la différence entre les instances normales et les attaques.

Sur NSL-KDD, on peut constater à partir de la figure 5.6.c que geomean LDA améliore de 3% le taux de détection par rapport à LDA et null space LDA, soit 1% par rapport à median LDA. L'approche dépasse en permanence Direct LDA. En terme de FPR, la figure 5.6.d affirme que geomean LDA produit un minimum de faux positives en comparaison avec median LDA et LDA.

Dans l'expérience suivante, nous avons évalué la méthode proposée en changeant le paramètre

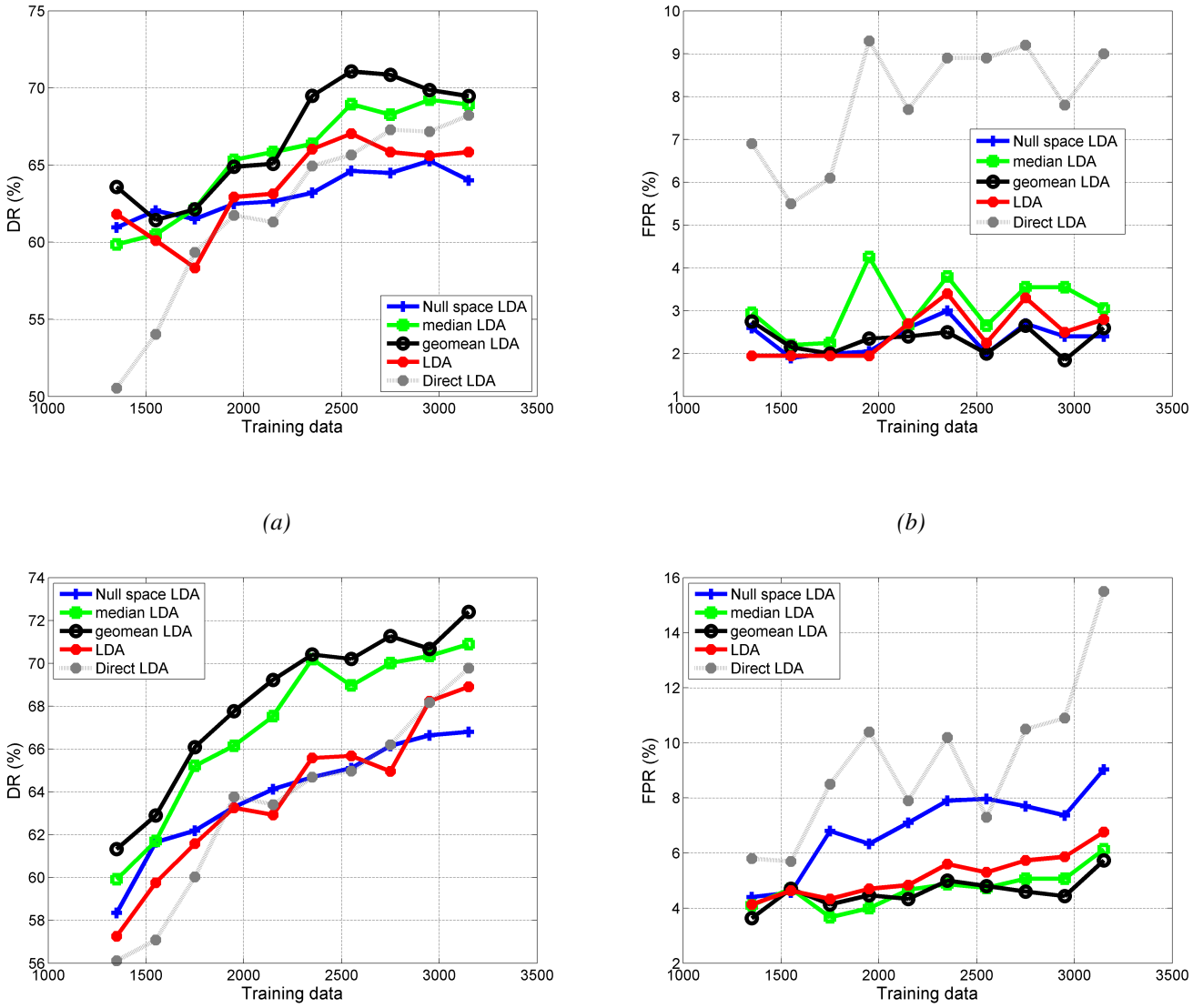


Figure 5.6: *DR et FPR de geomean LDA, LDA, Null space LDA et median LDA*

K du classifieur K-NN. Afin de rendre la manipulation possible, nous avons fixé un certain nombre de données d'apprentissage à : 1000 normales, 100 DOS, 50 U2R, 100 R2L et 100 instances de type PROBE. Ensuite, nous avons augmenté K et visualisé l'effet sur DR et FPR.

A partir de la figure 5.7.a, nous pouvons constater que geomean LDA conserve sa supériorité en donnant un DR élevé. Il produit au moins 62% et atteint 65% comme taux de détection maximal lorsque $K = 5$. De plus, la figure affirme que la méthode proposée depasse toutes les autres variantes de LDA mentionnées. En termes de taux de faux positifs, nous observons à partir de la figure 5.7.b que geomean LDA a le taux de faux positifs le plus faible avec ceux de nul space LDA et LDA.

