Classification par arbre de decision

Devoir #1

MARDINI Alexandre | PAULIAT Thibaud

Forage de donnée

Table des matières

[I. Résumé 2](#_Toc22676832)

[II. Format de données 3](#_Toc22676833)

[II. Sélection des attributs 3](#_Toc22676834)

[a. Sélection par corrélation 3](#_Toc22676835)

[b. Sélection par gain 4](#_Toc22676836)

[III. Tests 6](#_Toc22676837)

## Résumé

Il nous est d’abord expliqué que les algorithmes de détections d’intrusions possèdent beaucoup de problèmes de performances, de bas niveau d’intelligence, de faux positifs… En revanche l’algorithme basé sur ID3 a taux de réussite de détection d’intrusion de plus de 97%.

Il faut savoir que la détection d’intrusion et le processus d’identification des tentatives de compromission de la confidentialité, de l’intégrité ou de la disponibilité des ordinateurs ou des réseaux. Le but de ces algorithmes est donc de détecter des actions non autorisées. Actuellement leur système de reconnaissance sont basés sur des reconnaissances de signature/modèle. Ils nécessitent donc d’être mis à jour à chaque découverte d’une nouvelle signature. Les algorithmes basés sur ID apportent une intelligence de reconnaissance afin de détecter des attaques inconnues.

Les algorithmes se basent sur des arbres de décisions. Ceux-ci sont construit à partir de sources de données et regroupent les données par classes. Le principe de l’algorithme ID3 est de récursivement catégoriser les informations d’une base de données afin de réduire au maximum le nombre de classes.

La DARPA et le laboratoire Lincoln ont créé une feuille de données d'entraînement qui a servi à faire des millions de tests et de déterminer 22 types d’attaques rassemblées en 4 groupes : DoS, R2L, U2R and Probe. Chaque connexion enregistrée est composée de 41 attributs.

Un exemple d’algorithme de détection basé sur ID3 nous est donné, et son taux de réussite est de 97,74% et son taux d’erreur de 2,81%. Ainsi nous pouvons voir l’efficacité de cet algorithme.

## II. Format de données

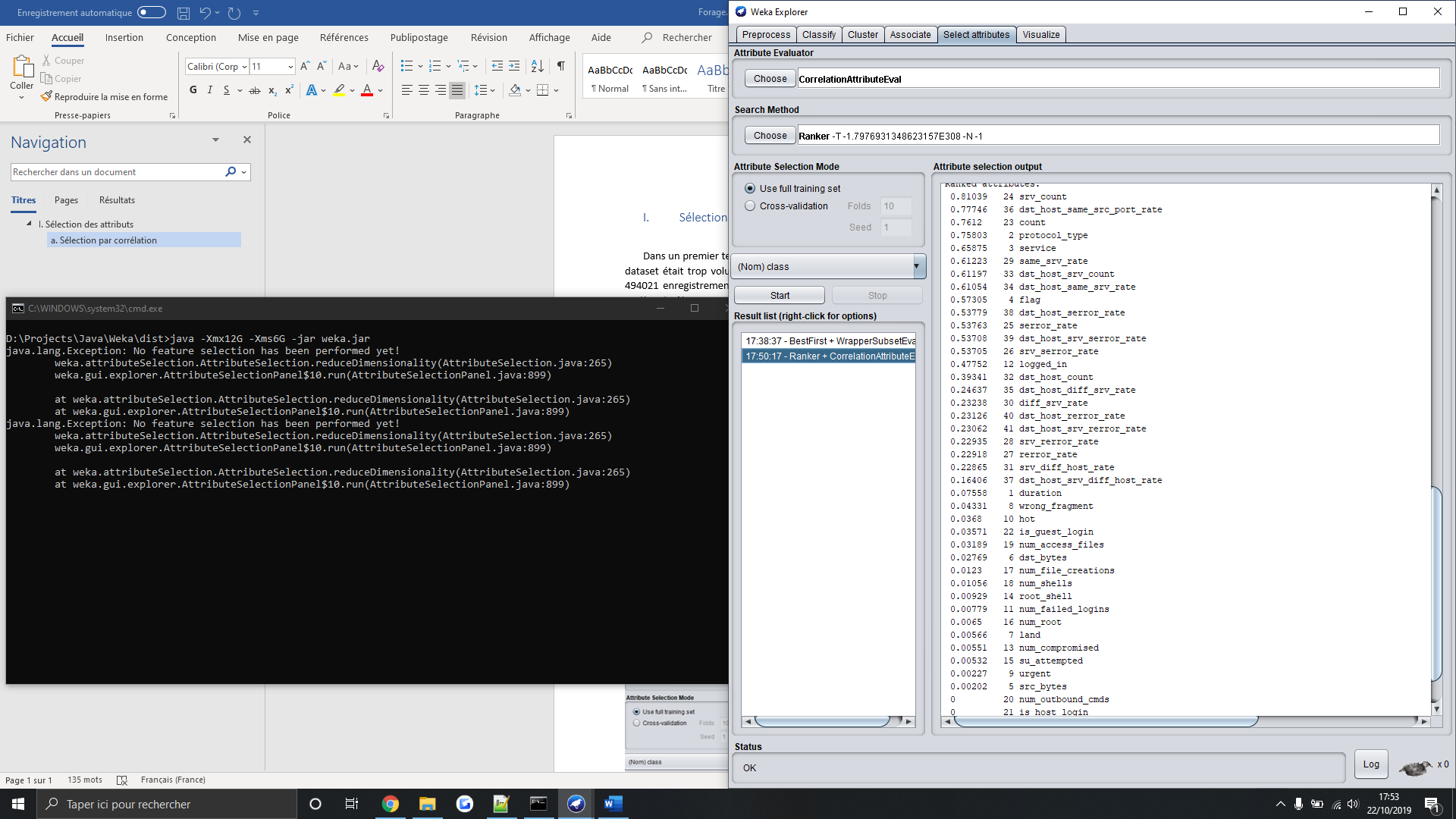
Nous avons utilisé 10% du dataset de KDDCup tout au long de nos tests. Nous avons dans un premier temps regroupé les différentes attaques en 4 Types (R2L, DoS, Probing et U2R) comme explicité dans l’article.

