CHAPITRE 4

IMPLEMENTATION

Introduction :

Dans ce chapitre nous présentons une architecture de CAH nommé ***IDSCAH « Système de D****étection d'Intrusions à base CAH »* qui nous permet de classifier des connexions TCP/IP de la base KDD'99 dans le but de construire un système de détection d'intrusions selon l'approche par signature. L'implémentation de ce système sera ensuite détaillée et suivie par les résultats obtenus.

1. : [6]

La base KDD est une base de données orientée détection d'intrusions. Elle présente des lignes TCP/IP où chaque ligne est une connexion caractérisée par 41 attributs, telle que la durée de la connexion, le type du protocole, etc. En tenant compte des valeurs de ses attributs, chaque connexion dans KDD est considérée comme étant une connexion normale ou bien une attaque.

Les données KDD sont en fait des données formatées fournies par DARPA. Ces données présentent 7 semaines de données libellées pour l'apprentissage et 2 semaines de données non libellées pour le test (Correspondant au trafic réseau simulant un réseau local d'US Air force). Il existe des travaux récents utilisant ces données (voir chapitre 2).

Cette base de données contient 24 types d'attaques qui peuvent être classés comme quatre types d'être désignés comme attaque par déni de service, Probe, remote-to-local (R2L), et user-to-root (U2R).

2. Outils utilisés :

Pour réaliser notre système de détection d'intrusion, nous avons utilisé des outils, Netbeans pour créer des classes Java, WinPCap et Wireshark deux outils coopératives permettant de capturer les paquets du réseau et MySQL pour créer et mettre à jour une base de données d'apprentissage.

2.1. NetBeans : [39]

L'IDE NetBeans est un outil rapide et complet pour développer des logiciels Java. Il respecte les standards et s'exécute sur n'importe quel système d'exploitation où une Machine Virtuelle Java est disponible. L'IDE NetBeans est open-source. Il peut être utilisé libre de coût pour n'importe quel type de développement logiciel. L'IDE NetBeans s'exécute sur n'importe quelle plateforme. Bien que l'IDE soit modulaire (ses fonctionnalités peuvent être étendues par des plug-ins), le principal objectif de netbeans.org est de fournir tous les outils nécessaire au développeur Java en un seul téléchargement, sans nécessiter de mise à jour ou de configuration supplémentaire pour commencer un travail productif avec l'IDE.

Le produit de base inclus le support pour le développement bureautique (AWT/Swing), le web (Servlets/JSP/JSF/Struts) et l'Edition Entreprise de Java (EJB et Services web), et inclus une base de données et un serveur d'application Java EE, aussi aucun téléchargement supplémentaire n'est nécessaire pour commencer à être productif.

2.1.1. JAVA : [40] [41]

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable.

Il permet aux programmeurs d'écrire des instructions d'ordinateur à l'aide des commandes en anglais, au lieu d'avoir à écrire dans des codes numériques. Elle est connue comme une langue «de haut niveau», car il peut être lu et écrit facilement par les humains. Comme l'anglais, Java a un ensemble de règles qui déterminent la façon dont les instructions sont écrites. Ces règles sont connues sous le nom de sa «syntaxe». Une fois qu'un programme a été écrit, les instructions de haut niveau sont converties en codes numériques que les ordinateurs peuvent comprendre et exécuter.

2.2. Winpcap : [42]

WinPcap est une bibliothèque open source pour la capture de paquets et l'analyse de réseau pour les plates-formes Win32. Il est facile d'accéder à des données sur le réseau avec cette approche depuis le système d'exploitation fait face avec les détails de bas niveau (traitement de protocole, le réassemblage de paquets, etc.) et fournit une interface familière qui est similaire à celui utilisé pour lire et écrire des fichiers. Parfois, cependant, ils ont besoin d'accéder aux données "brutes" sur le réseau sans interposition d'un traitement de protocole par le système d'exploitation.

2.3. Wireshark : [44]

Est un analyseur de paquets libre utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro-ingénierie. Son appellation d'origine (Ethereal) est modifiée en mai 2006 pour des questions relatives au droit des marques.

Wireshark utilise la bibliothèque logicielle GTK+ pour l'implémentation de son interface utilisateur et pcap pour la capture des paquets, il fonctionne sur de nombreux environnements compatibles UNIX ou Mac OSX, mais également sur Microsoft Windows. Aujourd'hui, Wireshark reconnaît 1 515 protocoles.

