Supervision de sécurité et détection d'intrusion

Détection d'anomalies par apprentissage

Guillaume HIET
guillaume.hiet@centralesupelec.fr

Equipe CIDRe, **CentraleSupélec**-Inria-IRISA(CNRS)

Septembre 2017

Notes

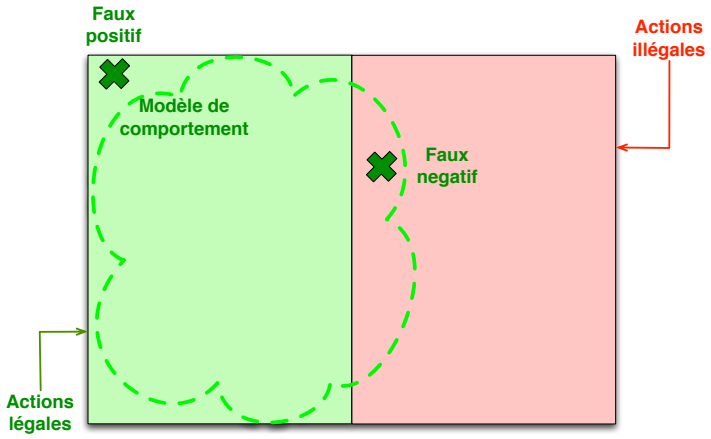
Plan de la présentation

Notes

Rappel : détection d'anomalies

Définition

- Modèle des comportements légaux
- Alerte si activité observée $\neq$ des comportements normaux
- Dans la pratique : analyseur $=$ apprentissage et modèle statistique
  - Légal $\rightarrow$ usuel



Notes

Principes (1/2)

Définition

L'apprentissage automatique (*machine learning*) consiste en l'étude des algorithmes permettant d'apprendre des modèles (concepts) à partir d'exemples sous formes de données (instances). Ces modèles permettent d'expliquer les données ou de prédire sans recourir à la programmation d'un modèle explicite.

- Les instances sont décrites par un ensemble d'attributs (*features)*
- L'ensemble des attributs et leurs valeurs forme un espace vectoriel
- Généralisation : recherche de motifs structurels qui permettent non seulement de décrire les données apprises mais également les nouvelles données (prédiction)
- Machine Learning vs. Data Mining
- Liens avec la statistique, AI, optimisation

Notes

Principes (2/2)

- Algorithme : rechercher, parmi l'espace des concepts, ceux qui représentent au mieux les données d'apprentissage, tout en limitant le sur-apprentissage
- Utilisation d'heuristiques (biais inductif) :
  - Biais de langage
  - Biais de recherche
  - Biais de validation (évite le sur-apprentissage)
- Différentes approches :
  - Apprentissage supervisé
  - Apprentissage non-supervisé
  - Apprentissage semi-supervisé
  - Apprentissage par renforcement
  - etc.

Notes

Différent types de concepts (*output*)

- Classes d'appartenance
  - Chaque exemple est associé à une ou plusieurs classes (probabilité d'appartenance)
  - Apprentissage supervisé
  - Critère d'évaluation : erreur d'association (faux + et faux -)
- Règles d'associations
  - Relations entre les attributs
  - Critère d'évaluation : taux d'application + "utilité"
- Clusters
  - Regrouper les exemples similaires
  - Utilisation possible en pré-traitement à la classification
  - Apprentissage non supervisé
  - *Instance-based learning*
  - Critère d'évaluation : homogénéité, taille, nombre de clusters, "utilité"
- Prédictions numériques

Notes

Exemple tiré de [WFHP16] : règles de décision

Données d'apprentissage

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
...	...	...	...	...

Modèle appris : règles de décision

```
If outlook = sunny and humidity > 83 then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity < 85 then play = yes
Play = yes
```

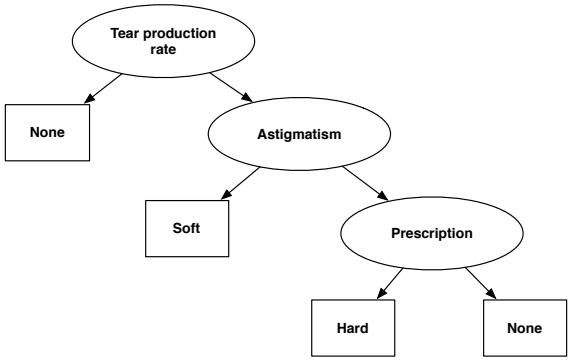

Notes

Exemple tiré de [WFHP16] : arbre de décision

Données d'apprentissage

Age	Prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
...	...	...	...	...