## Sélection des attributs

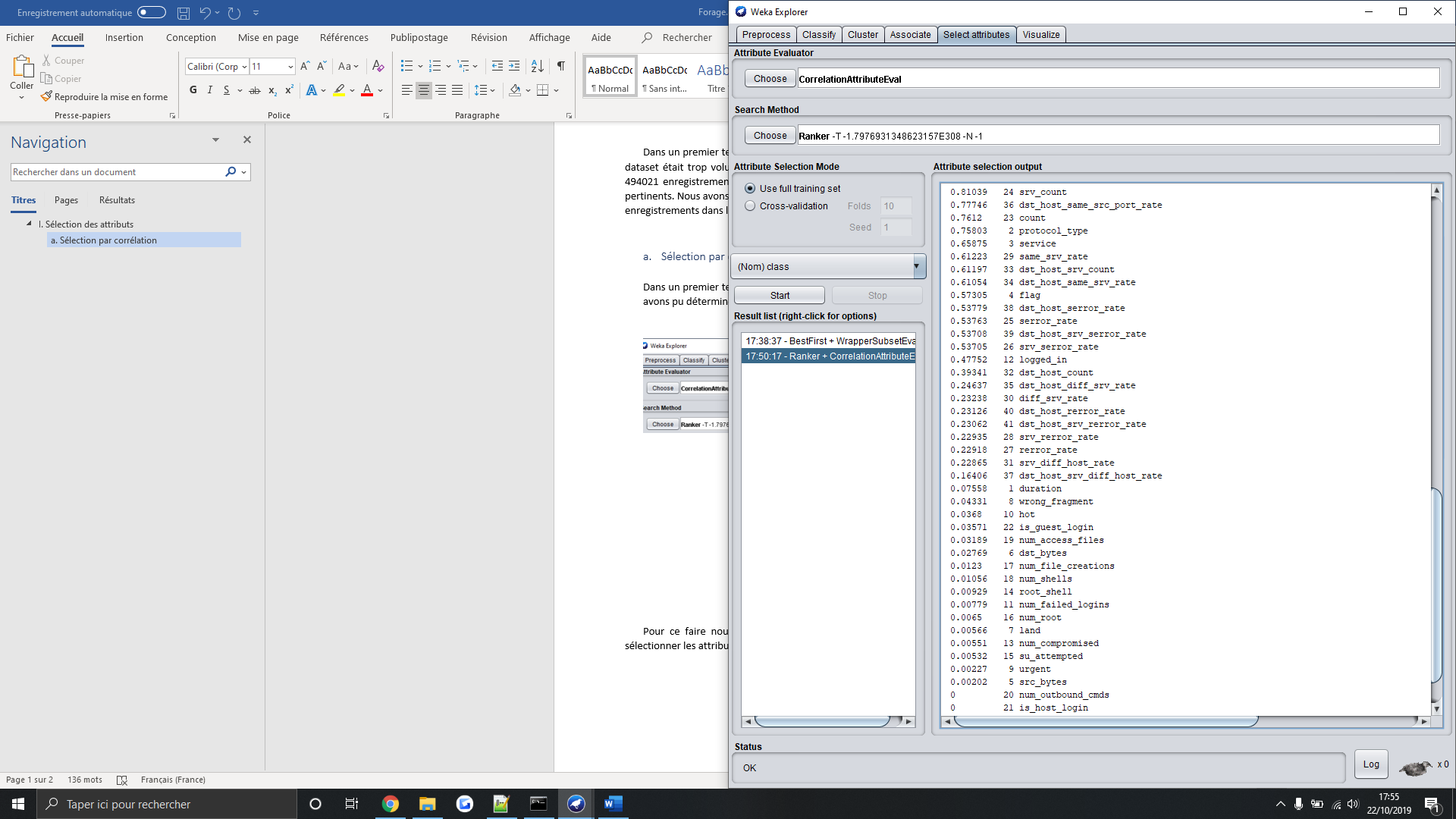
Ensuite, nous avons voulu utiliser les 10% du dataset KDD99 avec ID3, cependant, le dataset était trop volumineux pour être traité. En effet le dataset utilisé possède 42 attributs et 494021 enregistrements. Nous avions alors le choix de le réduire ou de sélectionner les attributs pertinents. Nous avons choisi de sélectionner les attributs pertinents et nous avons préféré garder les enregistrements dans leurs entièretés.