2.4. WampServer : [43]

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

3. Description de notre système :

La mise en place de notre système (***IDSCAH***) doit être dans un environnement d'un réseau au niveau d'un poste puisqu'il est basé sur les paquets circulant dans ce réseau.

Le but de notre système est de détecter toutes tentatives d'attaque (intrusion) dans un réseau local, donc il fait une classification de donnés de la base de données kddcup99 par l'algorithme CAH, et grâce à cette classification on peut détecter les intrusions par mesure de distance, donc, il calcule une distance entre chaque connexion entrée et chaque cluster préparé par notre classification CAH.

Il suffit de lancer une capture par un sniffer (Wireshark), et avec la commande TSHARK, on extrait toutes les informations requises de cette capture pour créer les connexions réseaux puis on les exporte dans une table d'une base de données Mysql pour faciliter leurs manipulations dans un environnement java.

Notre IDS ne détecte que les intrusions qui sont dans la base de données kdd99, donc il faut toujours faire la mise à jour. Ainsi qu'il prend de temps pour donner les résultats à cause de grande quantité de données manipulées.

4. Résultats :

Nous rapportons dans ce qui suit les résultats des expérimentations de CAH sur la base 10% de KDD'99 avec notre capture réseau qui consiste en un échantillon de 300 connexions. Ici, l'IDS à base de CAH est construit et testé sur les mêmes données en vue d'évaluer à quel point le système construit est-il cohérent avec les données de cette capture. Les figures suivantes montrent les dendrogrammes du système qui permet de visualiser les résultats de nôtres expérimentations.

4.1.Expérimentation du modèle sur un échantillon de 300 connexions en utilisant les trois critères d'agrégations:

L'expérimentation donne les résultats sous forme d'un dendrogramme regroupe les connexions dans des clusters forment une hiérarchie, les figures 1, 2 et 3 représentent respectivement l'utilisation de critères de saut minimal, le critère moyenne, saut maximal.