Modèle appris : arbre de décision



Notes

Exemple tiré de [WFHP16] : modèle linéaire

Données d'apprentissage

Cycle time (ns)	Main memory (Kb)		Cache (Kb)	Channels		Performance
MYCT	**MMIN**	**MMAX**	**CACH**	**CHMIN**	**CHMAX**	**PRP**
125	256	6000	256	16	128	198
29	8000	32000	32	8	32	269
480	512	8000	32	0	0	67
480	1000	4000	0	0	0	45

Modèles appris : régression linéaire

$$PRP = -55.9 + 0.0489 * MYCT + 0.0153 * MMIN + 0.0056 * MMAX + 0.6410 * CACH - 0.2700 * CHMIN + 1.480 * CHMAX$$

Notes

Différents types d'instances (*input*)

- On suppose généralement que les instances sont indépendantes
- Instances décrites dans une seule table (*flat file*)
  - Dénormalisation *flattening*
  - Problème de la taille
  - Risque de "découverte" de relations triviales
- Différents types d'attributs
  - Nominal (catégories non ordonnées) : seule la relation d'égalité s'applique
  - Ordinal (catégories ordonnées) : inégalité mais pas de notion de distance
  - Interval (type numérique) : distance entre attributs mais pas de référence
  - Ratio (type numérique) : distance entre attributs + référence
- En pratique, nominal vs. numérique
- Problèmes classiques
  - Valeurs manquantes
  - Valeurs erronées (bruit)

Notes

Quelques outils

Weka

- Framework de référence du domaine
- Implémentation en Java
- `http://www.cs.waikato.ac.nz/ml/weka/`

Scikit-Learn

- Implémentation en Python
- `http://scikit-learn.org/stable/`

R

- Outil très puissant pour les statistiques
- `http://www.r-project.org/`

Notes

(N)IDEs (1/3) [HSJ94]

T.Lunt et al., série de publis de 1990 à 1995

- Soit n variables $S_1, S_2, \ldots, S_n$
- Soit C la matrice de corrélation entre ces variables
- Distance = $(S_1 S_2 \ldots S_n) \times C^{-1} \times {}^t(S_1 S_2 \ldots S_n)$
- Exemples :
  - Si les variables sont indépendantes :
    Distance = $S_1^2 + S_2^2 + \ldots + S_n^2$
  - Si 2 variables sont corrélées à 99% :

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & \ldots \\ 0 & 1 & 0.99 & 0 & \ldots \\ 0 & 0.99 & 1 & 0 & \ldots \\ . & . & . & . & \ldots \end{pmatrix}$$

$$\Rightarrow C^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & \ldots \\ 0 & 50.25 & -49.75 & 0 & \ldots \\ 0 & -49.75 & 50.25 & 0 & \ldots \\ . & . & . & . & \ldots \end{pmatrix}$$

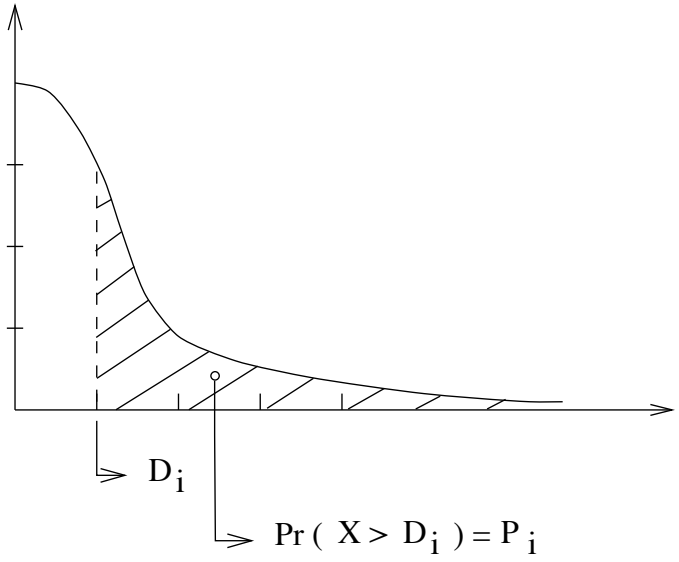
Distance $\simeq S_1^2 + \left(\frac{S_2+S_3}{2}\right)^2 + \ldots + S_n^2$

Notes

(N)IDEs (2/3) [HSJ94]

Problème : les unités des S_i

- Objectif : distance de valeur croissante avec l'anormalité
- Solution : faire l'hypothèse que chaque variable X_i est gaussienne



