Network-based intrusion detection using neural networks

Encadré par M. Leray Philippe

Florian Boitrel – Maxime Chambreuil

25 juin 2003

Déroulement

- Présentation de l'article
- Fonctionnement général
- La carte de Kohonen
- Le réseau de neurones : MLP
- Résultats
- Difficultés rencontrées
- Conclusion

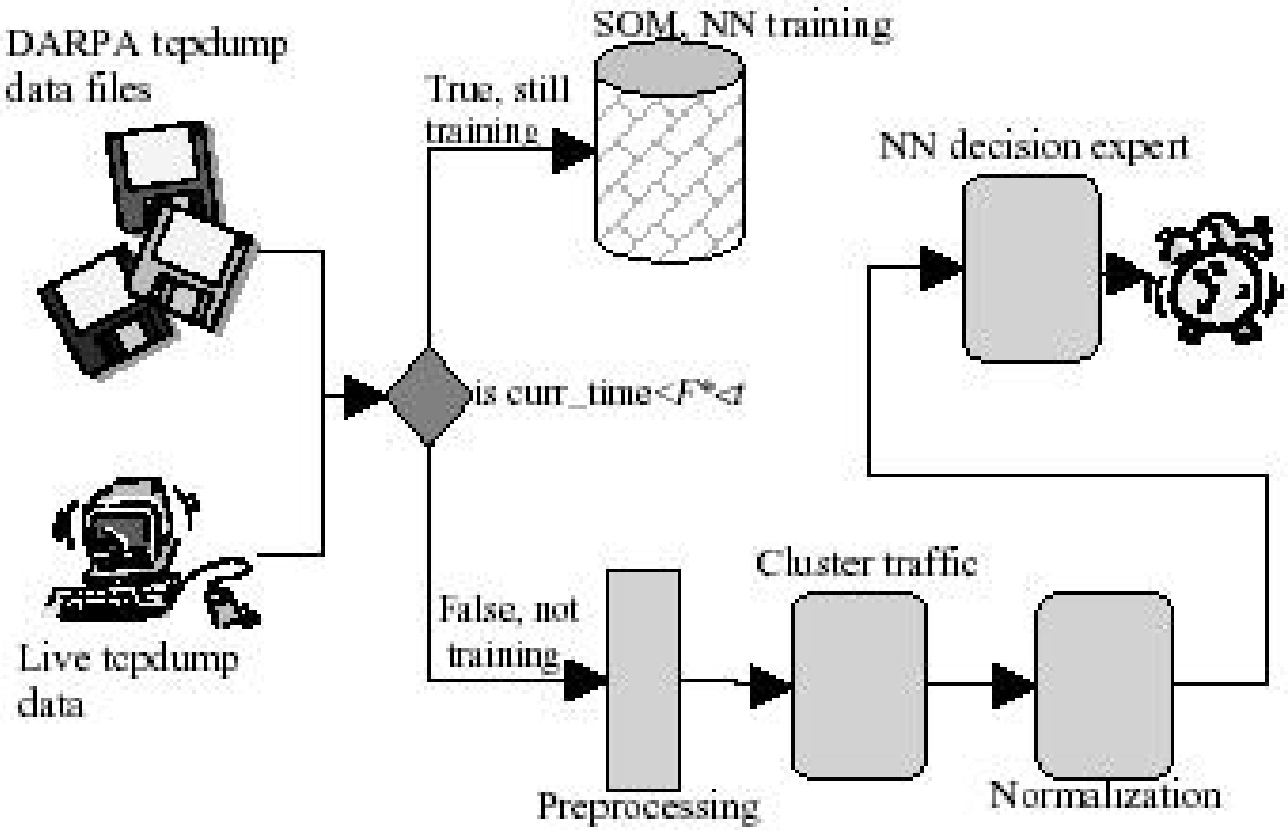
Présentation de l'article

- Les auteurs : Alan Bivens, Chandrika Palagiri, Rasheda Smith, Boleslaw Szymanski, Mark Embrechts