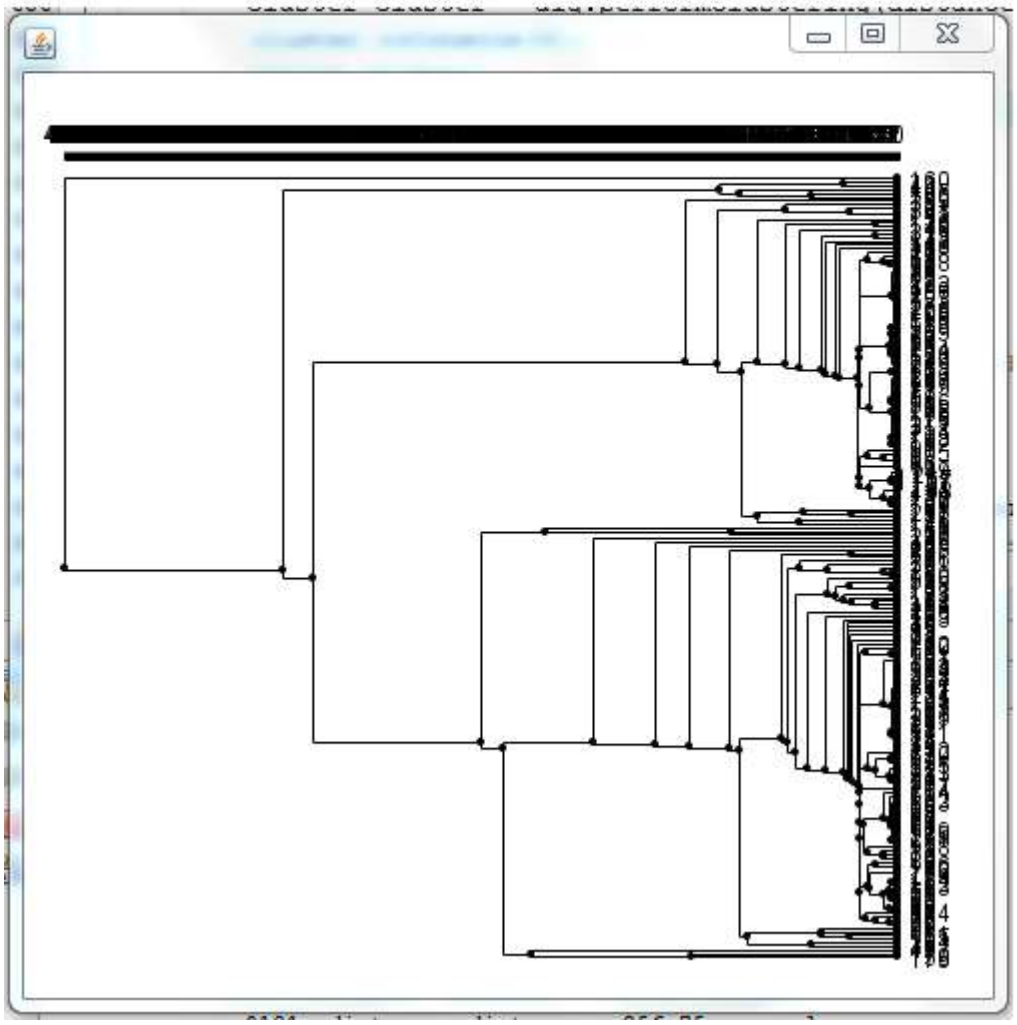

Figure 4.1 : Dendrogramme avec le critère d'agrégation «le saut minimal »

Dans cette fenêtre, nous avons affiché les résultats sous forme d'un dendrogramme de la classification de données de capture en utilisant le critère de saut minimal.

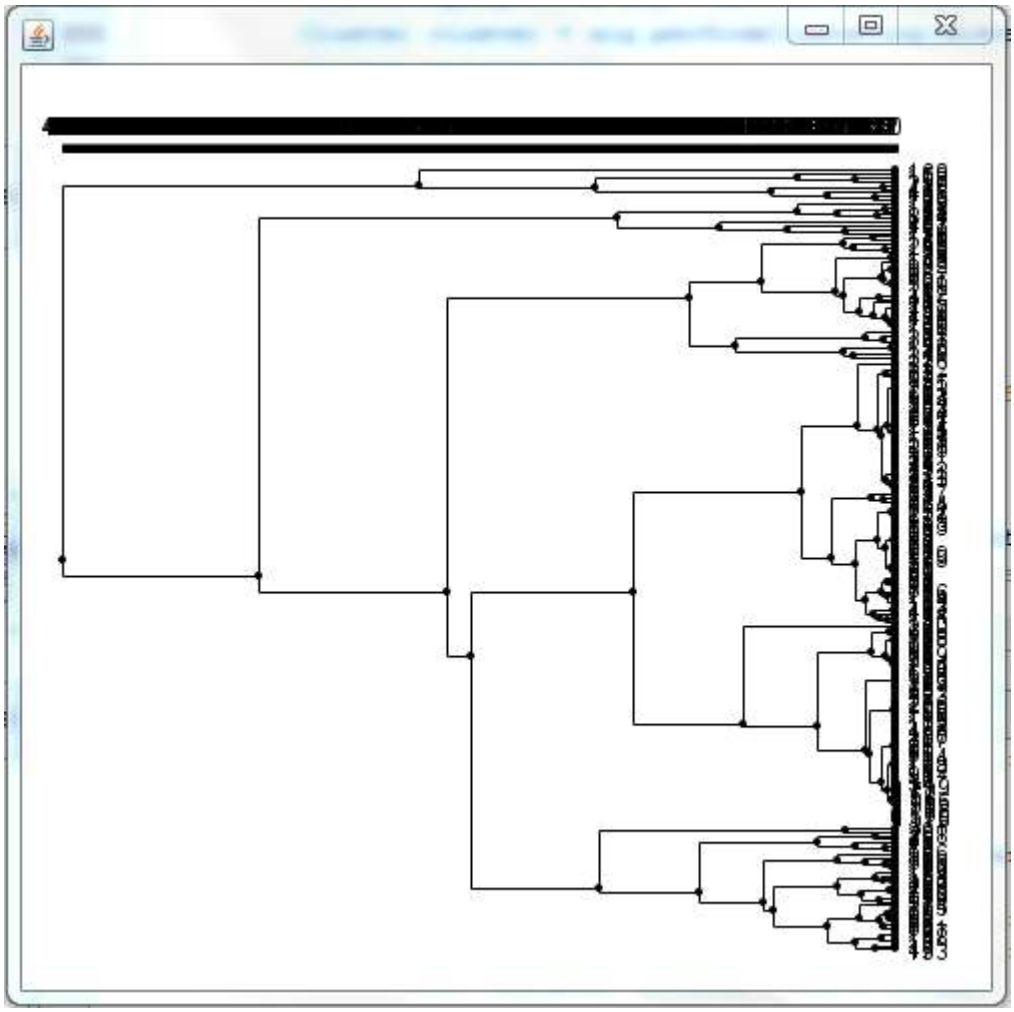

Figure 4.2 : Dendrogramme avec Le critère de la moyenne

Dans cette fenêtre, nous avons affiché les résultats sous forme d'un dendrogramme de la classification de données de capture en utilisant le critère de la moyenne.

