Classifications de programmes malicieux et non-malicieux à partir de propriétés binaires

Matthias Beaupère, Pierre Granger

Rapport DA - CHPS - 15 novembre 2018

Table des matières

1 Présentation du jeu de données

Nos données proviennent de la base de données UCI[1]. Cette base de données a été obtenue à partir de l'étude de 373 programmes informatiques malicieux et non-malicieux selon le processus expliqué dans un article de recherche en 2007 [2]. Cet article développe une méthode permettant d'extraire des caractéristiques à partir d'exécutables malins et bénins afin d'effectuer par la suite une classification de ces exécutables permettant de les distinguer. Trois types de carctéristiques sont extraites : des n-uplets binaires, des n-uplets assembleur et des appels à des fonctions appartenant à des librairies extérieures. Les caractéristiques binaires sont extraites des exécutables binaires tandis que les caractéristique assembleur sont obtenues après désassemblage de l'exécutable. Les carctéristiques liées aux appels de fonctions sont extraites depuis l'entête du programme.

2 Pré-traitement des données

Nous avons commencé notre étude des données par quelques visualisations de notre jeu de données. Nous avons tout d'abord visualisé l'histogramme du nombre de caractéristiques possédé par les programmes du jeu de données représenté sur la figure 2. On peut observer sur cet histogramme que la majorité des programmes possèdent une centaine de caractéristiques tandis qu'ils sont très peu à les posséder toutes ou bien à n'en avoir aucune.