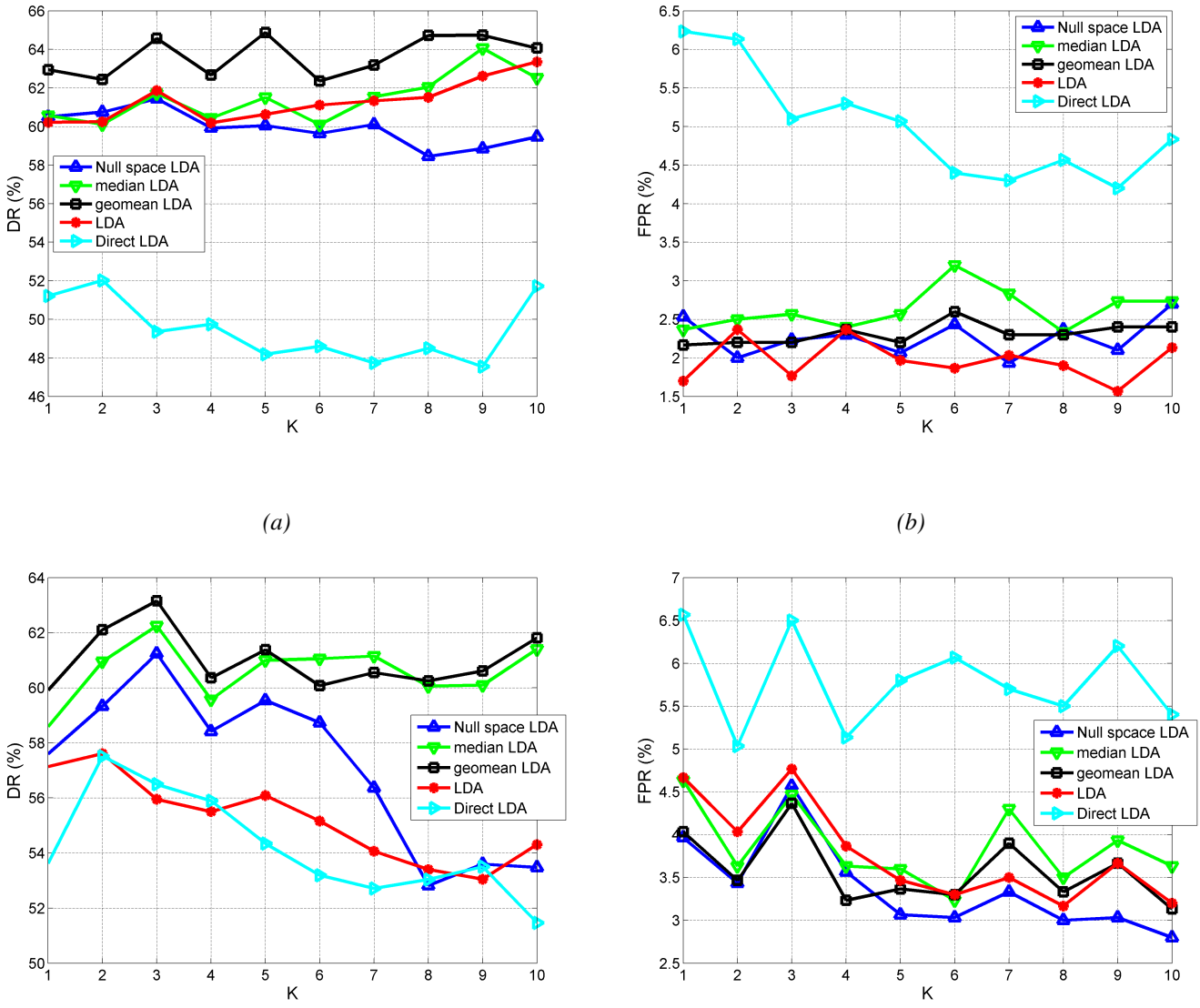


Figure 5.7: *DR(%) et FPR(%) des variantes de LDA en fonction des K plus proches voisins*

Lorsque nous reproduisons la même expérience sur NSL-KDD, nous obtenons les résultats suivants : A partir de la figure 5.7.c, il est clair que geomean LDA et median LDA sont les variantes de LDA qui assurent un meilleur DR. Le taux de faux positifs (5.7.d) de la méthode proposée est acceptable. En fait, il produit moins de FPR que Direct LDA et median LDA. Cependant, les autres méthodes prennent l'avantage une fois que K surpasse 4.

Dans cette section, nous avons amélioré la robustesse de LDA en introduisant la moyenne géométrique dans la formulation mathématique. Par conséquent, la méthode proposée appelée geomean LDA devient plus résistante face aux valeurs aberrantes et conserve des informations discriminantes utiles.

5.5 La combinaison de R1-PCA et Median LDA

L'idée qui consiste à introduire R1-PCA dans la formulation du median LDA afin d'améliorer la détection d'intrusion est basée sur les raisons suivantes:

1. La combinaison de la norme R1 avec la médiane offre davantage de résistance face aux données aberrantes.
2. L'obtention de plus de variance intra et inter classes.
3. Eviter la singularité des matrices rencontrée par median LDA.

Avant d'attaquer la formulation mathématique de l'approche proposée, on décrit celle de R1-PCA et median LDA.

5.5.1 R1-PCA

R1-PCA [46] est une variante de PCA. Elle propose d'utiliser la norme rotationnelle R1 et essaie d'obtenir les directions principales en recherchant les vecteurs propres d'une matrice de covariance robuste. Mathématiquement, nous essayons de trouver (PC) en minimisant :

$$||X-(PC)(PC)^T X||_{R1} \tag{5.34}$$

Où

$$|||X||_{R1} = \sum_{i=1}^{M}(\sum_{j=1}^{n} x_{ji}^2)^{\frac{1}{2}} \tag{5.35}$$

Ce qui mène d'après [46] à résoudre l'équation :

$$PC = arg\max(|(PC)^T C_r(PC)|) \tag{5.36}$$

Tel que :

$$C_r = \sum_i y_i x_i x_i^T \tag{5.37}$$

et $y_i = 1/||x_i-(PC)(PC)^T x_i||$

En utilisant y_i, on diminue la contribution des "outliers". L'équation (5.36) sera réécrite comme suit :

$$C_r(PC)_k = \lambda_k(PC)_k \tag{5.38}$$

La solution optimale globale PC est atteinte grâce à l'idée suivante: Comme valeur initiale, on travaille avec le PC^0 fourni par la matrice de covariance standard. Ensuite, à partir de PC^0, on calcule la matrice R1-covariance donnée par $Cr(PC^0)$. Après cela, on obtient une base orthogonale de PC en utilisant une fonction "orthogonalize" :

$$PC^{t+1/2} = C_r(PC^t)PC^t \tag{5.39}$$

$$PC^{t+1} = orthogonalize(PC^{t+1/2}) \tag{5.40}$$

En employant un algorithme itératif illustré par Algorithm 6, PC^t converge vers les vecteurs propres de C_r.

Algorithm 6 : Algorithme de R1-PCA

1. Entrée : matrice de données X, la dimension réduite k
2. Utiliser PCA pour obtenir PC^0
3. Initialiser PC : $PC = PC^0$.
4. Modifier PC en se basant sur les équations (5.39) et (5.40) et itérer jusqu'à convergence.
5. Sortie : PC

R1-PCA a une formulation plus solide pour gérer les valeurs aberrantes, malheureusement, il a une importante faiblesse. Il offre un grand poids aux caractéristiques ayants une variabilité plus élevée, qu'elles soient efficaces ou non. Ce fait mène a la situation où la direction principale sélectionnée représente l'attribut ayant la variabilité la plus élevée mais ayant un manque de discrimination.

5.5.2 Median LDA

median LDA [175] utilise la médiane de classe med_i pour estimer le vecteur moyen de la classe c_i et med' pour estimer c . Yang *et al.* [175] considèrent que la médiane représente plus précisément la tendance centrale que la moyenne traditionnelle. Rappelons que pour localiser la médiane d'une liste finie de nombres, nous devons classer ces nombres dans un ordre croissant. Ensuite, choisir la valeur qui existe au milieu. les nouvelles S'_w et S'_b s'écrivent sous forme de :

$$S'_w = \frac{1}{M}\sum_{i=1}^{k}\sum_{i\in X_i}(x - med_i)(x - med_i)^T \tag{5.41}$$

$$S'_b = \frac{1}{M}\sum_{i=1}^{k}(med_i - med')(med_i - med')^T \tag{5.42}$$

Le critère de Fisher sera réécrit comme :

$$W' = \arg\max \frac{W'^T S'_b W'}{W'^T S'_w W'} \tag{5.43}$$

Pour obtenir la matrice de projection, les auteurs effectuent :

$$S'^{-1}_w S'_b w'_i = \lambda'_i w'_i \tag{5.44}$$

Malheureusement, la median LDA souffre du "small sample size problem". C'est ce qui rend le calcul de $S_w'^{-1}$ impossible.

5.5.3 La formulation mathématique de la méthode proposée

Le but de la méthode est de trouver une matrice de projection W'' donnée par :

$$W'' = (PC)W' \tag{5.45}$$

Où,

$$PC = arg\max(|(PC)^T C_r (PC)|) \tag{5.46}$$

$$W' = arg\max \frac{|W'^T (PC)^T S_b' (PC)W'|}{|W'^T (PC)^T S_w' (PC)W'|} \tag{5.47}$$

