Rapport d'avancement à mi-parcours

Alexandre Lung-Yut-Fong

30 mars 2010

Titre de la thèse : Détection d'anomalies à partir de mesures collectées par un réseau de capteurs et application à la détection d'attaques dans le trafic Internet.

Responsables de la thèse : Olivier Cappé, Céline Lévy-Leduc

1 Introduction

La détection d'anomalies est une problématique importante ; les applications sont nombreuses : le contrôle de qualité, la détection de fraudes bancaires, le diagnostic médical, etc. Nous nous intéressons particulièrement à la sécurité des infrastructures de réseaux informatiques. Les systèmes de détection d'intrusion (*Intrusion Detection Systems*, ou *IDS*) permettent de repérer les comportements anormaux visant un réseau ou un hôte, résultant la plupart du temps d'attaques ayant un but malveillant. Typiquement, une attaque par déni de service (*Denial of Service*, ou *DoS*) vise par exemple à rendre indisponible les services (site web, messagerie éléctronique, serveur de fichiers, ...) fournis par la cible de l'attaque. Une telle attaque est générée en saturant intentionnellement ses ressources en envoyant depuis un ordinateur une très grande quantité de paquets réseau vers la cible de l'attaque. Une variante distribuée (*DDoS*, pour *Distributed DoS*) consiste à envoyer ces paquets depuis une multitude d'ordinateurs contrôlés plus ou moins légitimement par l'attaquant. Pour un opérateur de réseau, il peut être important de détecter une telle attaque et de localiser les machines victimes afin de pouvoir prendre les mesures adéquates.

De nombreuses solutions de détection d'intrusion ont été proposées, par exemple Snort [Roesch (1999)] ou Bro [Paxson (1999)]. Ces logiciels utilisent des méthodes comparant le trafic analysé à des attaques déjà rencontrées et dont les signatures sont répertoriées dans une base de données.

Une autre catégorie de méthodes consiste à utiliser une approche statistique pour détecter les anomalies réseaux, celles-ci ne nécessitant pas une base de données d'attaques. Dans ce cas, la détection d'anomalies se fait via un test de détection de ruptures qui est une problématique classique en statistique. En effet, les attaques de type déni de service sont connues pour provoquer des changements abrupts dans le trafic réseau.

Une méthode statistique couramment utilisée en détection d'anomalies réseau est l'algorithme CUSUM [Basseville et Nikiforov (1993)]. Cet algorithme a été, entre autres, utilisé

par Siris et Papagalou (2006) pour la détection d'attaques de type *TCP/SYN flooding*, qui est un exemple d'attaque de type *DoS*. Celle-ci exploite les caractéristiques du protocole TCP (*Transmission Control Protocol*) en engorgeant de paquets de synchronisation (SYN) la machine destination attaquée, qui doit tenir à jour une table de demandes de connexion en attente d'un message d'acquittement ACK (*ACKnowledgement*) de la part de la machine source. Les ressources de la machine et la taille de la table étant limitées, l'attaque peut conduire à une saturation et à une interruption du service fourni par la machine. Les machines victimes d'une attaque de type *TCP/SYN flooding* pourraient ainsi être détectées en appliquant un test de détection de ruptures aux séries temporelles correspondant au nombre de paquets SYN reçus par chaque machine destination présente dans le trafic.

Ces méthodes citées précédemment sont dites centralisées ; les données sont collectées en plusieurs emplacements d'un réseau, mais elles sont toutes transmises sans traitement à un unique point qui effectue l'analyse des données. La quantité de données circulant dans un réseau pouvant être immense, cette centralisation peut poser plusieurs problèmes. D'une part, la transmission des données des capteurs vers le point central est non négligeable par rapport au trafic utile de données. D'autre part, la puissance de calcul nécessaire pour analyser cette masse d'information peut être très coûteuse. Des méthodes permettant de diminuer les échanges réseau et le temps de calcul sont donc nécessaires à partir du moment où le réseau atteint une certaine échelle.

