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[?]. 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 [?]. 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

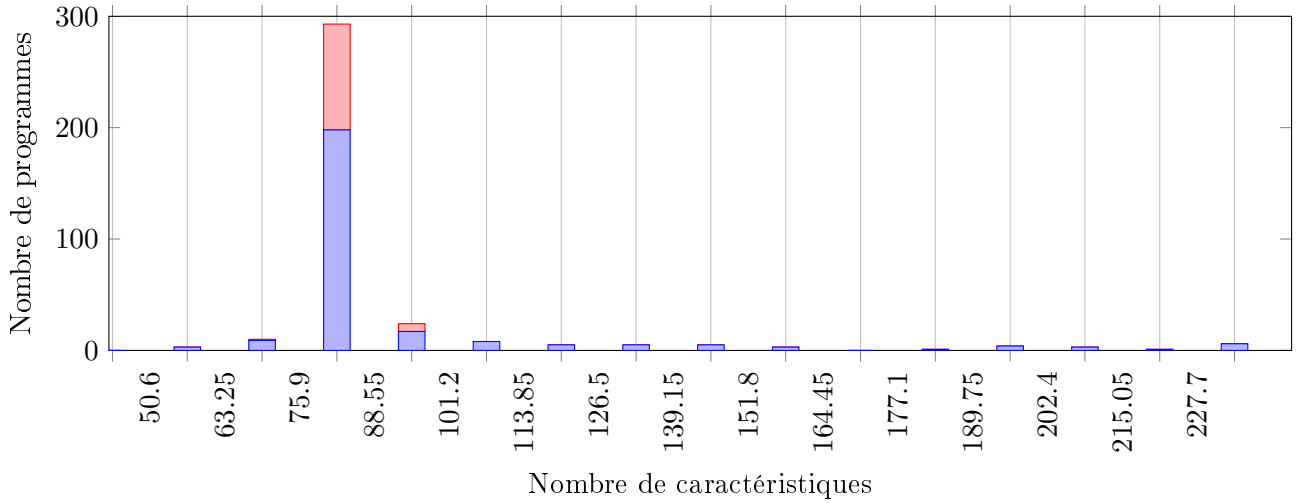


FIGURE 1 – VIRUS

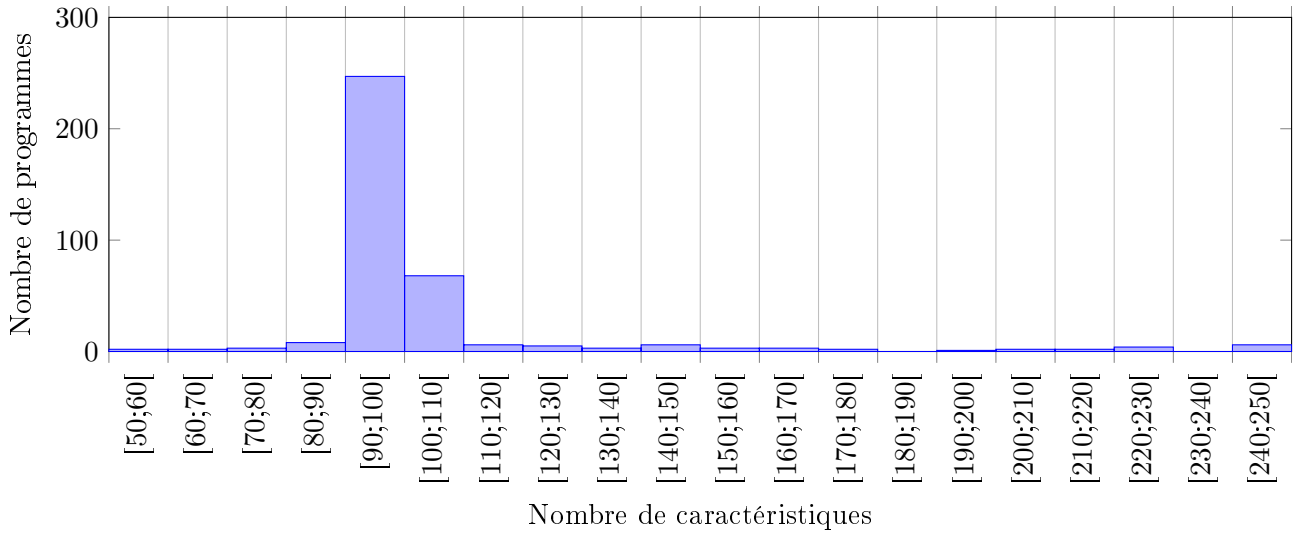


FIGURE 2 – Histogramme du nombre de caractéristiques possédé par chaque programme

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.

3 Techniques de validation

On présente ici les deux techniques utilisées pour valider les modèles de classification.