La projection d'une nouvelle connexion x_{new} dans l'espace construit par notre approche est obtenue par:

$$t_i = (W_i'')^T x_{new} \tag{5.48}$$

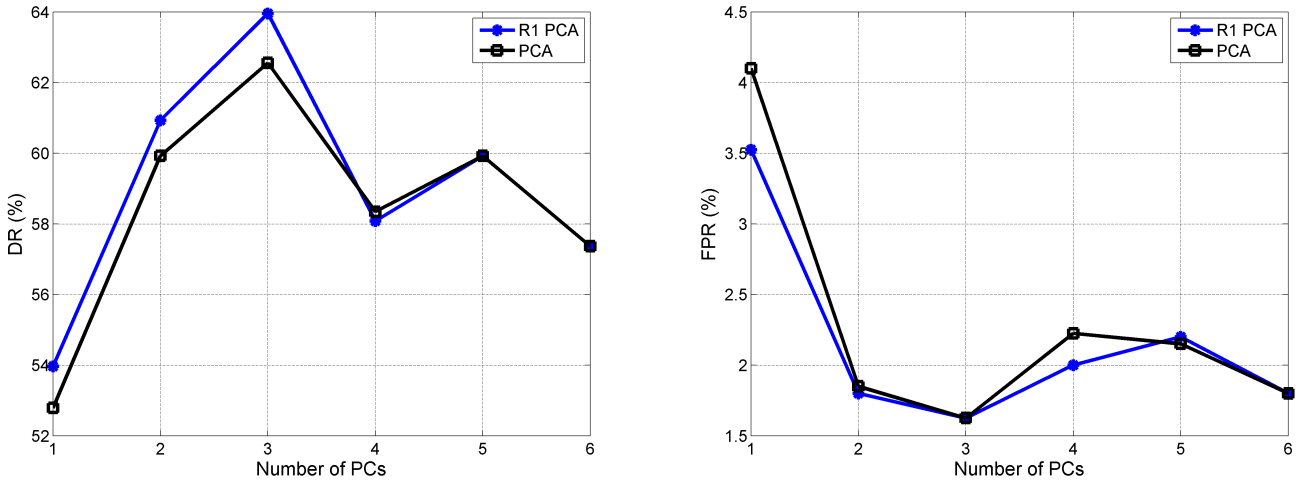
5.5.4 Tests expérimentaux

Dans la première expérience, nous cherchons le nombre adéquat de composantes principales (PC) qui mènent R1-PCA à produire une meilleur performance. Puisque la sélection automatique de (PCs) fait l'objet de sujets de recherche en cours, nous décidons de faire varier le nombre de PC de 1 à 6 et de choisir celui qui donne le meilleur DR et le FPR le plus bas. Les figures 5.8.a et 5.8.b indiquent que 3 PCs représentent le bon choix. En même temps, nous observons que R1-PCA dépasse PCA, à cause de sa forte résistance aux valeurs aberrantes. En fait, si un échantillon est aberrant, il obtiendra un faible poids y_i durant le processus de R1-PCA. Dans le cas idéal, le y_i correspondant à la valeur aberrante tendrait vers zéro. Ceci élimine les valeurs aberrantes de l'ensemble des échantillons d'apprentissage de sorte que les vecteurs propres calculés par R1-PCA soient plus représentatifs.

Dans l'expérience suivante, nous prouvons l'efficacité de R1-PCA+median LDA en la comparant à PCA + LDA, PCA + Median LDA et Null space LDA.

En se basant sur l'expérience précédente, nous choisissons 3 composantes principales dans la première étape de ces méthodes. Dans la deuxième étape, nous choisissons 3 LDs. Null space LDA prend en compte les 4 vecteurs discriminants supérieurs. Ensuite, nous augmentons le nombre des données d'apprentissage et nous évaluons leur influence sur DR et FPR. D'un cote, nous observons à partir de la figure 5.9.a que notre approche dépasse les autres variantes de LDA en termes de DR. D'un autre côté, il est montré sur la figure 5.9.b que l'approche produit le FPR le plus bas une fois le nombre de données d'apprentissage dépasse 3500.

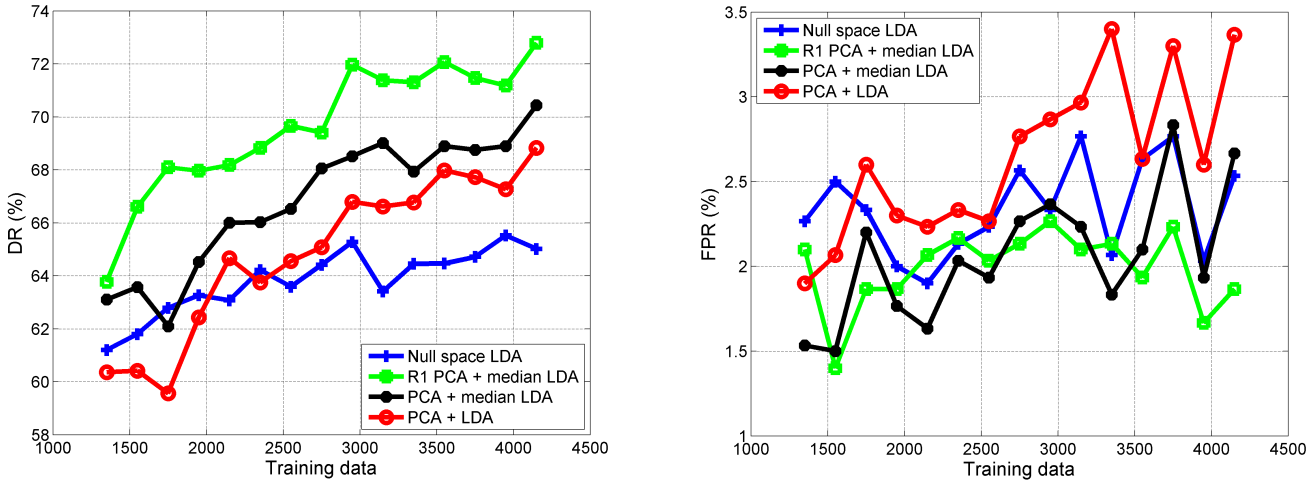
Pour aller plus loin dans notre investigation, nous avons calculé le taux de détection des approches LDA précitées pour chaque type d'attaque (DOS, U2R, R2L et PROBE). Selon le tableau 5.3, nous notons que R1-PCA+median LDA détecte plus efficacement les attaques DOS



(a) *(b)*

Figure 5.8: *DR(%) et FPR(%) de R1-PCA et PCA pour KDDcup99*

et PROBE. Néanmoins, les attaques U2R et R2L ne sont pas correctement identifiées à cause de leur rareté.



(a) *(b)*

Figure 5.9: *Comparaison de R1-PCA+median LDA avec d'autre variantes de LDA*

Dans cette section nous avons considéré une fusion de deux puissants algorithmes de réduction de caractéristiques pour améliorer la détection d'intrusions d'anomalies. L'approche consiste à combiner R1-PCA et médiane LDA. En conséquence, nous bénéficions des avantages suivants : plus de résistance contre les valeurs aberrantes, on fournit une variance de données supplémentaire intra et inter-classes. On mène les composants principaux de R1-PCA de plus de discrimination,

Table 5.3: *Echantillons d'apprentissage vs. DR (%) individuel de R1-PCA+median LDA*

Training data	method	DOS	U2R	R2L	PROBE
1350	Null space LDA	80.2	15.4	2.3	77.3
	R1-PCA+median LDA	**95.03**	14.4	0.6	71.6
	PCA+LDA	86.8	14	2.06	76.7
	PCA+median LDA	85.9	12.3	2.9	64.5
2150	Null space LDA	87.4	15.3	0.8	82.06
	R1-PCA+median LDA	**94.96**	14.26	0.9	**82.16**
	PCA+LDA	84.13	14.06	2.3	77
	PCA+median LDA	86.8	11.9	2.5	74.4
3350	Null space LDA	88.06	13.7	0.7	84.7
	R1-PCA+median LDA	**94.7**	13.8	0.6	**86.9**
	PCA+LDA	86.6	13.3	1.4	81.1
	PCA+median LDA	87.6	10.5	1.7	78.7
4150	Null space LDA	89	15.4	0.16	86.13
	R1-PCA+median LDA	**95.5**	14.4	1.1	**86.7**
	PCA+LDA	86.9	14.3	1.3	82.7
	PCA+median LDA	88.9	10.60	1.6	80.13

on résout enfin le problème (SSS) rencontré par median LDA.