Notes

(N)IDEs (3/3) [HSJ94]

Observations passées de la grandeur X_i

- 1% des $X_i \in [0,1[\Rightarrow P_1 = 1\% \Rightarrow D_1 = 2,58$
- 3% des $X_i \in [8,max[\Rightarrow P_2 = 3+1 = 4\% \Rightarrow D_2 = 2,06$
- 24% des $X_i \in [4,8[\Rightarrow P_3 = 4+24 = 28\% \Rightarrow D_3 = 1,9$
- 30% des $X_i \in [1,2[\Rightarrow P_4 = 28+30 = 58\% \Rightarrow D_4 = 0,56$
- 42% des $X_i \in [2,4[\Rightarrow P_5 = 58+42 = 100\% \Rightarrow D_5 = 0$

Prochaine observation de X_i : on retient les valeurs :

- $S_i = 0$ si $X_i \in [2,4[$
- $S_i = 0,56$ si $X_i \in [1,2[$
- $S_i = 1,9$ si $X_i \in [4,8[$
- $S_i = 2,06$ si $X_i \in [8,max[$
- $S_i = 2,58$ si $X_i \in [0,1[$

Calcul de la distance ($\sum_{i=1}^{n} S_i^2$ si variables indépendantes) :

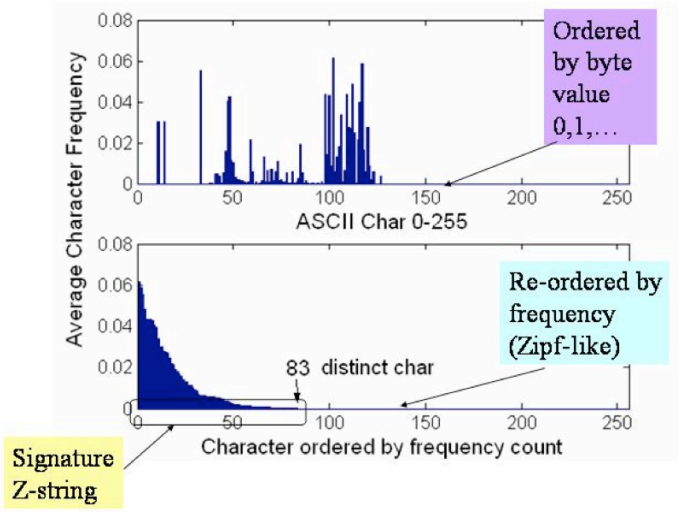
- 0 si toutes les X_i “tombent” dans leur intervalle le plus probable
- un nombre croissant si les X_i “tombent” dans des intervalles de moins en moins probables

Notes

PAYL [WS04]

Wang et Stolfo, Anomalous Payload-based Network Intrusion, RAID 2004

- Modèle == fréquence d'apparition des octets dans un flux http



- Détection : « *each incoming payload is scanned and its byte value distribution is computed. This new payload distribution is then compared against model [ndlr : Mahalanobis Distance] ; if the distribution of the new payload is significantly different from the norm, the detector flags the packet as anomalous and generates an alert* »

Notes

STIDE [FHSL96]

Forrest et al., A Sense of Self for Unix Processes, S&P 1996

- Suite de syscalls :
  open, read, mmap, mmap, open, getrlimit, mmap, ...
- Suite de N-grams :

open	read	mmap	mmap
read	mmap	mmap	open
mmap	mmap	open	getrlimit
mmap	open	getrlimit	mmap
.	.	.	.

- `sendmail` modélisé de manière « stable » avec 1500 6-grams

Taille idéale des N-grams ?

- Kymie M. C. Tan, Roy A. Maxion : "Why 6 ?". IEEE Symposium on Security and Privacy. 2002

Notes

ANAGRAM [WPS06]

Wang, Parekh and Stolfo, Anagram : A Content Anomaly Detector Resistant to Mimicry Attack, RAID 2006

- Résister aux attaque par mimétisme $\rightarrow$ augmenter taille des N-grammes
- Problème : taille de l'espace des attributs (256^n, 640 Go pour 5-gramme)
- Approche fréquentielle peu adaptée (taille réduite des paquets) $\rightarrow$ approche binaire
- Utilisation de filtres de Bloom ($res = h(i) mod n$)
- Critère de détection : nombre de N-gram nouveaux / nombre N-gram du paquet
- Apprentissage semi-supervisé (*bad content model*)
- Permet la génération de signatures
- Résistance intrinsèque au mimétisme + échantillonnage aléatoire
- Problème : saturation du filtre