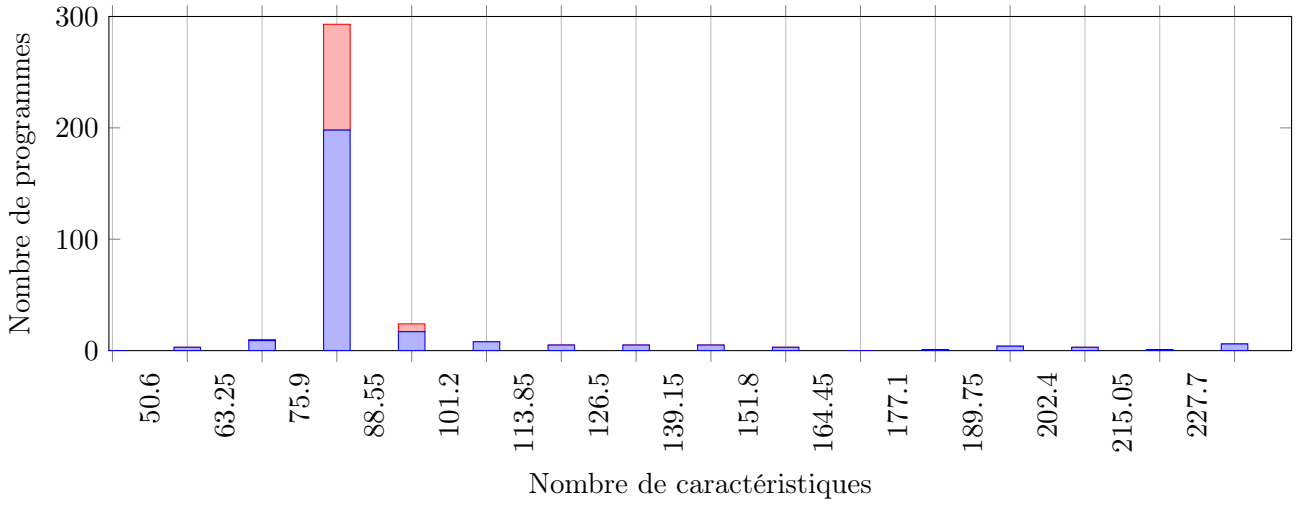


FIGURE 1 – VIRUS

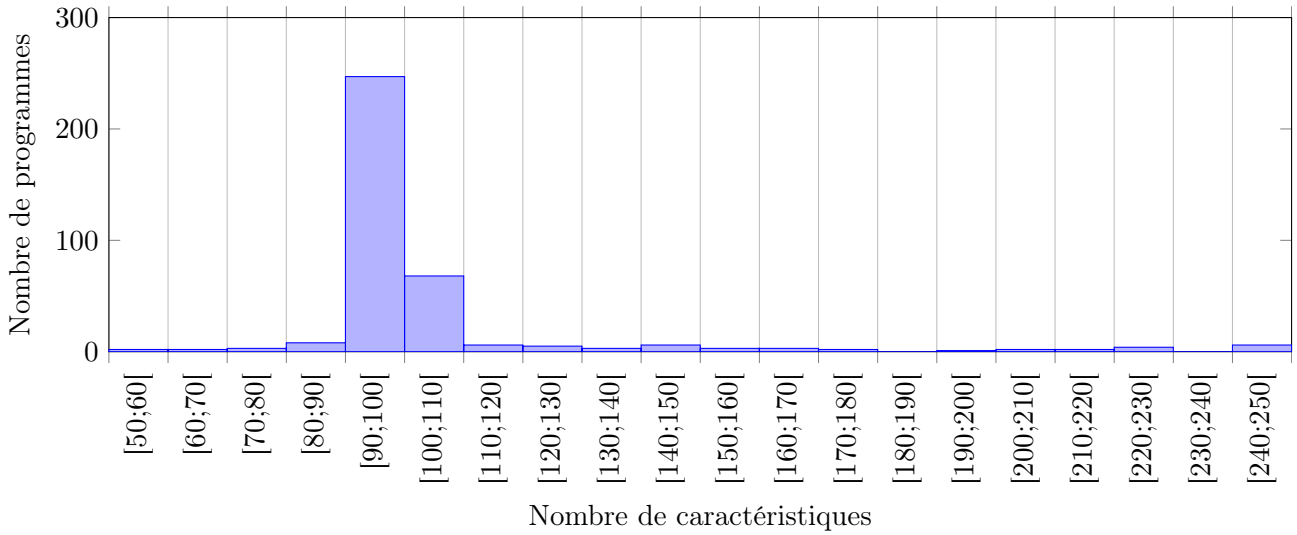


FIGURE 2 – Histogramme du nombre de caractéristiques possédé par chaque programme