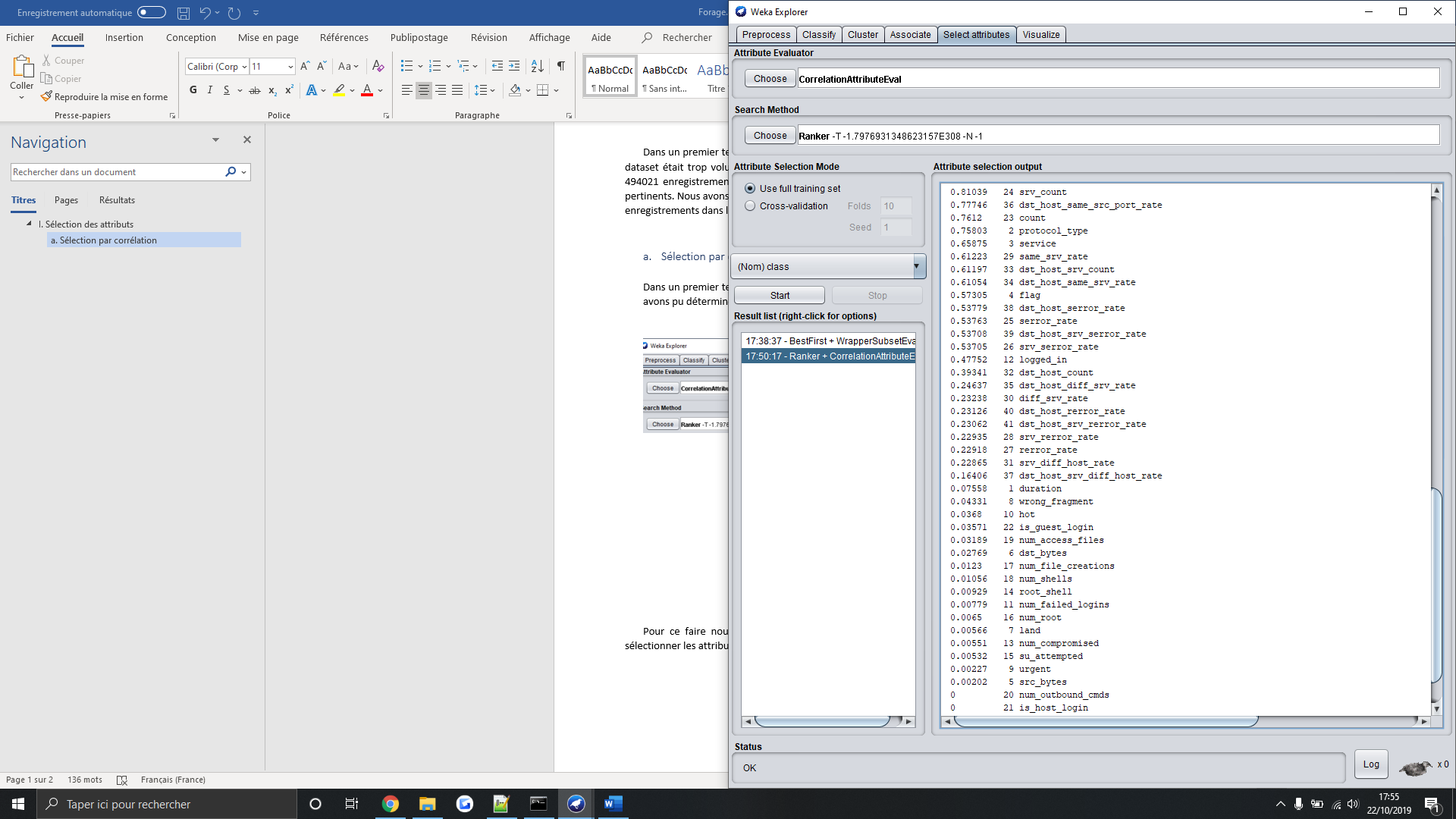
### Sélection par corrélation

Dans un premier temps, nous avons sélectionner les attributs par corrélation grâce au PCC. Nous avons pu déterminer l’importance de chaque attributs (FIG. 1 et FIG. 2)