Notes

Analyse expérimentale [HSB+12]

Hadžiosmanović et al., N-Gram against the Machine : On the Feasibility of the N-Gram Network Analysis for Binary Protocols, RAID 2012

Objectif : évaluer l'efficacité des approches à base de n-grammes pour détecter des attaques sur les protocoles binaires (Samba, ModBus)

Différent types de NAD

- Analyse des flux (entêtes + statistiques agrégées)
- Analyse de la charge (*payload*) des paquets
- Approches complémentaires mais détection de certaines classes d'attaques (injection, etc.) seulement possible avec l'analyse de la charge

Approche retenue

Détection d'anomalies réseau reposant sur des attributs correspondant à des n-grammes extraits de la charge

Notes

Travaux évalués

Différentes approches à base de n-grammes

- Construction de l'espace des attributs
  - Compter les occurrences *count embedding*
  - Mesurer les fréquences relatives*frequence embedding*
  - Noter la présence *binary embedding*
- Modélisation de toute la payload ou d'une partie (échantillonnage, compression)
- Utilisation de différents algorithmes de classification (SOM, Markov Model, etc.)

Algorithmes comparés

- PAYL
- POSEIDON
- ANAGRAM
- McPAD

Notes

Résultats et conclusions

Résultats

- SMB/CIFS : pas de détection avec moins de 1% de FP $\rightarrow$ inutilisable
- Filtrage des messages RPC $\rightarrow$ ANAGRAM donne de bons résultats
- Tous détectent les *shellcodes* (séries de NOP)
- ModBus : ANAGRAM s'en sort très bien
- Pas de meilleur algo a priori : ANAGRAM donne les meilleurs résultats sur RPC filtré et sur ModBus mais les pires résultats sur SMB

Conclusions

- L'efficacité de l'approche dépend fortement du trafic modélisé et des attaques à détecter
- La forte variabilité du traffic réseau limite les performances
- Meilleurs résultats si approche appliquée sur des champs sémantiques particuliers (par exemple URL)

Notes

Comparaison avec d'autres domaines [SP10]

Sommer and Paxson, Outside the Closed World : On Using Machine Learning For Network Intrusion Detection, S&P 2010

Malgré des années de recherche, les IDS utilisant du ML sont rarement déployés dans "la vraie vie"

- Quelques exceptions (Arbor Network Peakflow)
- Cas d'usage spécifique
- Spectre d'attaques détectées limité

Objectifs de l'étude

- Quelles sont les spécificités des IDS par rapport à d'autres domaines où le ML est utilisé avec un certain succès ?
  - Exemples : systèmes de recommandation (Amazon), filtres anti-SPAM, traduction automatique, etc.
- Identifier et comprendre les limites de l'approche
- Etablir des recommandations

Notes

Comment expliquer cette situation ?

Reproches "classiques"

- Approche peu fiable (trop de faux +)
- Nécessité d'apprendre du trafic réaliste mais sain
- Problème des attaques en mimétisme (*mimicry attack*) et qui visent à biaiser l'apprentissage (*learning attack*)

Pistes explorées

- ML classifie bien ce qui a été appris or IDS doit détecter ce qui n'a pas été appris (*outlier detection*)
- Problème du coût des erreurs (faux +)
- Difficulté pour obtenir de bonnes données d'apprentissage
- Gap sémantique
- Variabilité des données
- Difficultés liées à l'évaluation

Notes

Outlier detection

- A la différence des autres cas de succès, il faut détecter ce qui n'a pas été appris
- Similaire à la classification binaire MAIS :
  - Les classes sont fortement déséquilibrées
  - L'apprentissage porte généralement uniquement sur le trafic sain
- ML donne de bon résultats si classe équilibrées et données apprentissage bien répartie entre les classses
- Apprendre uniquement sur des exemples positif (hypothèse *closed world*) donne en général de mauvais résultats car trop peu de données d'apprentissage pour couvrir tous les cas
- A l'inverse, classer différentes catégories équilibrées donne de meilleurs résultats (bonne séparation des classes même si les concepts sont peu précis)
- Exemple des filtres de SPAM : apprentissage sur exemples + et - (apprentissage supervisé possible car classification manuelle "facile")

Notes

Coût des erreurs et fossé sémantique

Coût des erreurs

- En IDS, coût des erreurs important :
  - Faux + nécessite une vérification (d'autant plus difficile en AD, cf gap sémantique)
  - Faux - $\rightarrow$ problème de sécurité potentiellement important
- Différence par rapport aux autres domaine
  - exemple : erreur de recommandation, filtres anti-spam (cf coût vérification)
- Problème des faux + exacerbé par le déséquilibre des classes : cf *base-rate falacy*