5.6 Conclusion

Dans ce chapitre nous avons proposé trois méthodes basées sur l'analyse discriminante linéaire pour la détection des intrusions. Chacune essaye de contourner une faiblesse spécifique de LDA. Tout d'abord, median NN-LDA est une méthode qui a été conçue pour résoudre le problème d'inconsistance des données. Dans cette approche, nous exploitons la médiane de chaque classe pour calculer les matrices de dispersion. Par conséquent, on bénéficie de deux avantages. On préserve les distributions locales et globales d'une part et on réduit la sensibilité aux classes non gaussiennes d'autre part.

Afin de lutter contre l'effet des données aberrantes, geomean LDA améliore la robustesse de LDA en introduisant la moyenne géométrique dans la formulation mathématique. Ainsi, la méthode conserve des informations discriminantes utiles et mène à un taux de détection satisfaisant.

Pour achever le même but, on a proposé de combiner R1-PCA avec median LDA. En plus de sa résistance face aux "outliers", l'approche fournit une variance de données supplémentaire intra et inter-classes, elle mène les composants principaux de R1-PCA de plus de discrimination, enfin elle résout le problème (SSS) rencontré par median LDA.

Pour démontrer l'efficacité des méthodes proposée, nous les comparons a LDA, Direct LDA , null space LDA et pseudo LDA.

Conclusion générale et perspectives

Le système de détection d'intrusion comportementale souffre de plusieurs faiblesses, tel que l'étroite dépendance à l'environnement cible, la difficulté liée à l'évaluation, et les limites de performance qui mènent à une augmentation de fausses alarme et la diminution de taux de détection.

Les limites de performance sont causées généralement par l'incapacité de manipuler un grand trafic réseau contenant des informations inutiles. Afin de pallier cette carence, nos travaux contribuent à la mise au point de nouveaux algorithmes d'extraction de caractéristiques basées sur l'analyse en composantes principales (PCA) et l'analyse discriminante linéaire (LDA). Ces méthodes visent à trouver un sous-espace de projection dans lequel les connexions réseau ont une dimension réduite tout en gardant l'information essentielle.

PCA permet de réduire la dimension d'un trafic réseau tout en maximisant sa variance. Par ailleurs, cette méthode souffre de deux limitations. La première concerne sa nature linéaire. Vu que les connexions réseau peuvent avoir une structure non linéaire, le PCA classique ne sera pas très utile dans ce cas-là. Pour remédier à cette limitation, nous proposons l'utilisation de KPCA avec des nouvelles fonctions à noyau : le noyau a puissance et le noyau sphérique. L'idée de base de KPCA est fondée sur l'utilisation de l'astuce du noyau "kernel trick" pour transformer les données d'entrée dans un espace de caractéristiques implicite. Puis, ces données sont traitées dans cet espace pour produire des caractéristiques non linéaires et discriminantes qui facilitent la classification des connexions.

Une autre limitation de PCA concerne sa sensibilité aux données aberrantes (outliers). Pour y remédier, nous avons proposé un nouveau PCA-Lp Basée sur le gradient conjugué au lieu du gradient simple utilisé dans la littérature. Cette technique tente de trouver des projections qui maximisent la covariance totale en utilisant la norme Lp ($p < 2$) au lieu de la norme L2. Parmi les avantages les plus notables du gradient conjugué on cite la faible exigence de la mémoire et la vitesse de convergence. Dans le même sens, cette thèse propose une méthode appelée QR-OMPCA qui partage la même philosophie que celle de PCA-Lp. Elle cherche un modèle plus robuste face aux données aberrantes et essaye de corriger une mal formulation de PCA-Lp(p=1) et R1-PCA en incluant un processus de calcul de la moyenne optimale. Cette approche est caractérisée par l'utilisation de la décomposition QR au lieu du SVD employée dans un travail antérieur.

Par conséquent, l'intégration de QR-OMPCA dans le système de détection d'intrusion (IDS) rend ce dernier plus efficace et plus rapide.

Afin de déterminer le sous-espace de projection, les approches basées sur (LDA) maximisent la séparabilité entre classes tout en minimisant la séparation intra classe. Par ailleurs, la méthode souffre de plusieurs problèmes. Le premier est connu sous le nom de "Small Sample Size Problem" (SSS) qui surgit avec les données qui ont une grande dimension par rapport au faible nombre d'échantillons d'apprentissage. Pour y remédier plusieurs travaux ont été proposés.

Un deuxième problème découle de la formulation de LDA. Cette dernière exploite la moyenne classique pour calculer les matrices de dispersion. Or l'utilisation de cette moyenne amplifie l'effet des valeurs aberrantes dans les données et mène à une projection erronée. Pour éviter cela, on propose geomean LDA et R1-PCA+median LDA. La première variante remplace la moyenne classique utilisée dans LDA par une moyenne géométrique plus robuste face aux données aberrantes. La deuxième combine deux puissants algorithmes pour profiter des avantages de la norme R1 et du median.

Un troisième problème concerne le type de structure de données manipulée par LDA qui accorde plus d'attention à la structure globale des classes. En conséquence, les caractéristiques discriminantes produites sont souvent imprécises. Pour corriger cette carence, Median NN-LDA se base sur les connexions les plus proches du médiane de chaque classe pour préserver les distributions locales et globales. Par conséquent, cette méthode offre une meilleure séparabilité entre les classes dans l'espace réduit. Ce qui facilite la détection des intrusions par la suite.

Finalement, nous envisageons de nombreuses perspectives de ces travaux autour des différents aspects abordés durant cette thèse.

Généralement, tous les algorithmes basés sur LDA développés au cours de cette thèse supposent que chaque connexion peut appartenir à une seule classe. Toutefois, dans la réalité, cette affectation ne peut se faire sans ambiguïté, et cela pour diverses raisons. Par exemple, lorsque les informations disponibles sur les connexions sont incertaines ou incomplètes ou bien si les différentes classes se chevauchent et les frontières entre elles ne sont pas très claires. L'ambiguïté ne permet pas donc une identification certaine des classes, ce qui fausse les résultats de classification par la suite. Pour résoudre ces problèmes Zadeh [179] a introduit la théorie des sous-ensembles flous permettant à chaque élément d'appartenir plus ou moins à plusieurs classes, par le biais d'une fonction d'appartenance floue. Ceci dit, nous envisagons dans des travaux ultérieurs l'intégration de la théorie des sous-ensembles flous dans les nouvelles variantes de LDA afin d'améliorer davantage la détection des intrusions.

Par ailleurs, il serait aussi intéressant d'appliquer les méthodes développées dans ce mémoire, dans un système de détection d'intrusions opérationnel. Dans ce cas, ces algorithmes doivent répondre aux exigences du temps réel.

Bibliographie

[1] URL: http://kdd.ics.uci.edu/databases/kddcup99/.

[2] URL: http://nsl.cs.unb.ca/KDD/NSLKDD.html.

[3] Tarek Abbes. “Classification du trafic et optimisation des règles de filtrage pour la détection d’intrusions”. In: *université Henri Poincaré-Nancy* (2004).

[4] A Abdel-Aziz and J Esler. “Intrusion detection & response-leveraging next generation firewall technology”. In: *SANS-Institue, Tech. Rep* (2009).

[5] Eric Alata. “Observation, caractérisation et modélisation de processus d’attaques sur Internet”. PhD thesis. INSA de Toulouse, 2007.

[6] Ion Alberdi, Jean Gabes, and Emilien L Jamtel. “UberLogger: un observatoire niveau noyau pour la lutte informatique défensive”. In: *Symposium sur la Sécurité des Technologies de l’Information et des Communications (SSTIC-2005), Rennes, France*. 2005.