**FIG 1. Screenshot de la configuration du PCC**

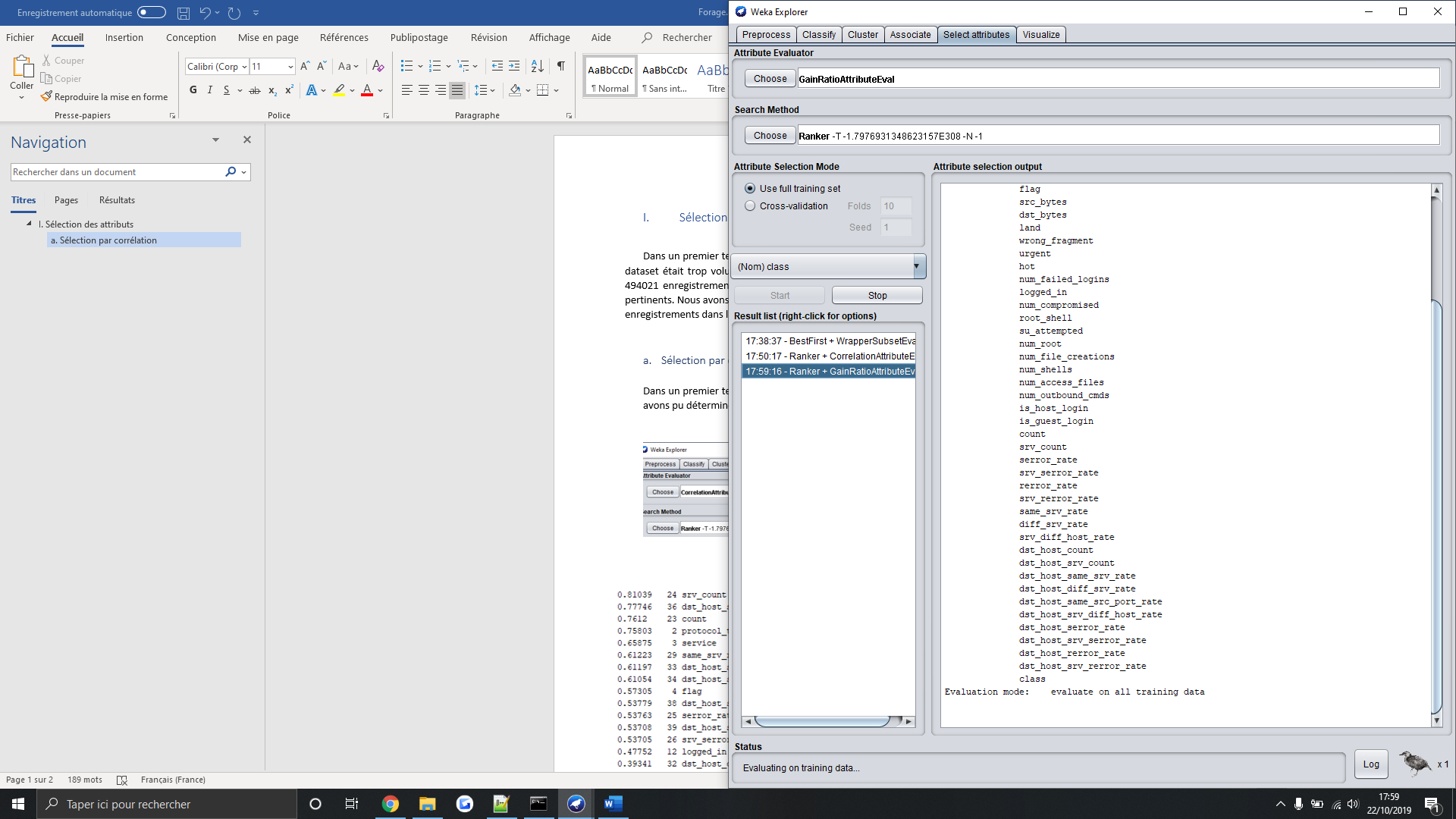




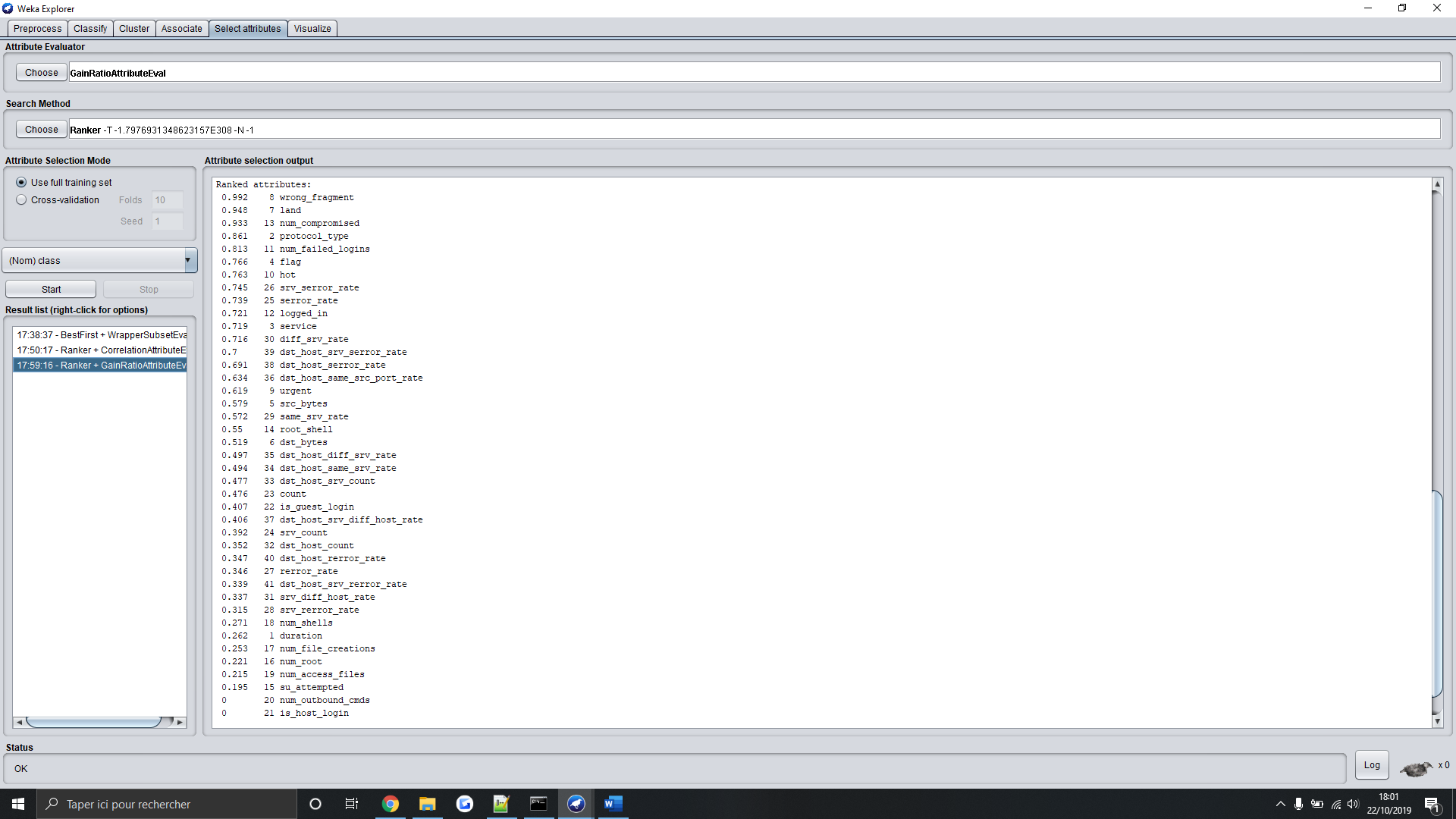
**FIG 2. Screenshot du résultat du PCC**

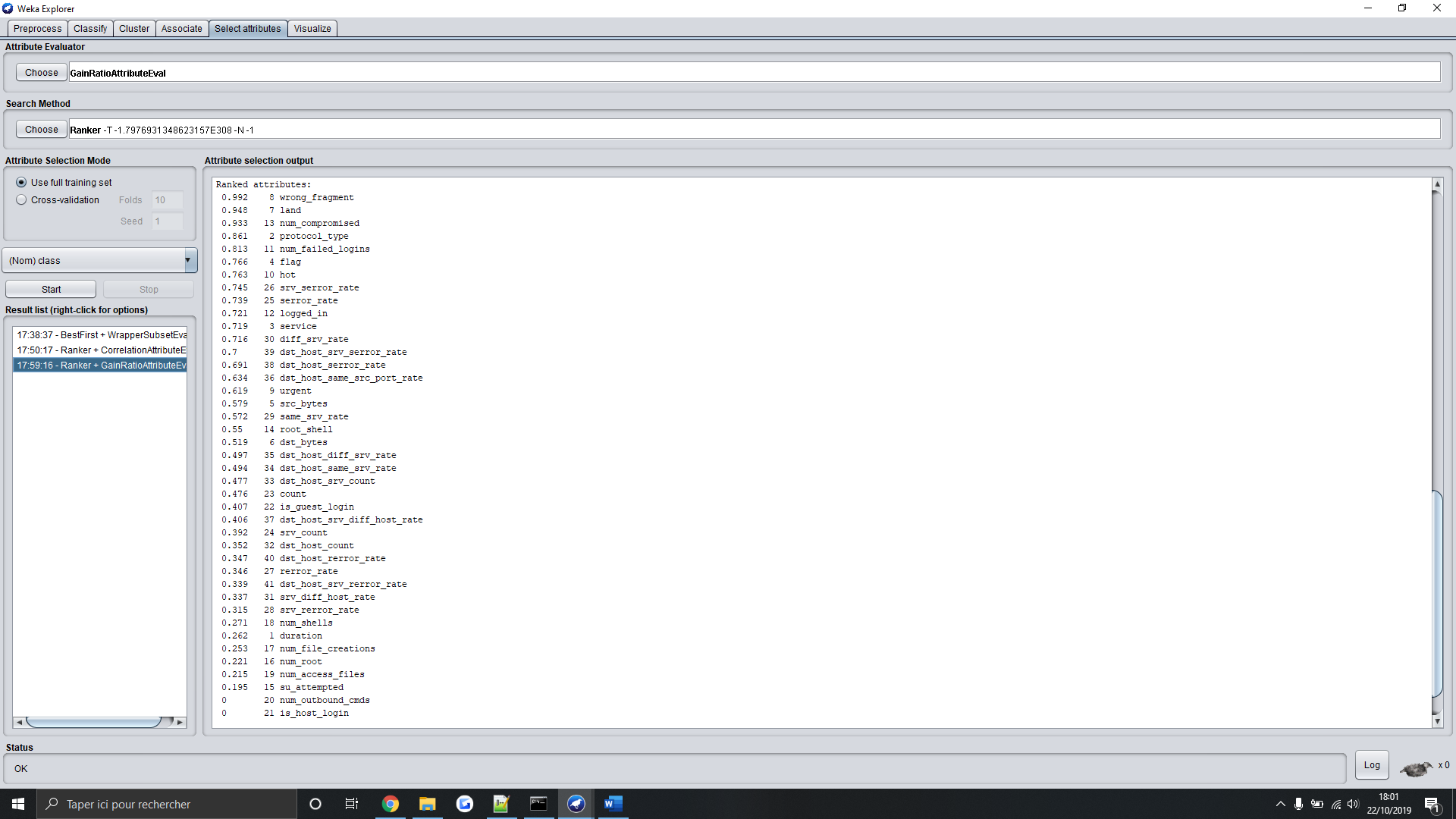
Plus la valeur de l’indice de corrélation est élevée, plus l’attribut est jugé comme important. De ce fait, on peut remarquer que plusieurs attributs n’ont pas l’air important. Pour confirmer nos dires, nous avons effectués un test en fonction du Gain. Sous Weka, nous avons utilisés le GainRatioAttributeEval (FIG. 3) afin de voir l’impact du gain en fonction des attributs.