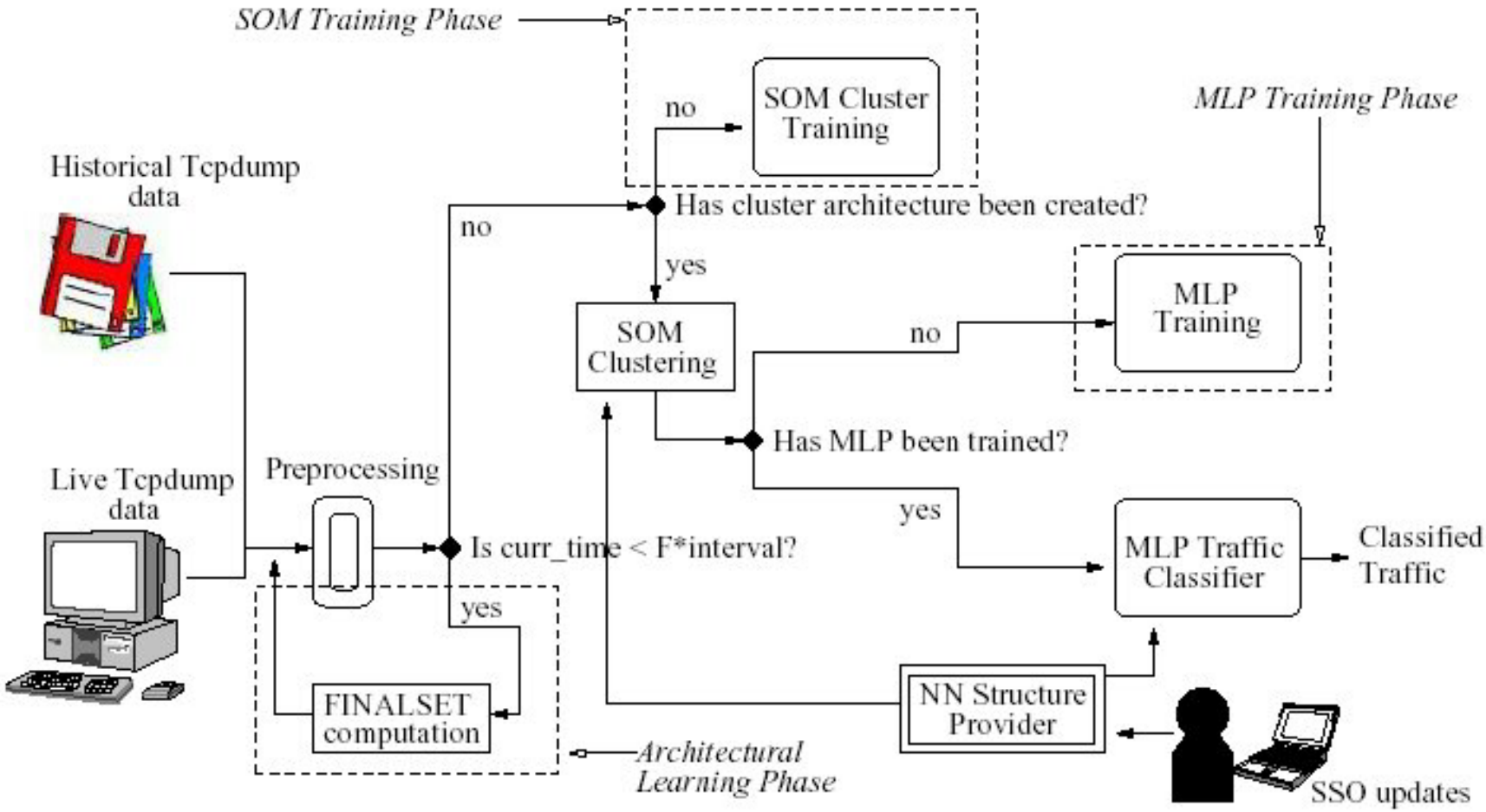
- Rensselaer Polytechnic Institute, NY

- 23 novembre 2002

Fonctionnement (1/2)



Fonctionnement (2/2)