[7] Fatemeh Amiri et al. “Mutual information-based feature selection for intrusion detection systems”. In: *Journal of Network and Computer Applications* 34.4 (2011), pp. 1184–1199.

[8] James P Anderson et al. *Computer security threat monitoring and surveillance*. Tech. rep. Technical report, 1980.

[9] Matej Artac, Matjaz Jogan, and Ales Leonardis. “Incremental PCA for on-line visual learning and recognition”. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 3. IEEE. 2002, pp. 781–784.

[10] Paul Baecher et al. “The nepenthes platform: An efficient approach to collect malware”. In: *Recent Advances in Intrusion Detection*. Springer. 2006, pp. 165–184.

[11] B Balajinath and SV Raghavan. “Intrusion detection through learning behavior model”. In: *Computer Communications* 24.12 (2001), pp. 1202–1212.

[12] Jai Sundar Balasubramaniyan et al. “An architecture for intrusion detection using autonomous agents”. In: *Computer security applications conference, 1998. Proceedings. 14th annual*. IEEE. 1998, pp. 13–24.

[13] Jay Beale et al. *Nessus network auditing*. Syngress Publishing, 2004.

[14] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 711–720.

[15] Mikhail Belkin and Partha Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation”. In: *Neural computation* 15.6 (2003), pp. 1373–1396.

[16] Salem Benferhat, Tayeb Kenaza, and Aicha Mokhtari. “A naive bayes approach for detecting coordinated attacks”. In: *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International*. IEEE. 2008, pp. 704–709.

[17] Leyla Bilge et al. “EXPOSURE: a passive DNS analysis service to detect and report malicious domains”. In: *ACM Transactions on Information and System Security (TISSEC)* 16.4 (2014), p. 14.

[18] Leyla Bilge et al. “EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.” In: *Ndss*. 2011.

[19] Philipe Biondi. *Architecture expérimentale pour la détection d'intrusions dans un système informatique*. 2001.

[20] Alan Bivens et al. “Network-based intrusion detection using neural networks”. In: *Intelligent Engineering Systems through Artificial Neural Networks* 12.1 (2002), pp. 579–584.

[21] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[22] Adel Bouhoula et al. “Firewall filtering rules analysis for anomalies detection”. In: *International Journal of Security and Networks* 3.3 (2008), pp. 161–172.

[23] Djemaa Boukhlouf. “Une approche à base d'agents mobiles pour la sécurité des systèmes d'informations sur le web”. PhD thesis. Université Mohamed Khider-Biskra, 2016.

[24] Yacine Bouzida et al. “Efficient intrusion detection using principal component analysis”. In: *3éme Conférence sur la Sécurité et Architectures Réseaux (SAR), La Londe, France*. 2004, pp. 381–395.

[25] Krupa Brahmkstri et al. “Ontology based multi-agent intrusion detection system for web service attacks using self learning”. In: *Networks and Communications (NetCom2013)*. Springer, 2014, pp. 265–274.

[26] David Brumley et al. “Towards automatic generation of vulnerability-based signatures”. In: *Security and Privacy, 2006 IEEE Symposium on*. IEEE. 2006, 15–pp.

[27] José Camacho et al. “PCA-based multivariate statistical network monitoring for anomaly detection”. In: *Computers & Security* 59 (2016), pp. 118–137.

[28] John E Canavan. *Fundamentals of network security*. Artech House, 2001.

[29] James Cannady. “Artificial neural networks for misuse detection”. In: *National information systems security conference*. 1998, pp. 368–81.

[30] Srilatha Chebrolu, Ajith Abraham, and Johnson P Thomas. “Feature deduction and ensemble design of intrusion detection systems”. In: *Computers & security* 24.4 (2005), pp. 295–307.

[31] Hwann-Tzong Chen, Huang-Wei Chang, and Tyng-Luh Liu. “Local discriminant embedding and its variants”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 846–853.

[32] Li-Fen Chen et al. “A new LDA-based face recognition system which can solve the small sample size problem”. In: *Pattern recognition* 33.10 (2000), pp. 1713–1726.

[33] Long-Sheng Chen and Jhih-Siang Syu. “Feature Extraction based Approaches for Improving the Performance of Intrusion Detection Systems”. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. Vol. 1. 2015, pp. 18–20.

[34] Bill Cheswick. “An Evening with Berferd in which a cracker is Lured, Endured, and Studied”. In: *Proc. Winter USENIX Conference, San Francisco*. 1992, pp. 20–24.

[35] Sung-Bae Cho and Sang-Jun Han. “Two sophisticated techniques to improve HMM-based intrusion detection systems”. In: *Recent Advances in Intrusion Detection*. Springer. 2003, pp. 207–219.

[36] Sung-Bae Cho and Hyuk-Jang Park. “Efficient anomaly detection by modeling privilege flows using hidden Markov model”. In: *Computers & Security* 22.1 (2003), pp. 45–55.

[37] Fan RK Chung. *Spectral graph theory*. 92. American Mathematical Soc., 1997.

[38] Pierre Comon. “Independent component analysis, a new concept?” In: *Signal processing* 36.3 (1994), pp. 287–314.

[39] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. “A survey of man in the middle attacks”. In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2027–2051.

[40] Frédéric Cuppens and Rodolphe Ortalo. “Lambda: A language to model a database for detection of attacks”. In: *Recent advances in intrusion detection*. Springer. 2000, pp. 197–216.

[41] Dao-Qing Dai and Pong C Yuen. “Regularized discriminant analysis and its application to face recognition”. In: *Pattern Recognition* 36.3 (2003), pp. 845–847.

[42] Herve Debar, Monique Becker, and Didier Siboni. “A neural network component for an intrusion detection system”. In: *Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on*. IEEE. 1992, pp. 240–250.

[43] Jonathan-Christofer Demay. “Génération et évaluation de mécanismes de détection des intrusions au niveau applicatif”. PhD thesis. Université Rennes 1, 2011.

[44] Dorothy E Denning. “An intrusion-detection model”. In: *IEEE Transactions on software engineering* 2 (1987), pp. 222–232.

[45] Whitfield Diffie and Martin Hellman. “New directions in cryptography”. In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[46] Chris Ding et al. “R 1-PCA: rotational invariant L 1-norm principal component analysis for robust subspace factorization”. In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 281–288.

[47] Steven T Eckmann, Giovanni Vigna, and Richard A Kemmerer. “STATL: An attack language for state-based intrusion detection”. In: *Journal of computer security* 10.1-2 (2002), pp. 71–103.

[48] Taher ElGamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[49] Z Elkhadir, K Chougdali, and M Benattou. “Intrusion Detection System Using PCA and Kernel PCA Methods.” In: *IAENG International Journal of Computer Science* 43.1 (2016).

[50] Zyad Elkhadir, Khalid Chougdali, and Mohamed Benattou. “Network Intrusion Detection System Using PCA by Lp-Norm Maximization Based on Conjugate Gradient”. In: *International Review on Computers and Software (IRECOS)* 11.1 (2016), pp. 64–71.

[51] Zyad Elkhadir, Khalid Chougdali, and Mohammed Benattou. “A Median Nearest Neighbors LDA for Anomaly Network Detection”. In: *International Conference on Codes, Cryptology, and Information Security*. Springer. 2017, pp. 128–141.

[52] Know Your Enemy. *Learning with VMware. Building Virutal Honeynets using VMware*. 2006.

[53] Know Your Enemy. *Sebek, A kernel based data capture tool, The Honeynet Project*. 2003.

[54] Ahmad Faour. “Une Architecture semi-supervisée et adaptative pour le filtrage d’alarmes dans les systèmes de détection d’intrusions sur les réseaux”. PhD thesis. INSA de Rouen, 2007.