### Sélection par gain



**FIG 3. Screenshot de la configuration de GainRatioAttributeEval**





**FIG 4. Screenshot du résultat de GainRatioAttributeEval**

**Nous avons ensuite utilisé la moyenne de ces deux indices afin de trouver les attributs les plus importants (FIG 5.).**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Weight\_Correl** | **Weight\_Gain** | **Id** | **Colonne1** | **AttrName** | **Avg** |
| 0,861 | 0,75803 | 2 |  | protocol\_type | 0,809515 |
| 0,634 | 0,77746 | 36 |  | dst\_host\_same\_src\_port\_rate | 0,70573 |
| 0,719 | 0,65875 | 3 |  | service | 0,688875 |
| 0,766 | 0,57305 | 4 |  | flag | 0,669525 |
| 0,745 | 0,53705 | 26 |  | srv\_serror\_rate | 0,641025 |
| 0,739 | 0,53763 | 25 |  | serror\_rate | 0,638315 |
| 0,476 | 0,7612 | 23 |  | count | 0,6186 |
| 0,7 | 0,53708 | 39 |  | dst\_host\_srv\_serror\_rate | 0,61854 |
| 0,691 | 0,53779 | 38 |  | dst\_host\_serror\_rate | 0,614395 |
| 0,392 | 0,81039 | 24 |  | srv\_count | 0,601195 |
| 0,721 | 0,47752 | 12 |  | logged\_in | 0,59926 |
| 0,572 | 0,61223 | 29 |  | same\_srv\_rate | 0,592115 |
| 0,494 | 0,61054 | 34 |  | dst\_host\_same\_srv\_rate | 0,55227 |
| 0,477 | 0,61197 | 33 |  | dst\_host\_srv\_count | 0,544485 |
| 0,992 | 0,04331 | 8 |  | wrong\_fragment | 0,517655 |
| 0,948 | 0,00566 | 7 |  | land | 0,47683 |
| 0,716 | 0,23238 | 30 |  | diff\_srv\_rate | 0,47419 |
| 0,933 | 0,00551 | 13 |  | num\_compromised | 0,469255 |
| 0,813 | 0,00779 | 11 |  | num\_failed\_logins | 0,410395 |
| 0,763 | 0,0368 | 10 |  | hot | 0,3999 |
| 0,352 | 0,39341 | 32 |  | dst\_host\_count | 0,372705 |
| 0,497 | 0,24637 | 35 |  | dst\_host\_diff\_srv\_rate | 0,371685 |
| 0,619 | 0,00227 | 9 |  | urgent | 0,310635 |
| 0,579 | 0,00202 | 5 |  | src\_bytes | 0,29051 |
| 0,347 | 0,23126 | 40 |  | dst\_host\_rerror\_rate | 0,28913 |
| 0,346 | 0,22918 | 27 |  | rerror\_rate | 0,28759 |
| 0,406 | 0,16406 | 37 |  | dst\_host\_srv\_diff\_host\_rate | 0,28503 |
| 0,339 | 0,23062 | 41 |  | dst\_host\_srv\_rerror\_rate | 0,28481 |
| 0,337 | 0,22865 | 31 |  | srv\_diff\_host\_rate | 0,282825 |
| 0,55 | 0,00929 | 14 |  | root\_shell | 0,279645 |
| 0,519 | 0,02769 | 6 |  | dst\_bytes | 0,273345 |
| 0,315 | 0,22935 | 28 |  | srv\_rerror\_rate | 0,272175 |
| 0,407 | 0,03571 | 22 |  | is\_guest\_login | 0,221355 |
| 0,262 | 0,07558 | 1 |  | duration | 0,16879 |
| 0,271 | 0,01056 | 18 |  | num\_shells | 0,14078 |
| 0,253 | 0,0123 | 17 |  | num\_file\_creations | 0,13265 |
| 0,215 | 0,03189 | 19 |  | num\_access\_files | 0,123445 |
| 0,221 | 0,0065 | 16 |  | num\_root | 0,11375 |
| 0,195 | 0,00532 | 15 |  | su\_attempted | 0,10016 |
| 0 | 0 | 20 |  | num\_outbound\_cmds | 0 |
| 0 | 0 | 21 |  | is\_host\_login | 0 |