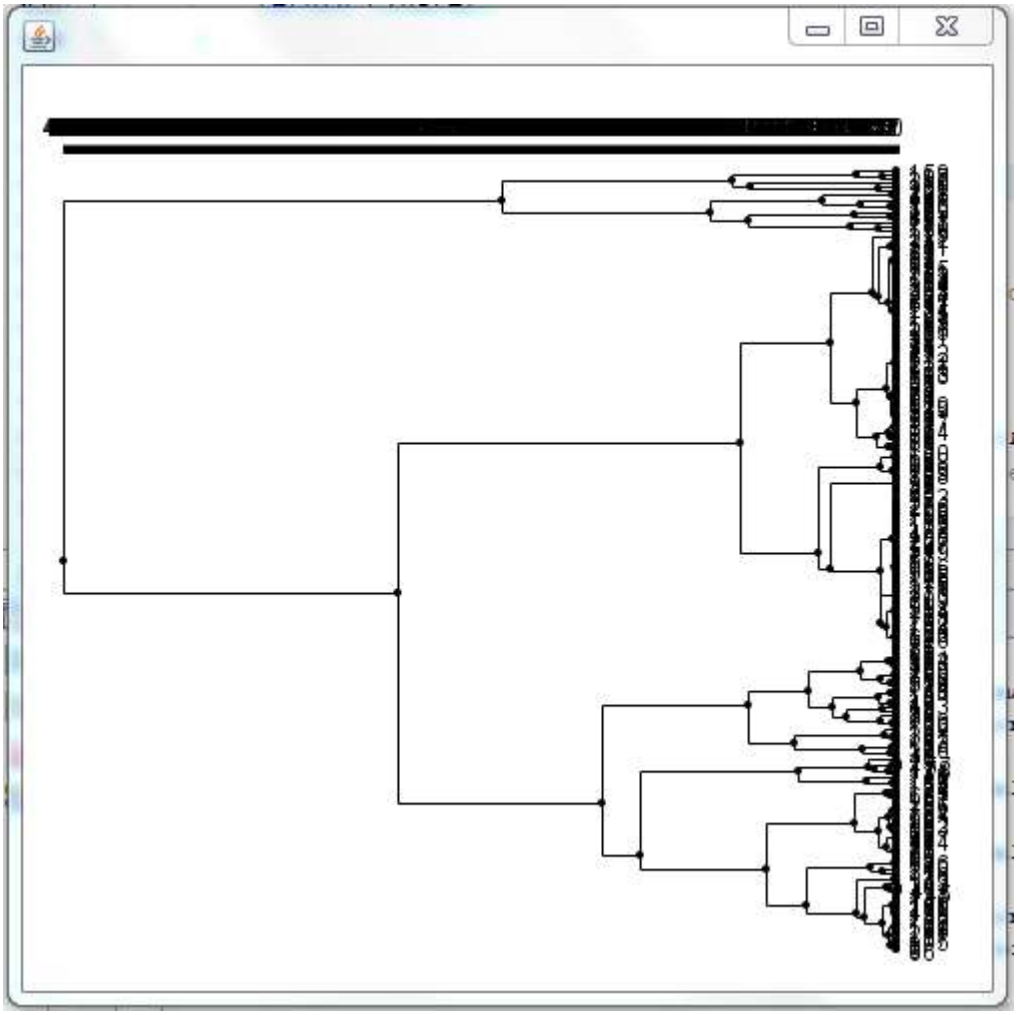

Figure 4.3 : Dendrogramme avec le critère d'agrégation «le saut maximal »

Dans cette fenêtre, nous avons affiché les résultats sous forme d'un dendrogramme de la classification de données de capture en utilisant le critère de saut maximal.

4.2. Mesure des performances

Les différents paramètres utilisés dans le but de vérifier les performances et l'observation expérimentale des trois scénarios de CAH décrits ci-dessus, sont :

- **Taux de détection (DR):** il est défini comme le nombre de cas d'intrusions détectées par le système (Vrai positif) divisé par le nombre total de cas d'intrusions présentes dans l'ensemble de données.

$$DR = \frac{TP}{le\ nombre\ total\ de\ cas\ d'intrusions\ dans\ l'ensemble\ de\ données.}$$

- **Vrai positif (TP)** : correspond à une action légitime qui déclenche une IDS pour produire une alarme.