Fossé sémantique

- En entrée : la sélection des attributs devrait tenir compte de la sémantique du protocole et des attaques
- En sortie : le résultat de la classification donne souvent peu d'information exploitable par un opérateur de sécurité sur les causes de cette classification

Notes

Variabilité des données et évaluation

Variabilité des entrées

- Souvent, très grande variabilité des grandeurs (débit, temps d'arrivé, taille des paquets, valeurs des données, etc.) sur une courte période pour du trafic sain
- Situation difficile à modéliser pour les algo de ML (difficile de trouver des motifs)
- Améliorer en considérant des données agrégées sur une plus grande période (volume par heure, moyenne, etc.)
  - Limite : attaques détectées souvent triviales et facilement détectables par d'autres approches *ad hoc* plus simples

Evaluation

- Importance de l'évaluation : difficile de prévoir ce que va détecter un AD ("boite noire")
- Pas de dataset public à jour, difficile de générer des dataset
- Problème de *privacy* $\rightarrow$ simuler ou anonymiser

Notes

Conclusion

- L'apprentissage automatique donne de bon résultats en IDS si :
  - On a une idée de ce que l'on veut détecter
  - On restreint la classe des attaques détectables
  - On sélectionne les attributs en conséquence
- *Anomaly Detection* $\rightarrow$ *misuse-based detection*
- Limite de l'approche "boite noire"
  - Il faut comprendre pourquoi le système se trompe ou a raison
  - Ne pas se limiter à la classification : importance des concepts, des attributs
  - Connaissance réutilisable (par exemple, en utilisant d'autres formes de détection)

"The intrusion detection community does not benefit any further from yet another study measuring the performance of some previously untried combination of a machine learning scheme with a particular feature set, applied to something like the DARPA dataset. The nature of our domain is such that one can always find a variation that works slightly better than anything else in a particular setting."

Notes

Bibliographie

[FHSL96] Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, and Thomas A. Longstaff, A Sense of Self for Unix Processes, Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society, IEEE Computer Society Press, May 1996, pp. 120–128.

[GT07] Carrie Gates and Carol Taylor, Challenging the anomaly detection paradigm : A provocative discussion, Proceedings of the 2006 Workshop on New Security Paradigms (New York, NY, USA), NSPW '06, ACM, 2007, pp. 21–29.

[HSB+12] Dina Hadžiosmanović, Lorenzo Simionato, Damiano Bolzoni, Emmanuele Zambon, and Sandro Etalle, N-gram against the machine : On the feasibility of the n-gram network analysis for binary protocols, Research in Attacks, Intrusions, and Defenses, Lecture Notes in Computer Science, vol. 7462, Springer Berlin Heidelberg, 2012.

Notes

Bibliographie (cont.)

[HSJ94] Alfoso Valdes Harold S. Javitz, The nides statistical component description and justification, Tech. Report A010, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025-3493, March 1994.

[PAF+09] Roberto Perdisci, Davide Ariu, Prahlad Fogla, Giorgio Giacinto, and Wenke Lee, Mcpad : A multiple classifier system for accurate payload-based anomaly detection, Comput. Netw. **53** (2009), no. 6, 864–881.

[SP10] Robin Sommer and Vern Paxson, Outside the closed world : On using machine learning for network intrusion detection, Proceedings of the 2010 IEEE Symposium on Security and Privacy (Washington, DC, USA), SP '10, IEEE Computer Society, 2010, pp. 305–316.

Notes

Bibliographie (cont.)

[WFHP16] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal, Data mining, fourth edition : Practical machine learning tools and techniques, 4th ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2016.

[WPS06] Ke Wang, Janak J. Parekh, and Salvatore J. Stolfo, Anagram : A Content Anomaly Detector Resistant to Mimicry Attack, Recent Advances in Intrusion Detection (Diego Zamboni and Christopher Kruegel, eds.), Lecture Notes in Computer Science, vol. 4219, Springer, 2006, pp. 226–248.

[WS04] Ke Wang and Salvatore J. Stolfo, Anomalous Payload-Based Network Intrusion Detection, Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID'2004) (Erland Jonsson, Alfonso Valdes, and Magnus Almgren, eds.), Lecture Notes in

Notes

Bibliographie (cont.)

Computer Science, vol. 3224, Springer, September 15-17 2004, pp. 203–222.

Notes