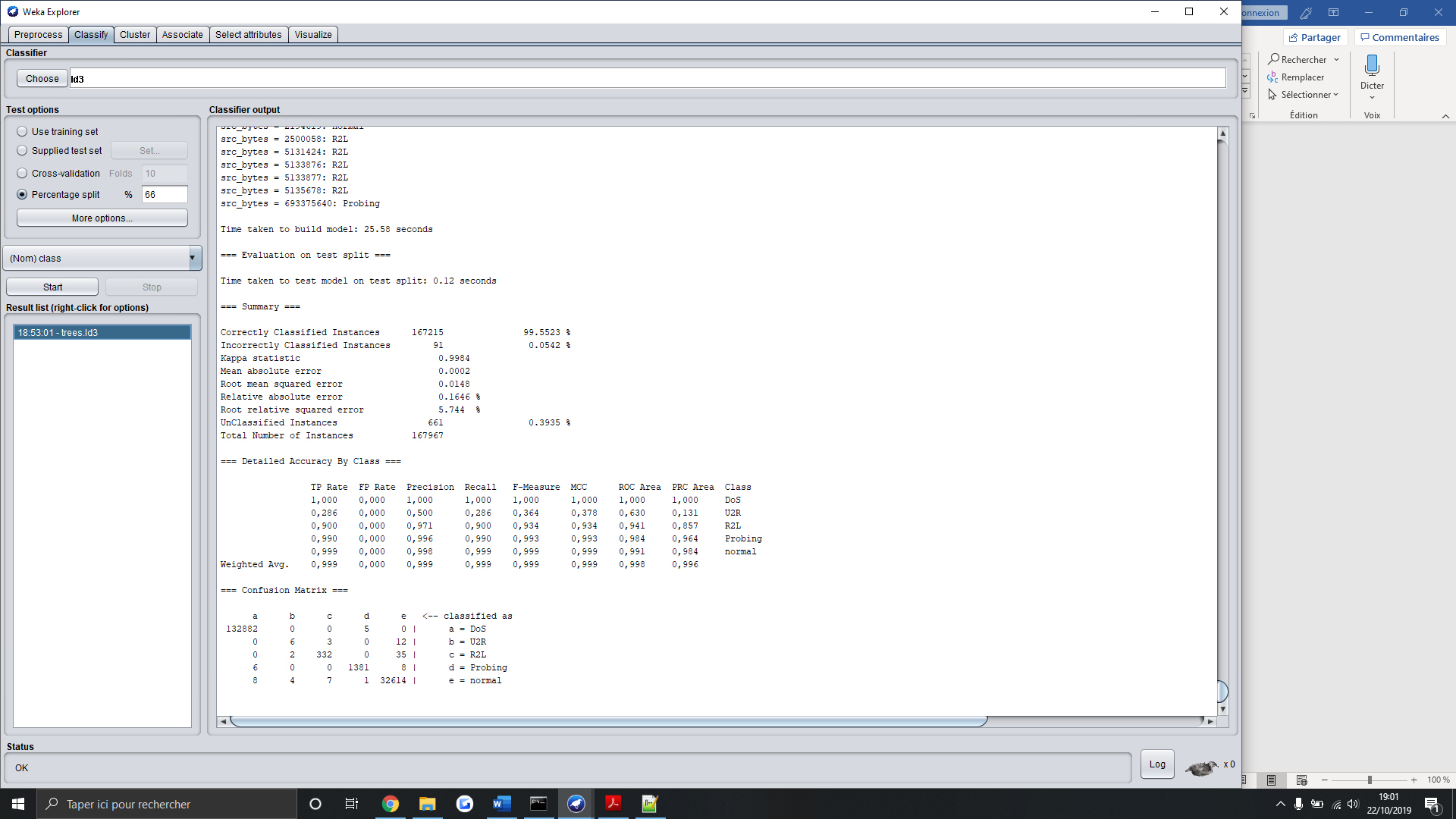
Attributs non utilisés pour le reste de l’étude.

**FIG 5. Moyenne effectuée sur Excel**

## Tests

Nous avons ensuite opté pour une utilisation limite de 10Go MAX par Weka. Puis nous avons enlevés au fur et a mesure les différents attributs par ordre d’importance décroissant, jusqu’à atteindre notre limitation de 10 Go de RAM.

Nous avons donc pu effectuer nos tests avec 28 attributs. Dans un premier temps nous avons effectués un test avec Id3 original. Nous avons utilisé 66% des entrées pour construire le modèle et 33% pour l’essayer. L’algorithme Id3 Original utilise l’entropie de Shannon (FIG 6.), contrairement à l’algorithme proposé qui lui, utilise l’entropie d’Havrda Charvat.