3.1 Data-splitting

La première technique de validation utilisée est la technique de data-splitting. On divise aléatoirement le jeu de données en deux, la première partie servant à entrainer le modèle. On utilise la deuxième partie

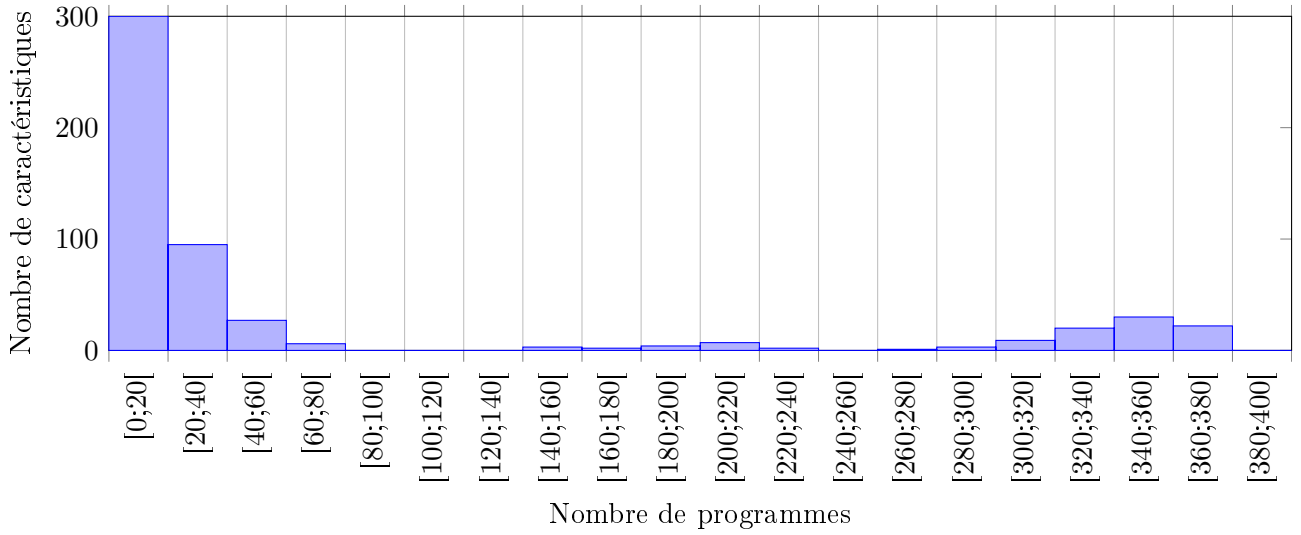


FIGURE 3 – Histogramme du nombre de programme possédant chaque caractéristique

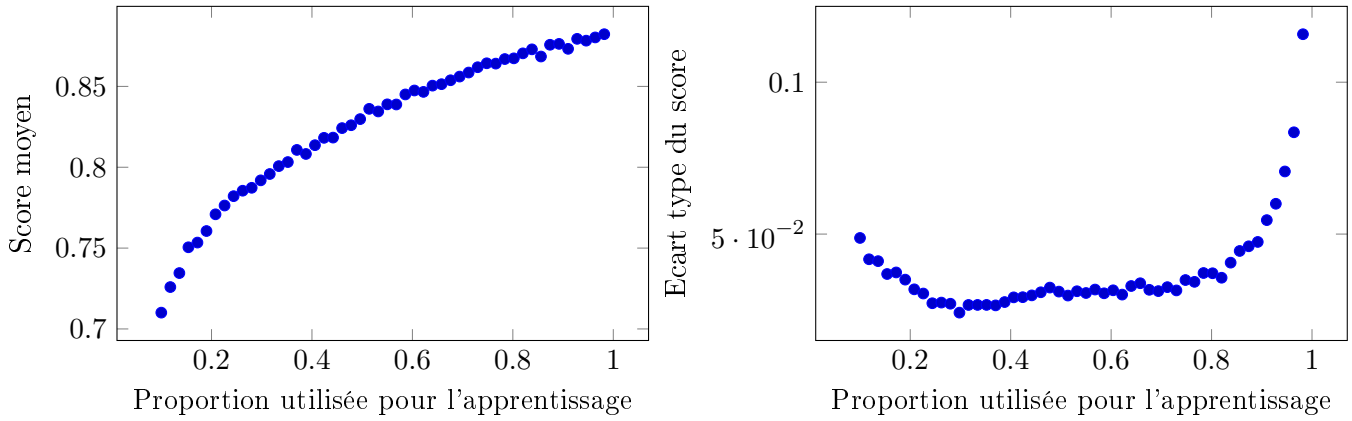


FIGURE 4 – Evolutions de la valeur moyenne et de l'écart type du score en fonction de la proportion de données utilisées pour l'apprentissage dans le cas de la régression logistique. L'évaluation est effectuée sur la partie complémentaire.

des données pour valider le modèle en comparant les prédictions du modèle et les valeurs réelles. Cette technique produit des résultats très différent suivant la répartition des données dans les deux jeux. C'est pourquoi on rejouera souvent un grand nombre de fois le data-splitting et on observera la moyenne et l'écart-type pour connaitre la robustesse de la validation.

3.2 Cross-validation

La deuxième technique consiste à diviser le jeu de données en n parties égales. On itère sur chaque partie de la façon suivant : les $n-1$ autres parties servent à entrainer le modèle et la partie courante est utilisée pour le valider.