[55] Reeves Fletcher and Colin M Reeves. “Function minimization by conjugate gradients”. In: *The computer journal* 7.2 (1964), pp. 149–154.

[56] Stephanie Forrest et al. “A sense of self for unix processes”. In: *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*. IEEE. 1996, pp. 120–128.

[57] Jérôme François, Issam Aib, and Raouf Boutaba. “FireCol: a collaborative protection network for the detection of flooding DDoS attacks”. In: *IEEE/ACM Transactions on Networking (TON)* 20.6 (2012), pp. 1828–1841.

[58] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.

[59] Debin Gao, Michael K Reiter, and Dawn Song. “Behavioral distance measurement using hidden markov models”. In: *RAID*. Springer. 2006, pp. 19–40.

[60] Gene H Golub and Charles F Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.

[61] Jean Goubault-Larrecq and Julien Olivain. “A Smell of Orchids.” In: *RV*. Springer. 2008, pp. 1–20.

[62] Kent Griffin et al. “Automatic Generation of String Signatures for Malware Detection.” In: *RAID*. Vol. 5758. Springer. 2009, pp. 101–120.

[63] Xiaohong Guan, Wei Wang, and Xiangliang Zhang. “Fast intrusion detection based on a non-negative matrix factorization model”. In: *Journal of Network and Computer Applications* 32.1 (2009), pp. 31–44.

[64] Mike Hall and Kevin Wiley. “Capacity verification for high speed network intrusion detection systems”. In: *International Workshop on Recent Advances in Intrusion Detection*. Springer. 2002, pp. 239–251.

[65] Trevor Hastie and Werner Stuetzle. “Principal curves”. In: *Journal of the American Statistical Association* 84.406 (1989), pp. 502–516.

[66] Simon S Haykin. *Neural networks: a comprehensive foundation*. Tsinghua University Press, 2001.

[67] Guy Helmer et al. “Lightweight agents for intrusion detection”. In: *Journal of systems and Software* 67.2 (2003), pp. 109–122.

[68] Michaël Hervieux and Thomas Meurisse. “Uml comme pota miel”. In: *Symposium sur la Sécurité des Technologies de l’Information et des Communications (SSTIC’03)*. 2003.

[69] Steven A Hofmeyr, Stephanie Forrest, and Anil Somayaji. “Intrusion detection using sequences of system calls”. In: *Journal of computer security* 6.3 (1998), pp. 151–180.

[70] Harold Hotelling. “Relations between two sets of variates”. In: *Biometrika* 28.3/4 (1936), pp. 321–377.

[71] Peg Howland and Haesun Park. “Generalizing discriminant analysis using the generalized singular value decomposition”. In: *IEEE transactions on pattern analysis and machine intelligence* 26.8 (2004), pp. 995–1006.

[72] Wenjie Hu, Yihua Liao, and V Rao Vemuri. “Robust Support Vector Machines for Anomaly Detection in Computer Security.” In: *ICMLA*. 2003, pp. 168–174.

[73] Rui Huang et al. “Solving the small sample size problem of LDA”. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 3. IEEE. 2002, pp. 29–32.

[74] Sumaiya Thaseen Ikram and Aswani Kumar Cherukuri. “Improving accuracy of intrusion detection model using PCA and optimized SVM”. In: *Journal of computing and information technology* 24.2 (2016), pp. 133–148.

[75] Wayne Jansen et al. “Applying mobile agents to intrusion detection and response”. In: *National Institute of Standards and Technology, Computer Security Division* (1999).

[76] Ian T Jolliffe. “Principal component analysis and factor analysis”. In: *Principal component analysis* (2002), pp. 150–166.

[77] Ali Kartit. “Une nouvelle approche de détection d’intrusions et étude des problèmes liés au déploiement de politiques de sécurité dans les réseaux informatiques”. In: (2011).

[78] Jon R Kettenring. “Canonical analysis of several sets of variables”. In: *Biometrika* 58.3 (1971), pp. 433–451.

[79] Amir R Khakpour and Alex X Liu. “First step toward cloud-based firewalling”. In: *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*. IEEE. 2012, pp. 41–50.

[80] Latifur Khan, Mamoun Awad, and Bhavani Thuraisingham. “A new intrusion detection system using support vector machines and hierarchical clustering”. In: *The VLDB Journal—The International Journal on Very Large Data Bases* 16.4 (2007), pp. 507–521.

[81] Jongsun Kim et al. “Effective representation using ICA for face recognition robust to local distortion and partial occlusion”. In: *IEEE transactions on pattern analysis and machine intelligence* 27.12 (2005), pp. 1977–1981.

[82] Andrew P Kosoresow and SA Hofmeyer. “Intrusion detection via system call traces”. In: *IEEE software* 14.5 (1997), pp. 35–42.

[83] Christopher Kruegel and Thomas Toth. “Using decision trees to improve signature-based intrusion detection”. In: *Recent Advances in Intrusion Detection*. Springer. 2003, pp. 173–191.

[84] Fangjun Kuang, Weihong Xu, and Siyang Zhang. “A novel hybrid KPCA and SVM with GA model for intrusion detection”. In: *Applied Soft Computing* 18 (2014), pp. 178–184.

[85] Fangjun Kuang et al. “A novel approach of KPCA and SVM for intrusion detection”. In: *Journal of Computational Information Systems* 8.8 (2012), pp. 3237–3244.

[86] Fangjun Kuang et al. “A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection”. In: *Soft Computing* 19.5 (2015), pp. 1187–1199.

[87] No Kwak. “Principal Component Analysis by L_p -Norm Maximization”. In: *IEEE Transactions on, Cybernetics* 44.5 (2014), pp. 594–609. ISSN: 2168-2267. DOI: 10.1109/TCYB.2013.2262936.

[88] Nojun Kwak. “Principal component analysis based on L1-norm maximization”. In: *IEEE transactions on pattern analysis and machine intelligence* 30.9 (2008), pp. 1672–1680.

[89] Kwok-Yan Lam, Lucas Hui, and Siu-Leung Chung. “A data reduction method for intrusion detection”. In: *Journal of Systems and Software* 33.1 (1996), pp. 101–108.

[90] Kwok Ho Law et al. “IDS false alarm filtering using KNN classifier”. In: *International Workshop on Information Security Applications*. Springer. 2004, pp. 114–121.

[91] Daniel D Lee and H Sebastian Seung. “Algorithms for non-negative matrix factorization”. In: *Advances in neural information processing systems*. 2001, pp. 556–562.

[92] Daniel D Lee and H Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755 (1999), pp. 788–791.

[93] Yuh-Jye Lee, Yi-Ren Yeh, and Yu-Chiang Frank Wang. “Anomaly detection via online oversampling principal component analysis”. In: *IEEE transactions on knowledge and data engineering* 25.7 (2013), pp. 1460–1470.

[94] Haifeng Li, Tao Jiang, and Keshu Zhang. “Efficient and robust feature extraction by maximum margin criterion”. In: *Advances in neural information processing systems*. 2004, pp. 97–104.

[95] Li Li, Hongwei Ge, and Jianqiang Gao. “Maximum–minimum–median average MSD-based approach for face recognition”. In: *AEU-International Journal of Electronics and Communications* 70.7 (2016), pp. 920–927.

[96] Xi Li et al. “Linear discriminant analysis using rotational invariant L 1 norm”. In: *Neurocomputing* 73.13 (2010), pp. 2571–2579.

[97] Xiaodong Li, Shumin Fei, and Tao Zhang. “Weighted maximum scatter difference based feature extraction and its application to face recognition”. In: *Machine Vision and Applications* 22.3 (2011), pp. 591–595.