- **Vrai Négatif (TN)** : Un événement où aucune attaque n'a eu lieu et aucune détection n'est faite.
- **Taux de faux positifs (FP)** est : $FP = \frac{FP}{FP+TN}$

Le tableau 4.1 ci-dessous récapitule les résultats de classification d'un test de capture pour les trois scénarios :

techniques de clustering	Taux de détection	Faux positif
le saut minimal	63.38%	36.61%
Le critère de la moyenne	64.4%	35.6%
le saut maximal	64.5%	35.3%

Tableau4.1 : Le résultat d'algorithme en utilisant des techniques de clustering

Ces résultats montrent clairement que les connexions normales sont bien classifiées par le système avec un taux de détection de l'ordre de 63% lorsque le système a appris plus de 300 connexions.

De plus, il est clair que les résultats des trois scénarios sont proches les uns des autres avec une légère performance du saut maximal dû à l'application des distances maximales dans le clusturing.

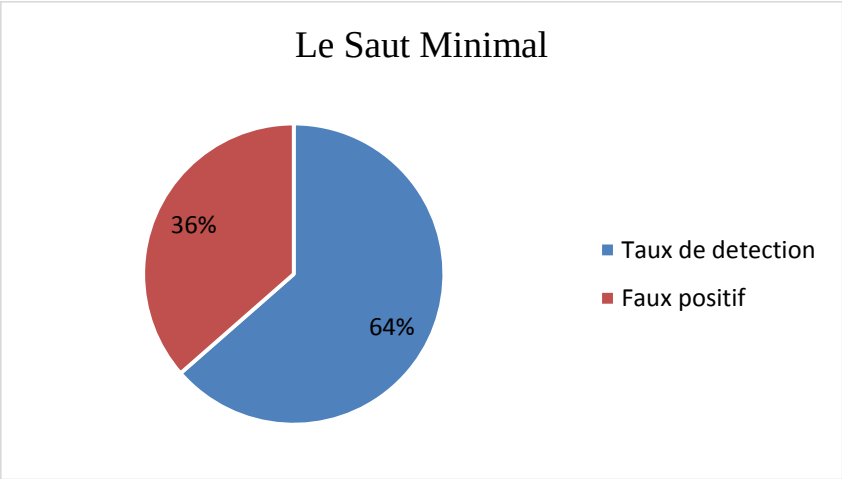

Figure4.4 : représentation graphique de classification dans le cas de saut minimal.

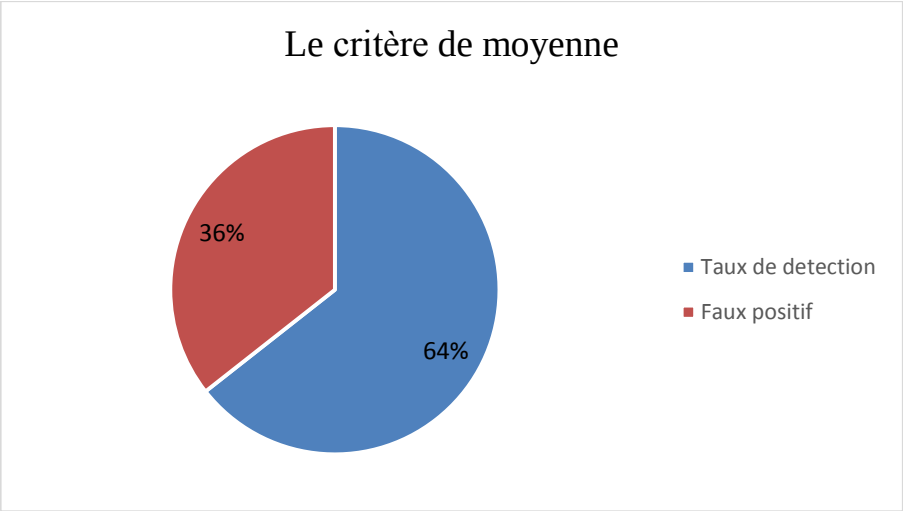

Figure4.5 : représentation graphique de classification dans le cas de Le critère de moyenne.