4 Différents algo

4.1 LogReg

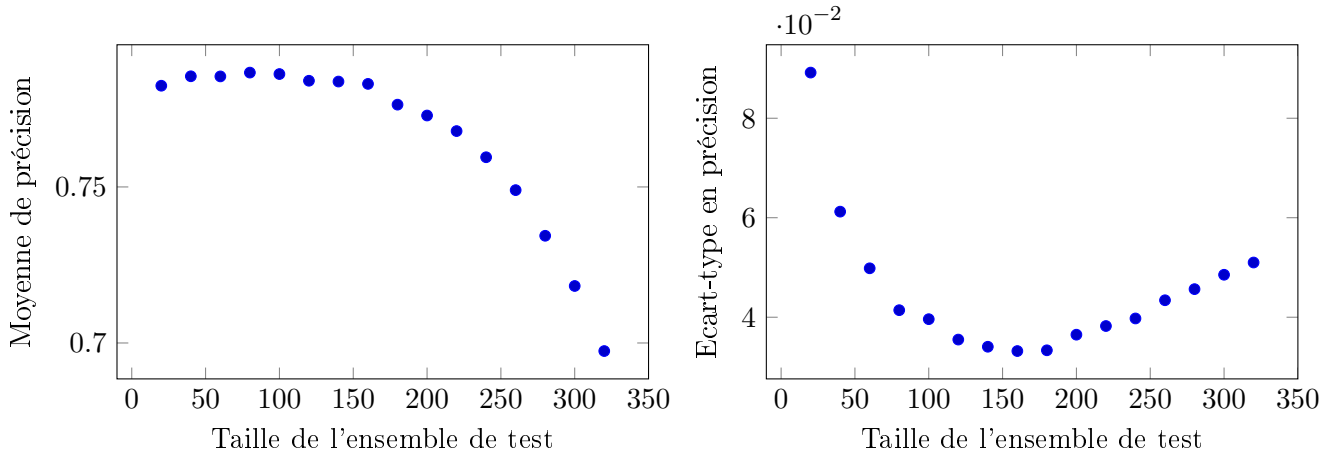


FIGURE 5 – Moyenne et Ecart-type pour 1000 data-splitting

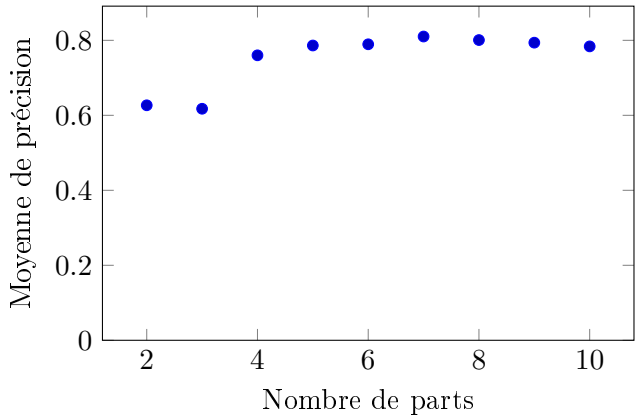
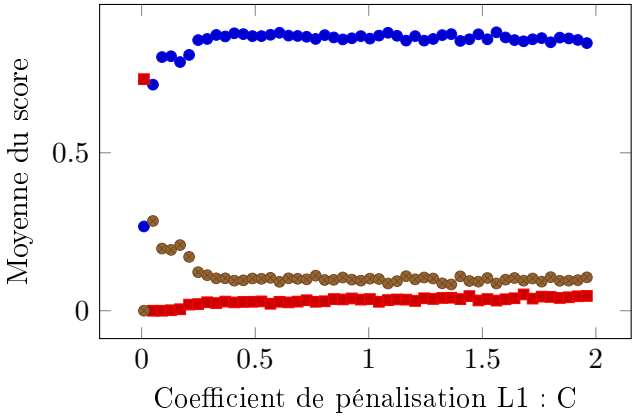


FIGURE 6 – Moyenne pour 1000 cross-validations



4.2 SVM

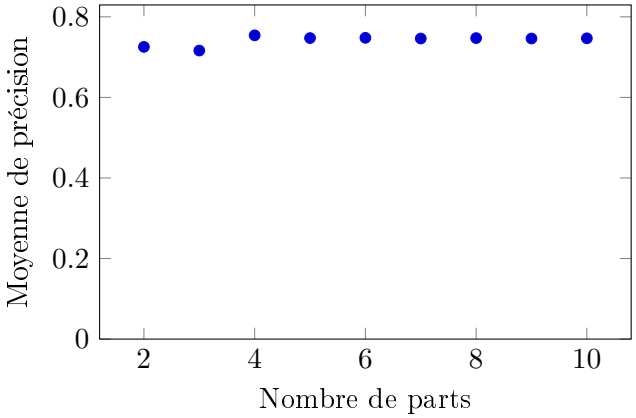
4.3 SVC

4.4 Classification avec kmeans

4.5 Arbre de décision

4.5.1 Arbre simple

4.5.2 Random forest



5 Analyse des résultats

6 Pour aller plus loin...

7 Conclusion

On mérite au moins 21/20.