Tcpdump

Timestamp	Temps de réception du paquet par tcpdump
Source	Machine qui envoit le paquet
Destination	Machine qui reçoit le paquet
Protocol	Protocole utilisé pour transmettre le paquet
Source Ports	Port à partir duquel le paquet est envoyé
Destination Ports	Port de destination

Carte de Kohonen

La carte de Kohonen (1/4)

- Rôle : distinguer des comportements identiques entre des machines, les regrouper pour former des « sources »

  - Il peut y avoir plusieurs sources en communication avec la victime
  - Une machine peut être utilisée en tant que source d ’attaque pendant un certain laps de temps et en tant que source normale pendant un autre laps de temps

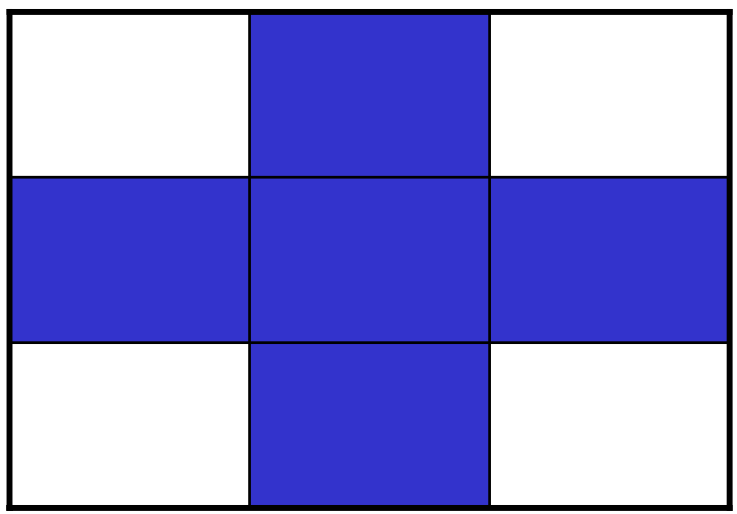
La carte de Kohonen (2/4)

- En entrée :

Port \ Machine	Machine 1	Machine 2	Machine 3
21	3	4	0
22	5	0	10
23	0	2	5
25	1	13	6
80	20	15	100

La carte de Kohonen (3/4)

- BMU : Best Maching Unit
- On récupère les 4 meilleurs BMUs, centres de chaque « source »
- Le voisinage :



La carte de Kohonen (4/4)

- En sortie, on obtient les vecteurs des 4 BMUs, formant chacun une « source »

- Conclusion : La carte de Kohonen sert toujours en apprentissage et jamais pour obtenir le neurone qui répond à une donnée