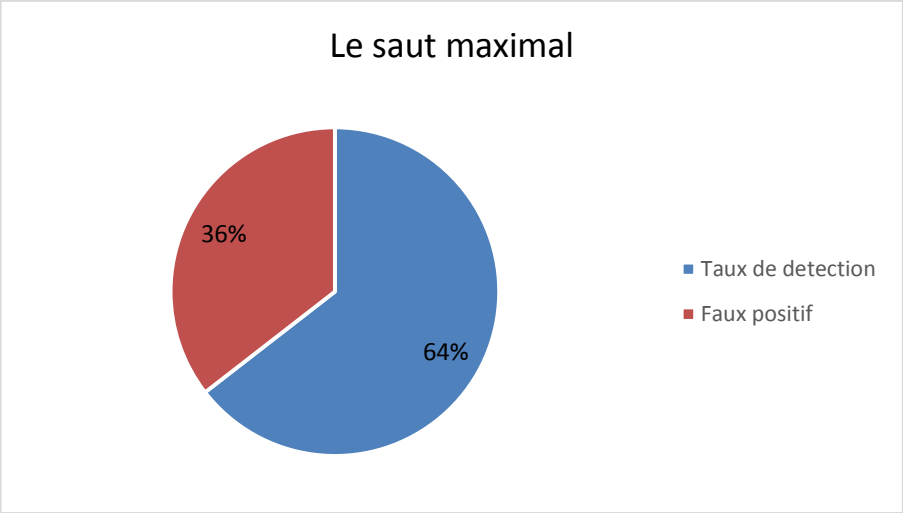

Figure4.6 : représentation graphique de classification dans le cas de saut maximal.

4.3.Validation :

L'évaluation du système se fait en introduisant au système les données de test qui sont stockées dans la base de test et qui comporte 311029 connexions parmi lesquelles 60593 sont des connexions normales et les autres sont des attaques, on choisit un échantillon de 200, 300 et 500 connexions. Les résultats obtenus sont représentés dans le tableau4.2 suivant :

Nom de fichier	Taux de détection	Faux positif
Moyenne	71.18%	23.81%

Tableau4.2 : Résultats des données de test (Base d'apprentissage de 200,300 et 500 connexions)

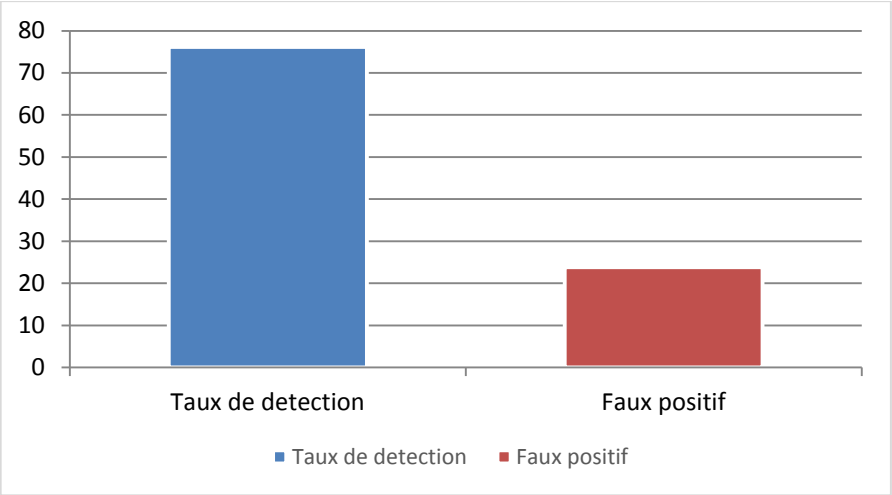

Figure4.7 : représentation graphique de moyenne de DR et PF.

5. Discussion :

L'apprentissage explique le taux de détection moyen généré par le système, tel que, par le test de la capture qui comporte 300 connexions normales le système à trouver un taux correspond à 63.38% avec la technique de saut minimal, 64.4% avec le critère de moyenne et 64.5% avec la technique de saut maximal. Ce principe a permis au système de classer correctement certaines connexions normales de la base de test mais il a généré un taux remarquable de faux positifs, à cause de la phase de l'extraction des champs de la capture, dans cette dernière, nous avons fait l'extraction de sept champs pour chaque connexion ce qui a conduit à des résultats peu faibles. Par contre nous avons trouvé des meilleurs résultats avec les données de fichier de test à cause de la cohérence de ces données.

Conclusion :

Au terme de ce dernier chapitre, nous avons implémenté un système de détection d'intrusions IDS dont le but est de sécuriser notre réseau local. Cette implémentation offre, via un ensemble de programme et de bases de données la possibilité de détecter les attaques.

Les motivations de ce travail sont la construction d'un IDS détecte les utilisations malveillante du système et aussi génère le minimum de fausses alertes.