[98] Yang Li and Li Guo. “An active learning based TCM-KNN algorithm for supervised network intrusion detection”. In: *Computers & security* 26.7 (2007), pp. 459–467.

[99] Yinhui Li et al. “An efficient intrusion detection system based on support vector machines and gradually feature removal method”. In: *Expert Systems with Applications* 39.1 (2012), pp. 424–430.

[100] Yihua Liao and V Rao Vemuri. “Use of k-nearest neighbor classifier for intrusion detection”. In: *Computers & security* 21.5 (2002), pp. 439–448.

[101] Jia-Ling Lin, Xiaoyang Sean Wang, and Sushil Jajodia. “Abstraction-based misuse detection: High-level specifications and adaptable strategies”. In: *Computer Security Foundations Workshop, 1998. Proceedings. 11th IEEE*. IEEE. 1998, pp. 190–201.

[102] Ulf Lindqvist et al. “Designing IDLE: The intrusion data library enterprise”. In: *Doktorsavhandlingar vid Chalmers Tekniska Hogskola* 1530 (1999), pp. 193–203.

[103] Richard P Lippmann and Robert K Cunningham. “Improving intrusion detection performance using keyword selection and neural networks”. In: *Computer Networks* 34.4 (2000), pp. 597–603.

[104] Alex X Liu. *Firewall design and analysis*. Vol. 4. World Scientific, 2011.

[105] Chengjun Liu and Harry Wechsler. “Comparative assessment of independent component analysis (ICA) for face recognition”. In: *International conference on audio and video based biometric person authentication*. Citeseer. 1999.

[106] Guangcan Liu et al. “Robust recovery of subspace structures by low-rank representation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 171–184.

[107] Qingshan Liu et al. “Face recognition using kernel-based fisher discriminant analysis”. In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE. 2002, pp. 197–201.

[108] Carl Livadas et al. “Usilng machine learning technliques to identify botnet traffic”. In: *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. IEEE. 2006, pp. 967–974.

[109] Frédéric Majorczyk et al. “Anomaly detection with diagnosis in diversified systems using information flow graphs”. In: *IFIP International Information Security Conference*. Springer. 2008, pp. 301–315.

[110] Frédéric Majorczyk et al. “Détection d’intrusions et diagnostic d’anomalies dans un systéme diversifié par comparaison de graphes de flux d’informations”. In: *Proceedings of the 6th Conference on Security and Network Architectures (SARSSI), Annecy, France*. 2007.

[111] S Manocha and Mark A Girolami. “An empirical analysis of the probabilistic k-nearest neighbour classifier”. In: *Pattern Recognition Letters* 28.13 (2007), pp. 1818–1824.

[112] Carla Marceau. “Characterizing the behavior of a program using multiple-length n-grams”. In: *Proceedings of the 2000 workshop on New security paradigms*. ACM. 2001, pp. 101–110.

[113] Tommaso Martiriggiano et al. “Facial feature extraction by kernel independent component analysis”. In: *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*. IEEE. 2005, pp. 270–275.

[114] John McHugh. “Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory”. In: *ACM Transactions on Information and System Security (TISSEC)* 3.4 (2000), pp. 262–294.

[115] Ludovic Mé and Véronique Alanou. “Détection d’intrusion dans un système informatique: méthodes et outils”. In: *TSI. Technique et science informatiques* 15.4 (1996), pp. 429–450.

[116] Peter Mell et al. *An overview of issues in testing intrusion detection systems*. 2003.

[117] Christoph Michael and Anup Ghosh. “Using finite automata to mine execution data for intrusion detection: A preliminary report”. In: *Recent Advances in Intrusion Detection*. Springer. 2000, pp. 66–79.

[118] Sebastian Mika et al. “Fisher discriminant analysis with kernels”. In: *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.* IEEE. 1999, pp. 41–48.

[119] Jelena Mirkovic. “D-WARD: source-end defense against distributed denial-of-service attacks”. PhD thesis. University of California, Los Angeles, 2003.

[120] Baback Moghaddam. “Principal manifolds and probabilistic subspaces for visual recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.6 (2002), pp. 780–788.

[121] Baback Moghaddam and Alex Pentland. “Probabilistic visual learning for object representation”. In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 696–710.

[122] James Newsome and Dawn Song. “Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software”. In: (2005).

[123] Feiping Nie, Jianjun Yuan, and Heng Huang. “Optimal mean robust principal component analysis”. In: *International Conference on Machine Learning*. 2014, pp. 1062–1070.

[124] Caleb C Noble and Diane J Cook. “Graph-based anomaly detection”. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2003, pp. 631–636.

[125] Julien Olivain and Jean Goubault-Larrecq. “The Orchids intrusion detection tool”. In: *Computer Aided Verification*. Springer. 2005, pp. 225–231.

[126] Francesco Palmieri, Ugo Fiore, and Aniello Castiglione. “A distributed approach to network anomaly detection based on independent component analysis”. In: *Concurrency and Computation: Practice and Experience* 26.5 (2014), pp. 1113–1129.

[127] Cludia Pascoal et al. “Detection of outliers using robust principal component analysis: A simulation study”. In: *Combining Soft Computing and Statistical Methods in Data Analysis* 77 (2010), pp. 499–507.

[128] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[129] John Pescatore and Greg Young. “Defining the next-generation firewall”. In: *Gartner RAS Core Research Note* (2009).

[130] Fabien Pouget, Marc Dacier, and Hervé Debar. “White paper: honeypot, honeynet, honeytoken: terminological issues”. In: *Rapport technique EURECOM* 1275 (2003).

[131] Jean-Philippe Pouzol and Mireille Ducassé. “From declarative signatures to misuse IDS”. In: *Recent Advances in Intrusion Detection*. Springer. 2001, pp. 1–21.

[132] Jean-Philippe Pouzol and Mireille Ducassé. “Handling generic intrusion signatures is not trivial”. In: *In Recent Advances in Intrusion Detection (RAID) Workshop*. Citeseer. 2000.

[133] Thomas H Ptacek and Timothy N Newsham. *Insertion, evasion, and denial of service: Eluding network intrusion detection*. Tech. rep. SECURE NETWORKS INC CALGARY ALBERTA, 1998.

[134] Rain Forest Puppy. *A look at whisker’s anti-IDS tactics*. 1999.

[135] Ronald L Rivest, Adi Shamir, and Leonard Adleman. “A method for obtaining digital signatures and public-key cryptosystems”. In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[136] Charissa Ann Ronao and Sung-Bae Cho. “Anomalous query access detection in RBAC-administered databases with random forest and PCA”. In: *Information Sciences* 369 (2016), pp. 238–250.

[137] Sam T Roweis and Lawrence K Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *science* 290.5500 (2000), pp. 2323–2326.

[138] Alaoui-Adib Saad, Chougdali Khalid, and Jedra Mohamed. “Network intrusion detection system based on Direct LDA”. In: *Complex Systems (WCCS), 2015 Third World Conference on*. IEEE. 2015, pp. 1–6.

[139] S Rasoul Safavian and David Landgrebe. “A survey of decision tree classifier methodology”. In: *IEEE transactions on systems, man, and cybernetics* 21.3 (1991), pp. 660–674.

[140] John W Sammon. “A nonlinear mapping for data structure analysis”. In: *IEEE Transactions on computers* 100.5 (1969), pp. 401–409.

[141] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Kernel principal component analysis”. In: *International Conference on Artificial Neural Networks*. Springer. 1997, pp. 583–588.

[142] Neda Afzali Seresht and Reza Azmi. “MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach”. In: *Engineering Applications of Artificial Intelligence* 35 (2014), pp. 286–298.

[143] Shahaboddin Shamshirband et al. “An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique”. In: *Engineering Applications of Artificial Intelligence* 26.9 (2013), pp. 2105–2127.