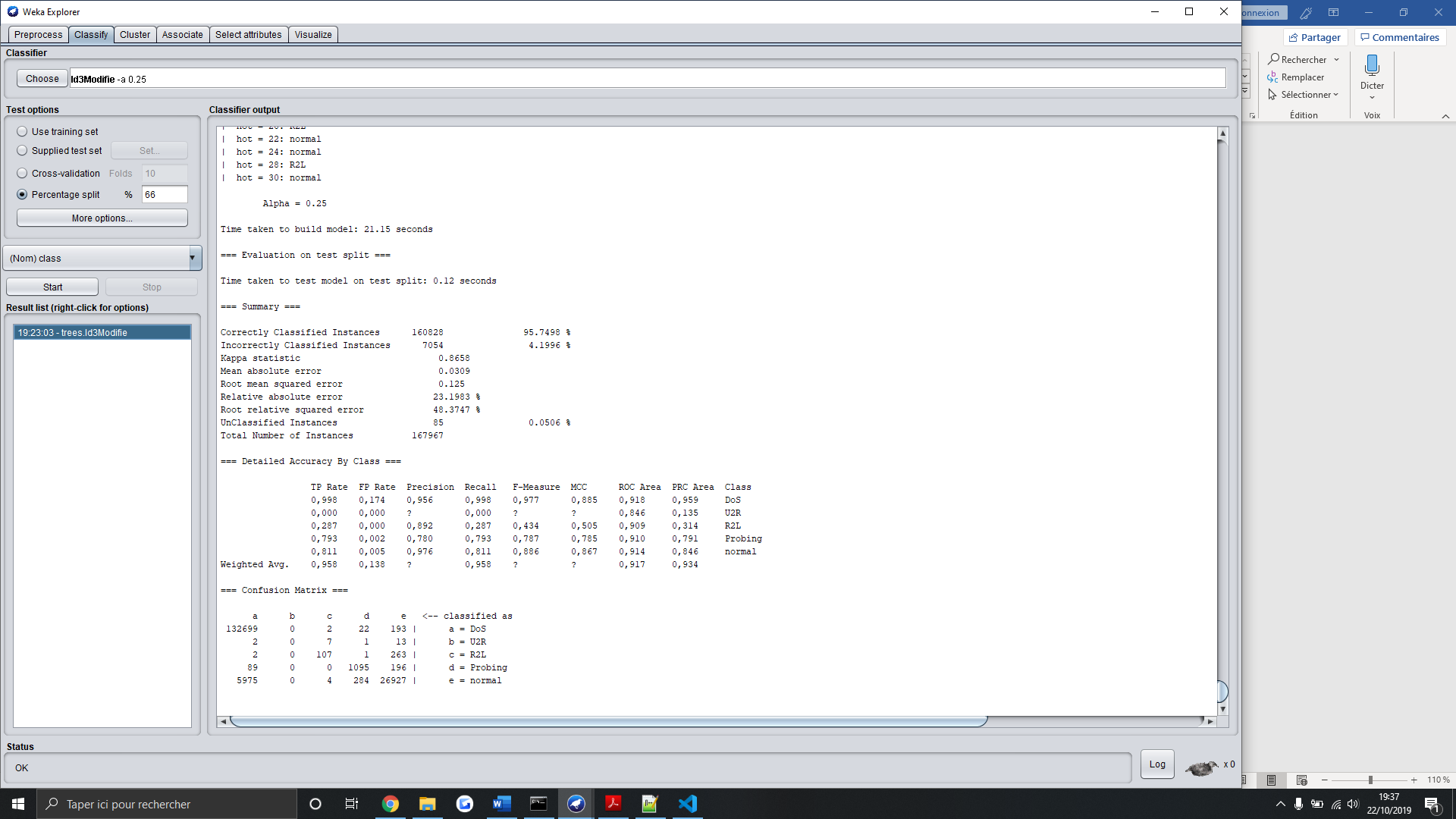
**FIG 6. Résultat ID3 Original**

On remarque plusieurs chose, la première est le temps d’exécution est relativement long (25.58s). Deuxièmement, comme le souligne l’étude, la détection des attaques de type U2R est très approximative, seulement 6 attaques sur 21 ont été classés correctement. Cependant on obtient tout de même un taux d’instance correctement classé de plus de 99% ce qui est plus que correcte.

Nous avons ensuite utilisé ID3Modifie en faisant varier le paramètre Alpha compris entre 0.1 et 2 par pas de 0.2. Nous avons fait en sorte de pouvoir modifier le paramètre Alpha directement depuis l’interface Weka, en effet cela a simplifié les démarches, et nous a évité de devoir recompiler tout le logiciel à chaque fois que l’on désirait changer la valeur du paramètre.

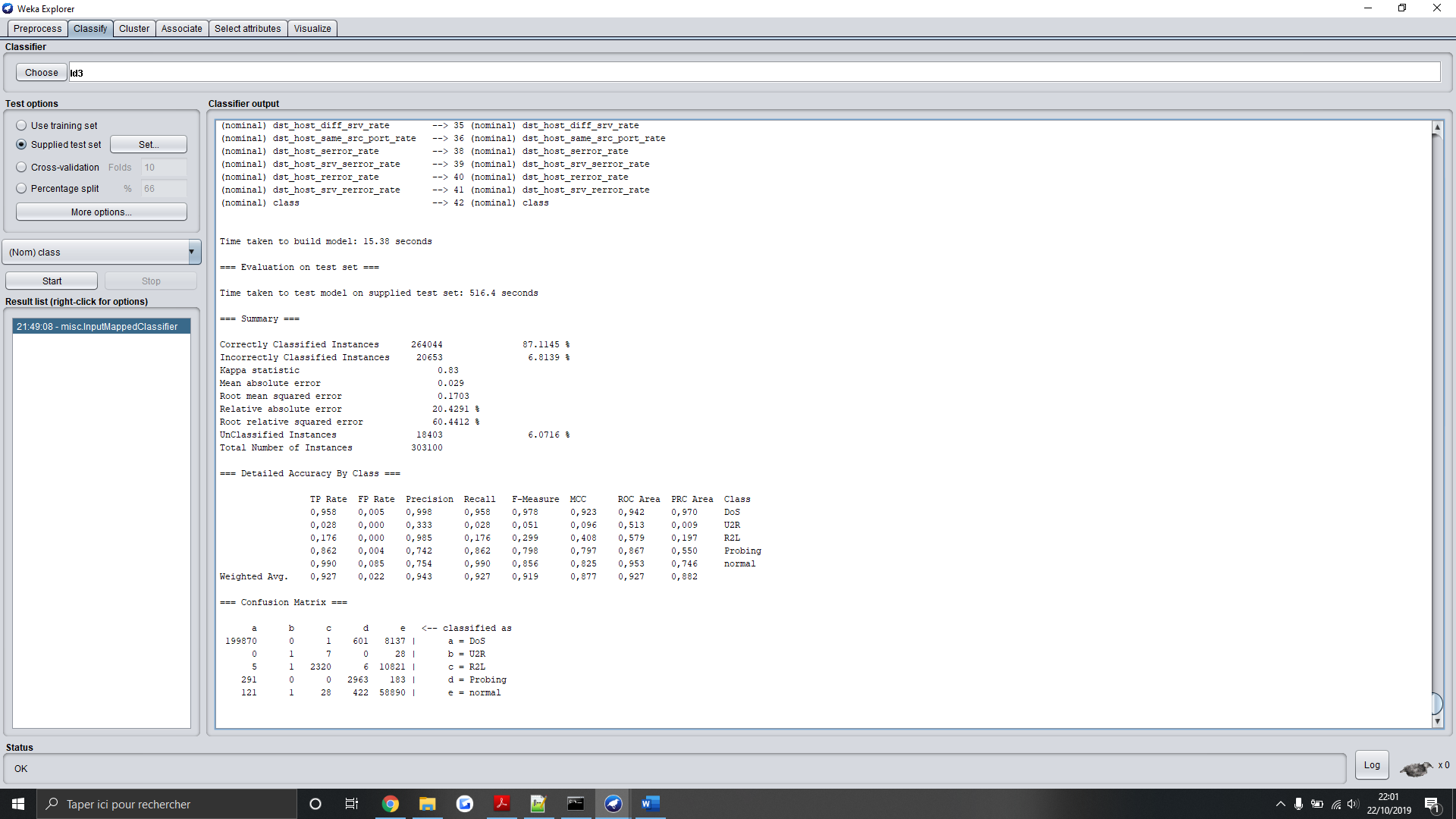
Nous avons cependant constaté que cette entropie donnait des résultats bien moins précis peu importe la valeur du paramètre malgré le gain en vitesse (entre 10% et 20% FIG. 7). De ce fait nous avons vérifié plusieurs fois l’algorithme. Celui-ci semble correcte. Nous avons donc regardé du côté du dataset.

