Devoir 1

Réalisé par Tristan Lamblot et Yoann Switala

Présenté à Abdenour Bouzouane
Dans le cadre du cours
8INF954 : Forage de données

UQAC
Université du Québec
à Chicoutimi

15 octobre 2016

1 Synthèse

This article introduces the different ways to improve network security towards intrusions of hackers. In order to protect the network from an intrusion, a function has been invented : IDS (Intrusion Detection System). This system is using data mining and probes to find unusual activity or user on the network. It also highlight the issue of FP (False Positives) that detects threats which are not.

It describes the different algorithms used to detect the threats : Decision Tree (DT), ID3 and ID3 with Havrda and Charvat Entropy.

In order to determine which one has the most efficient detection rate and the less FP rate, it has been tested with the KDD-99 dataset (containing single connection vectors, each with a different type of attack). After different situation tests, the article proves that the ID3 algorithm using Havrda and Charvat Entropy instead of Shannon's is more accurate in detection and makes less mistakes creating FPs for 5 types of attacks (Normal, Denial of Service, Remote to User, User to Root and Probing).

2 Modification Id3

L'algorithme Id3 modifié se situe dans l'arborescence de dossier qui constitue weka. Il se situe dans le package "tree". Son code est lisible dans le fichier "Id3Modified.java" qui se situe dans weka/src/-main/java/weka/classifiers/trees/. Les sources de l'algorithme proviennent du package Simple Learning Machine.

Pour compiler weka : il suffit de se positionner dans le dossier weka et d'executer la commande "ant exejar" puis de lancer weka.jar qui se situe dans le dossier weka/dist/.

De plus dans l'interface pour les options, dans la catégorie classifiers, la gestion du paramètre alpha a été rajoutée pour faciliter celle-ci comme indiqué comme bonus de ce devoir.

3 Expérimentation

Dans le cadre de l'expérimentation un échantillonneur a été écrit. Il permet de faire une échantillonnage avec remise. Il fonctionne avec tout fichier de type .arff (pas seulement pour celui de ce devoir). Ce programme se trouve dans le dossier Echantillonneur. Pour le lancer voir le fichier "commande.txt".

La taille de l'échantillon utilisé correspond au protocole décrit dans le papier. Il correspond à 10 % de la base de données totale, soit 49 500 entrées.

L'algorithme a été testé avec différentes valeurs de alpha. Les resultats pour ces différentes valeurs sont dans le dossier "Result". La plage de test pour alpha était [0.1, 0.5, 0.9, 1,5, 2, 2,5, 5, 10]. On remarque un seuil pour alpha. En effet, pour alpha < 1 on a un taux de succès de 93.0263 % alors que pour des valeurs de alpha > 1 on a un taux de succès de 97.0162 %.

On en déduit que la valeur optimale de alpha est de 1,1. Cette valeur de alpha fournit bien un taux de succès de 97.0162 %. En comparant ces résultats avec les données du papier on remarque une légére différence de 1 %. On suppose que cette différence vient de l'échantillonnage.

4 Comparaison des résultats

On remarque que les résultats avec la nouvelle entropie sont légérements supérieurs à ceux de l'algorithme avec l'ancienne entropie. De plus, en passant l'échantillon dans l'algorithme J48 de Weka ont obtient un résultat de 99.4831 %. un taux de succès supérieur à ceux annoncé par le papier.

5 Conclusion

On remarquera que la valeur optimale de alpha est de 1,1. Les résultats obtenus sont comparés entre les résultats de Id3, Id3Modified et J48. Cette comparaison met en évidence le rôle important de l'implémentation de l'algorithme ainsi que de la qualité de l'échantillon.