Réseau de Neurones

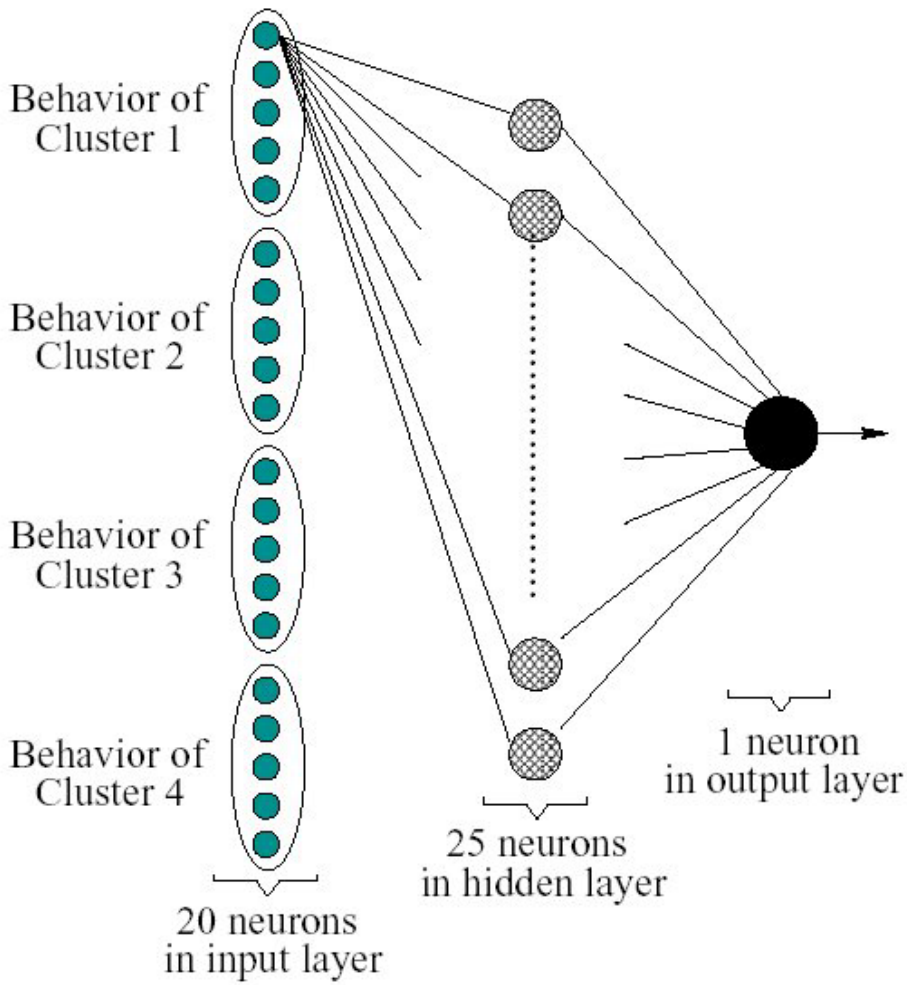
M L P

25 juin 2003

Le réseau de neurones : MLP (1/2)

- 4 ports spécifiques + 1 Extra Port
- 4 sources en entrée (4*5 = 20 neurones), issues de la carte de Kohonen
- 100 données d ’apprentissge
  (*early-stopping*, avec autant de données « attaques » que de données « normales » et 10000 itérations MAX)
- Tests sur 50 données
- Une couche cachée de 25 neurones
- Une sortie (un neurone, une valeur : attaque ou pas)

Le réseau de neurones : MLP (2/2)



Résultats

- On ne peut pas détecter toutes les attaques
- On peut détecter une attaque spécifique sur un port
  - **Name**: sendmail **Start_Time**: 08:48:12
    Duration: 00:00:02
    Ports: At_Attacker: 113{1} At_Victim: 25{1}

  - **Name**: New_Attack_1999 (sshprocesstable) **Start_Time**: 16:49:15 **Duration**: 00:08:21
    Ports: At_Attacker: At_Victim: 22{490}

Difficultés (1/2)

- Le prétraitement :
  - Les auteurs ont un programme Java, qu ’ils n ’ont jamais réussi à nous envoyer
  - Faire un programme (Perl) : long et fastidieux à mettre en œuvre
  - Utiliser un logiciel (Tcptrace) : utilisation peu concluante

⇒ **Fabrication de données aléatoires**

Difficultés (2/2)

- Quand réapprendre le MLP ?
  - Périodiquement, après une phase de test pour configurer la période d'un cycle (App + Détection)
  - Automatiquement, mais avec quels critères ?

Conclusion

- Projet intéressant
- Utilisation de la carte de Kohonen ingénieuse
- Résultats bons pour un type d ’attaque mais médiocres pour l ’union de toutes les attaques
- Une fois l ’apprentissage fait, le mlp peut effectuer des détections temps réels
- Futur : projets de recherche ?

Sources

- **L ’article**
  - *http://www.cs.rpi.edu/%7Eszymansk/papers/annie02.pdf*
- **DARPA Intrusion Detection**
  - *http://www.ll.mit.edu/IST/ideval/data/data_index.html*
- **Liste attaques**
  - *http://www.ll.mit.edu/IST/ideval/docs/1999/master_identifications.list*
- **Thèse d ’Alan BIVENS**
- **Tcpdump**
  - *http://www.tcpdump.org/*

Merci de votre attention

Des questions ???