Il faut noter que le dataset ne possède qu’un très faible nombre d’attaque U2R. Ce qui rend la détection très compliquée. Cela explique le faible taux de détection avec l’entropie de Shannon.



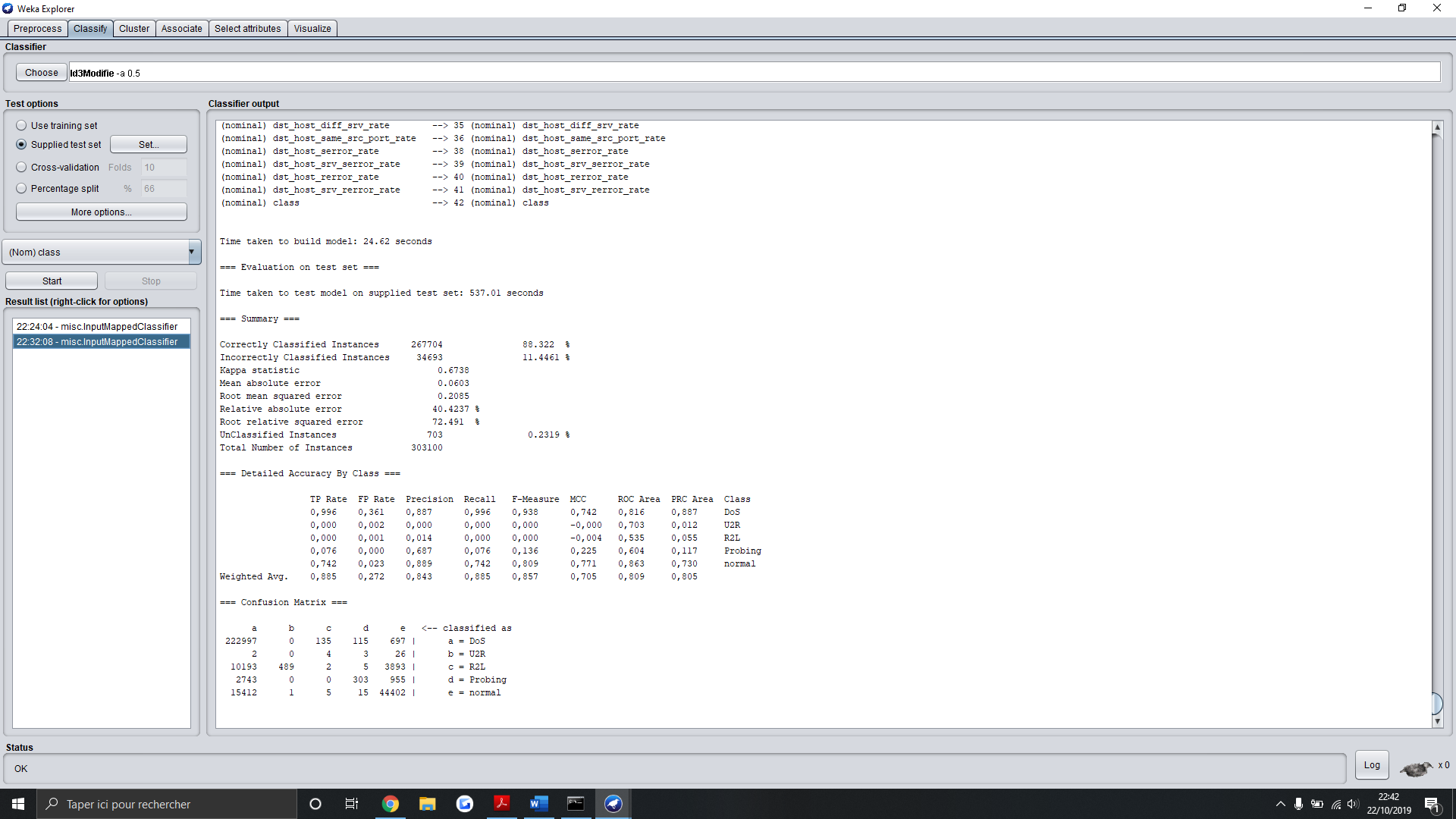
**FIG 6. Résultat ID3 Modifié avec alpha = 0.25**

Afin de vérifier que le problème ne venait pas du dataset, nous avons décidé d’utiliser un dataset échantillon totalement différent pour mener les tests (corrected depuis KDDCup), les résultats obtenus ne sont pas les mêmes qu’avec un unique dataset pour l’apprentissage et le test (Fig 7.). La Figure 7. Montre les résultats de ID3 Original avec un dataset de test totalement différent du dataset d’apprentissage. On peut remarquer que la détection des attaques R2L et U2R sont vraiment très compliquées. Nous sommes encore loin des chiffres proposés par l’étude notamment concernant le ratio de détection des attaques U2R et R2L, cependant, nous avons la même tendance.



**FIG 7. Résultat ID3 Original avec dataset différents**

Malgré les modifications, les résultats avec l’entropie de Charvat ne sont pas concluants. On peut voir sur la Fig 8. Que les résultats sont encore plus mauvais. Nous avons essayé beaucoup de valeurs pour alpha sans succès. On peut donc en conclure que le choix des attributs ainsi que le dataset doit avoir beaucoup d’influence sur la façon dont l’arbre est généré. En effet, les différences obtenues peuvent s’expliquer par le fait des divergences dans les données et dans les attributs entre notre approche et celle de l’article.



**FIG 8. Résultat ID3 Modifié avec dataset différents (Alpha = 0.25)**