UQAC

Université du Québec
à Chicoutimi

Devoir #1

Intrusion Detection and Classification Using Improved ID3 Algorithm of Data Mining

Alexis Fougerouse
2018 - 2019

Table des matières

1. Synthèse de l'article

Dans cette article, l'auteur nous donne les clés afin de modifier ID3 en changeant le calcul de l'entropie. Sur ID3 cette entropie est calculée suivant l'entropie de Shannon, ici c'est l'entropie de Havrada-Charvat, une généralisation de celle de Shannon, qui est utilisée. La principale raison de ce choix est que l'entropie de Shannon donne lui à des arbres de décisions beaucoup trop complexes, ce qui rend le processus décisionnel long.

Afin de tester cette version ID3, les deux auteurs de l'article ont utilisé un dataset de la KDD CUP de 1999 contenant 4.9 million d'enregistrement simulant 22 types de cyberattaques. L'échantillon du dataset utilisé possède deux classes pour spécifier les données, attack, regroupant les classes Probe, DOS, U2R et R2L du dataset original, et normal. Chaque vecteur de ce dataset à 14 attributs.

Les résultats de l'expérimentation comparant ID3 et l'algorithme établi dans l'article montre que la modification d'ID3 donne un taux d'erreur de 2.81% avec une précision de 97.74% et un pourcentage de faux positif, notamment pour les attaques détectées comme étant des comportement normaux, significativement meilleurs que pour ID3.

2. Implémentation et échantillonnage

Afin d'implémenter la solution décrite dans l'article je me suis basé sur un algorithme ID3 codé en python trouvé sur internet (https://github.com/tofti/python-id3-trees). L'algorithme de base se contentait de calculer l'arbre de décision à partir de données d'entraînement et de l'afficher. J'ai en plus rajouté le parcours de l'arbre permettant de calculer les taux de succès avec les valeurs d'entraînement et des valeurs de test.

Un prétraitement a été effectué sur les données KDDCup 99. Tout d'abord, seulement 10% des données et les 14 premiers attributs sont utilisés comme décrit dans l'article. Ensuite, les données d'entraînement ont été labellisées uniquement par deux classes, "attack" et "normal". Quand aux données de test, elles sont labellisées selon 5 classes "normal", "Probing", "DoS", "U2R" et "R2L". Comme beaucoup d'attributs possèdent des valeurs continues, les données ont été coupées en classes car ID3 n'accepte que des valeurs nominales. On a donc deux fichiers, un pour l'entraînement et un pour le test, le découpage correspond à environ ⅔, ⅓ des 10% des données de la KDDCup 99.

3. Résultats

L'algorithme ID3 sans modification donne un taux de succès de 99.8237% et un taux de faux positif de 0.4433% sur les données de test.

Les résultats avec ID3 modifié suivant les valeurs de alpha sont les suivant :

- 0.1 : 99.8237% 0.4433%
- 0.5 : 99.8227% 0.4510%
- 0.8 : 99.8222% 0.4548%
- 1.1 : 99.8242% 0.4433%
- 1.5 : 99.8237% 0.4433%
- 2 : 99.8237% 0.4433%

- 5 : 99.8237% 0.4433%
- 10 : 99.8237% 0.4433%

Pour chacune des valeurs de alpha observées, le taux de faux positif est faible. Cela veut dire que très peu d'attaques ont été considérée comme une activité normale.

On peut remarquer que pour un alpha supérieur à 1, alpha est décroissant et tend vers une asymptote horizontale se dégage pour une valeur de 99.8237%. En dessous, on remarque que pour un alpha qui tend vers 1 le taux de succès décroît. On peut conclure que la valeur optimale de alpha est 1.1 car ça semble être la valeur qui renvoi le taux de succès le plus élevé.

4. Interprétation des résultats

Si on compare le nouvelle algorithme ID3 avec la version original, on remarque que suivant la valeur de alpha, l'entropie de Harvrda-Charvat donne de meilleur résultat. On explique ce résultat par le fait que l'entropie de Harvrda-Charvat est une généralisation de celle Shannon. De plus, le paramètre alpha permet d'ajuster l'algorithme pour qu'il produise un meilleur modèle en fonction des données d'entraînement.

Par rapport aux résultats qui font suites à l'article, on remarque que le taux de succès est supérieur de 2% environ. Cela peut s'expliquer dans un premier temps par des différences dans l'implémentation de l'algorithme, mais aussi par le prétraitement qui a été fait sur les données de l'échantillon. La nominalisation des données continues que j'ai effectué est sans doute différente de celle faite par les rédacteurs de l'article.

5. Annexes

Exemple d'exécution de l'algorithme ID3 modifié.

```
[panderium@imtheonewhoknocks python-id3-trees]$ python3 id3-modified.py resources/IDS-simplified.cfg
Command line args are ['id3-modified.py', 'resources/IDS-simplified.cfg']:
Valeur de alpha : 1.5
IF service EQUALS telnet AND flag EQUALS SF AND hot EQUALS 0 AND dst_bytes EQUALS
EQUALS 0to442 AND protocol_type EQUALS icmp THEN normal
IF service EQUALS other AND flag EQUALS REJ AND urgent EQUALS 0 AND land EQUALS 0
IF service EQUALS ftp_data AND flag EQUALS SF AND hot EQUALS 0 AND logged_in EQUA
 0to442 AND protocol_type EQUALS tcp AND num_failed_logins EQUALS 0 AND wrong_fra
IF service EQUALS finger AND flag EQUALS RSTO AND hot EQUALS 2 THEN attack
IF service EQUALS finger AND flag EQUALS SF AND hot EQUALS 0 AND urgent EQUALS 0
IF service EQUALS tim_i AND urgent EQUALS 0 AND land EQUALS 0 AND duration EQUALS
IF service EQUALS telnet AND flag EQUALS S0 AND hot EQUALS 0 AND urgent EQUALS 0
IF service EQUALS other AND flag EQUALS S0 THEN attack
IF service EQUALS finger AND flag EQUALS RSTO AND hot EQUALS 19 THEN attack
IF service EQUALS ftp_data AND flag EQUALS SF AND hot EQUALS 6 THEN normal
Le taux de success sur les données d'entrainement est de 99.48432989553264%.
Le taux de faux positif sur les données d'entrainement est de 0.0%.
Le taux de faux positif sur les données de test est de 0.4433136733356463%.
Le taux de success sur les données de test est de 99.82373223793056%.
```