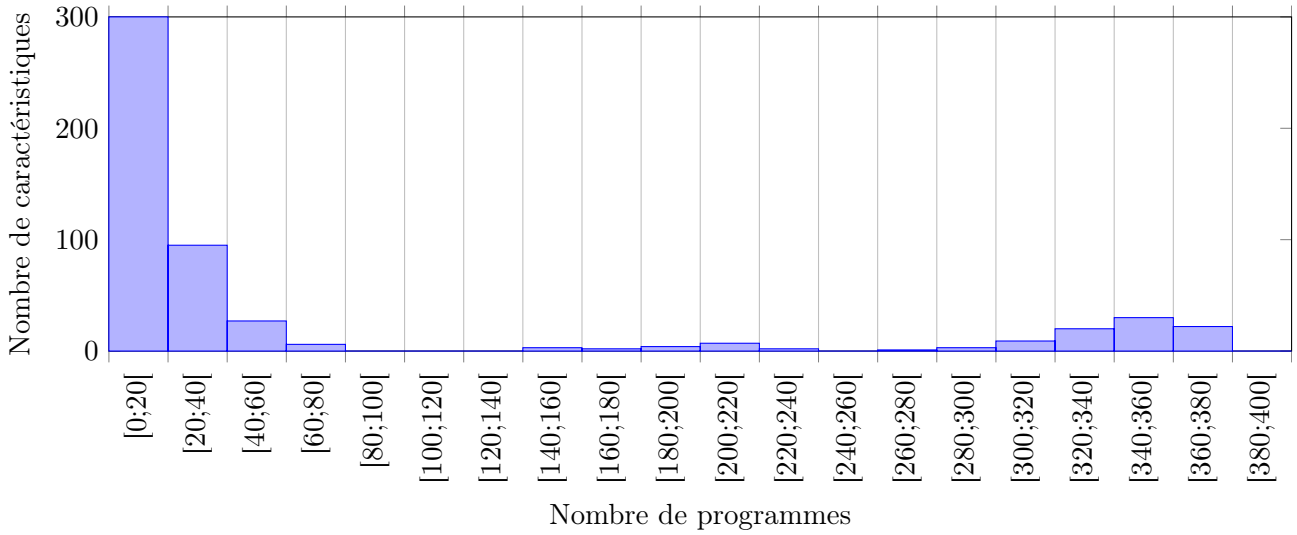


FIGURE 3 – Histogramme du nombre de programme possédant chaque caractéristique

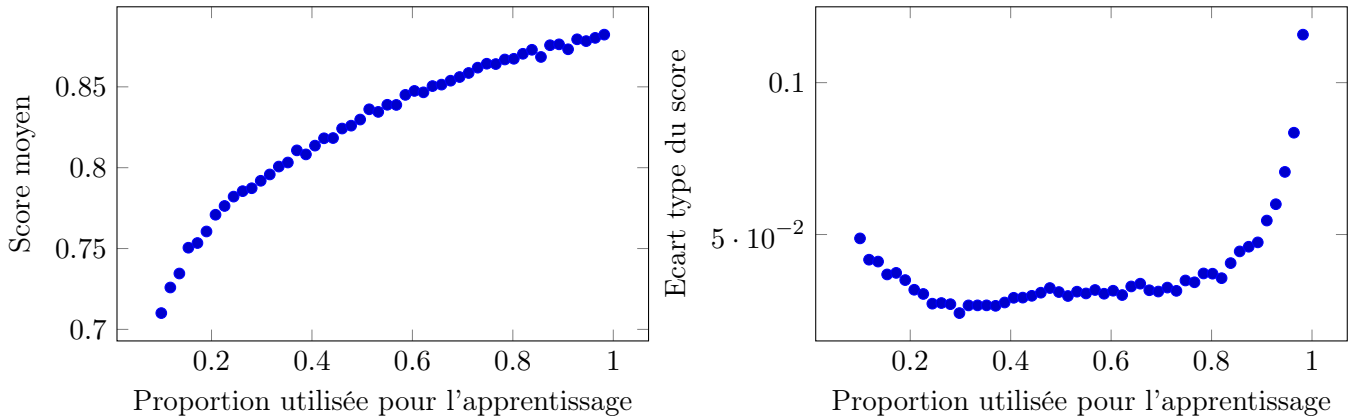
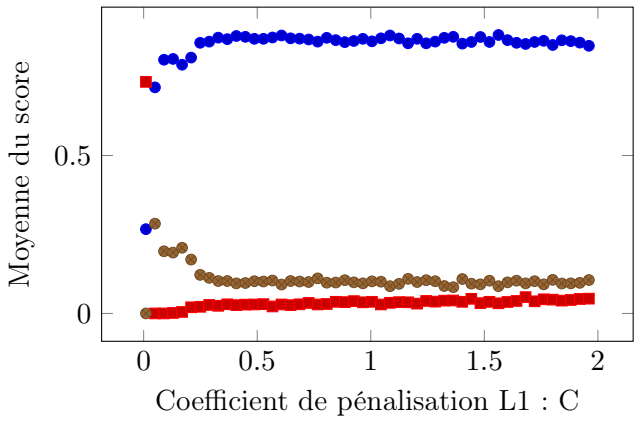


FIGURE 4 – Evolutions de la valeur moyenne et de l'écart type du score en fonction de la proportion de données utilisées pour l'apprentissage dans le cas de la régression logistique. L'évaluation est effectuée sur la partie complémentaire.

3 Différents algo

3.1 LogReg



3.2 SVM

3.3 SVC

3.4 Classification avec kmeans

3.5 Arbres de décision

4 Analyse des résultats

5 Pour aller plus loin...

6 Conclusion

On mérite au moins 21/20.

Références

[1] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2017.

[2] Mehedy Masud, Latifur Khan, and Bhavani Thuraisingham. A hybrid model to detect malicious executables. pages 1443–1448, 06 2007.