[144] Mei-Ling Shyu et al. *A novel anomaly detection scheme based on principal component classifier*. Tech. rep. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL and COMPUTER ENGINEERING, 2003.

[145] Stevan Stević. “Geometric mean”. In: *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 608–609.

[146] Salvatore J Stolfo et al. “Cost-based modeling for fraud and intrusion detection: Results from the JAM project”. In: *DARPA Information Survivability Conference and Exposition, 2000. DISCEX’00. Proceedings*. Vol. 2. IEEE. 2000, pp. 130–144.

[147] Clifford Stoll. “Stalking the wily hacker”. In: *Communications of the ACM* 31.5 (1988), pp. 484–497.

[148] Subashini Subashini and Veeraruna Kavitha. “A survey on security issues in service delivery models of cloud computing”. In: *Journal of network and computer applications* 34.1 (2011), pp. 1–11.

[149] Steve Suehring and Robert Ziegler. *Linux Firewalls (Novell Press)*. Novell Press, 2005.

[150] Masashi Sugiyama. "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis". In: *Journal of machine learning research* 8.May (2007), pp. 1027–1061.

[151] Masashi Sugiyama et al. "Semi-supervised local Fisher discriminant analysis for dimensionality reduction". In: *Machine learning* 78.1 (2010), pp. 35–61.

[152] Quan-Sen Sun et al. "A new method of feature fusion and its application in image recognition". In: *Pattern Recognition* 38.12 (2005), pp. 2437–2448.

[153] Quan-Sen Sun et al. "Improvements on CCA model with application to face recognition". In: *International Conference on Intelligent Information Processing*. Springer. 2004, pp. 125–134.

[154] Mahbod Tavallaee et al. "A detailed analysis of the KDD CUP 99 data set". In: *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. IEEE. 2009, pp. 1–6.

[155] Joshua B Tenenbaum, Vin De Silva, and John C Langford. "A global geometric framework for nonlinear dimensionality reduction". In: *science* 290.5500 (2000), pp. 2319–2323.

[156] Fernando De la Torre and Michael J Black. "Robust principal component analysis for computer vision". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 1. IEEE. 2001, pp. 362–369.

[157] Fernando De la Torre et al. "Representational oriented component analysis (ROCA) for face recognition with one sample image per training class". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 266–273.

[158] Eric Totel, Bernard Vivinis, and Ludovic Mé. "A language driven intrusion detection system for event and alert correlation". In: *Security and Protection in Information Processing Systems* (2004), pp. 208–224.

[159] Madeleine Udell et al. "Generalized low rank models". In: *Foundations and Trends® in Machine Learning* 9.1 (2016), pp. 1–118.

[160] Hank S Vaccaro and Gunar E Liepins. "Detection of anomalous computer session activity". In: *Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on*. IEEE. 1989, pp. 280–289.

[161] Vladimir Vapnik, Steven E Golowich, and Alex J Smola. "Support vector method for function approximation, regression estimation and signal processing". In: *Advances in neural information processing systems*. 1997, pp. 281–287.

[162] Giovanni Vigna, Steven T Eckmann, and Richard A Kemmerer. "Attack languages". In: *Proceedings of the IEEE Information Survivability Workshop*. Vol. 366. 2000.

[163] Haixian Wang et al. “Fisher discriminant analysis with L1-norm”. In: *IEEE transactions on cybernetics* 44.6 (2014), pp. 828–842.

[164] Haixian Wang et al. “Locality-preserved maximum information projection”. In: *IEEE Transactions on Neural Networks* 19.4 (2008), pp. 571–585.

[165] Jie Wang, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. “Selecting discriminant eigenfaces for face recognition”. In: *Pattern Recognition Letters* 26.10 (2005), pp. 1470–1482.

[166] Wei Wang and Roberto Battiti. “Identifying intrusions in computer networks with principal component analysis”. In: *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*. IEEE. 2006, 8–pp.

[167] Wei Wang, Xiaohong Guan, and Xiangliang Zhang. “A novel intrusion detection method based on principle component analysis in computer security”. In: *Advances in Neural Networks-ISNN 2004* (2004), pp. 88–89.

[168] Wei Wang, Xiaohong Guan, and Xiangliang Zhang. “Profiling program and user behaviors for anomaly intrusion detection based on non-negative matrix factorization”. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 1. IEEE. 2004, pp. 99–104.

[169] Christina Warrender, Stephanie Forrest, and Barak Pearlmutter. “Detecting intrusions using system calls: Alternative data models”. In: *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*. IEEE. 1999, pp. 133–145.

[170] Andreas Wespi, Marc Dacier, and Hervé Debar. “Intrusion detection using variable-length audit trail patterns”. In: *Recent advances in intrusion detection*. Springer. 2000, pp. 110–129.

[171] Yi Wu, David Wipf, and Jeong-Min Yun. “Understanding and evaluating sparse linear discriminant analysis”. In: *Artificial Intelligence and Statistics*. 2015, pp. 1070–1078.

[172] Huan Xu, Constantine Caramanis, and Shie Mannor. “Outlier-robust PCA: the high-dimensional case”. In: *IEEE transactions on information theory* 59.1 (2013), pp. 546–572.

[173] Makoto Yamada, Ali Pezeshki, and Mahmood R Azimi-Sadjadi. “Relation between kernel CCA and kernel FDA”. In: *Neural Networks, 2005. IJCNN’05. Proceedings. 2005 IEEE International Joint Conference on*. Vol. 1. IEEE. 2005, pp. 226–231.

[174] Dayu Yang and Hairong Qi. “A network intrusion detection method using independent component analysis”. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE. 2008, pp. 1–4.

[175] Jian Yang, David Zhang, and Jing-yu Yang. “Median LDA: a robust feature extraction method for face recognition”. In: *Systems, Man and Cybernetics, 2006. SMC’06. IEEE International Conference on*. Vol. 5. IEEE. 2006, pp. 4208–4213.

[176] Jieping Ye and Qi Li. “A two-stage linear discriminant analysis via QR-decomposition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.6 (2005), pp. 929–941.

[177] Hua Yu and Jie Yang. “A direct LDA algorithm for high-dimensional data—with application to face recognition”. In: *Pattern recognition* 34.10 (2001), pp. 2067–2070.

[178] Pong C Yuen and Jian-Huang Lai. “Face representation using independent component analysis”. In: *Pattern recognition* 35.6 (2002), pp. 1247–1257.

[179] Lotfi A Zadeh. “Information and control”. In: *Fuzzy sets* 8.3 (1965), pp. 338–353.

[180] Naser Zaeri. “Discriminant phase component for face recognition”. In: *Journal of Electrical and Computer Engineering* 2012 (2012), p. 10.

[181] Shuai Zheng et al. “A Harmonic Mean Linear Discriminant Analysis for Robust Image Classification”. In: *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*. IEEE. 2016, pp. 402–409.

[182] Wenming Zheng, Zhouchen Lin, and Haixian Wang. “L1-norm kernel discriminant analysis via Bayes error bound optimization for robust feature extraction”. In: *IEEE transactions on neural networks and learning systems* 25.4 (2014), pp. 793–805.

[183] Fujin Zhong and Jiashu Zhang. “Linear discriminant analysis based on L1-norm maximization”. In: *IEEE Transactions on Image Processing* 22.8 (2013), pp. 3018–3027.

[184] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. “A survey of coordinated attacks and collaborative intrusion detection”. In: *Computers & Security* 29.1 (2010), pp. 124–140.

[185] Elkhadir Zyad, Chougdali Khalid, and Benattou Mohammed. “Combination of R1-PCA and median LDA for anomaly network detection”. In: *2017 Intelligent Systems and Computer Vision (ISCV)*. IEEE. 2017, pp. 1–5.