Une première étape consiste donc à effectuer une réduction de la quantité de données à traiter. Une méthode élémentaire consiste à effectuer un échantillonage déterministe : seul un paquet sur N est retenu pour analyse. Cela permet de réduire de manière drastique la quantité de données mais au sacrifice des performances des algorithmes de détection utilisés en aval. Une autre méthode, proposée par Krishnamurthy *et al.* (2003), consiste à constituer un résumé ou agrégat des flux de données en effectuant une projection aléatoire appelée *sketch*. Cette idée a été mise en œuvre dans Li *et al.* (2006). On peut aussi citer une méthode de réduction de la dimension des données en utilisant une technique basée sur l'analyse en composante principales (ACP), voir Lakhina *et al.* (2004). Enfin, une approche basée sur un filtrage (ici par records) des données, comme dans Lévy-Leduc et Roueff (2009), peut être exploitée.

Ces méthodes restent malgré tout centralisées, c'est pourquoi on peut proposer une stratégie de décentralisation en confiant aux sondes, qui jusqu'à présent se contentaient d'effectuer les mesures, une partie du traitement et des calculs. Ainsi la capacité de calcul n'est plus limitée à celle de l'unique point central et on peut réduire la quantité de données transmises. Par ailleurs, en exploitant l'information de localisation des sondes et la topologie du réseau, les éventuelles anomalies et leurs sources peuvent être identifiées plus facilement.

Nous nous sommes jusqu'à présent intéressés à une approche que l'on peut qualifier de « semi-décentralisée ». Dans cette approche, les sondes filtrent l'information localement et envoient cette information filtrée à un collecteur central qui prend la décision finale concernant la présence éventuelle d'une anomalie. Par exemple, Huang *et al.* (2007) décentralisent la méthode utilisant l'ACP de Lakhina *et al.* (2004) : les sondes n'envoient que les données déviant suffisamment des informations connues au collecteur central.

Dans la partie 2, le *TopRank décentralisé* est présenté ; c'est une première méthode de

« semi-décentralisation »de l'algorithme TopRank de Lévy-Leduc et Roueff (2009) qui permet de faire de la détection d'anomalies en temps réel à partir de données de trafic (au niveau flots) à l'aide d'un test de rang non-paramétrique.

Alors que l'algorithme du TopRank décentralisé n'utilise pas la structure de dépendance des sondes, on propose dans la partie 3 un test de détection de changement multivarié qui exploite notamment la dépendance entre les données recueillies par les différentes sondes.

2 TopRank décentralisé

Le *TopRank distribué*, que nous avons proposé dans Lung-Yut-Fong *et al.* (2009c), utilise une approche rétrospective : les données sont traitées dans une fenêtre d'observation divisée en P segments, chacun de longueur Δ (typiquement, $P = 60$ et $\Delta = 1$ seconde). Les données traitées sont ensuite effacées après leur traitement à la fin de chaque fenêtre d'observation. Une première analyse est d'abord effectuée localement dans les sondes, puis une synthèse est effectuée au sein du collecteur central. Les K sondes dont on dispose sont notées dans la suite : $M_1, \ldots, M_K$.

On note $N_i^{(k)}(t)$ le nombre de paquets envoyés à la machine d'adresse IP i vus par le moniteur M_k dans l'intervalle t de longueur Δ de la fenêtre d'observation. Par exemple, pour détecter une attaque d'engorgement par paquets TCP/SYN, $(N_i^{(k)}(t))_{1 \leq t \leq P}$ représente la série temporelle du nombre de paquets SYN reçus par l'adresse IP i.

2.1 Traitement local dans les sondes

Chaque sonde M_k effectue un premier traitement sur les données qu'elle recueille ; il se déroule en quatre étapes et correspond à l'algorithme *TopRank* détaillé dans Lévy-Leduc et Roueff (2009) dont on rappelle ici le principe (pour alléger les notations, on omet l'exposant $^{(k)}$ désignant le numéro de moniteur dans les nombres de paquets) :

Filtrage par records Dans chaque sous-intervalle indexé par $t \in \{1, \ldots, P\}$ de longueur Δ secondes de la fenêtre d'observation, on garde les adresses IP i des M plus grands $N_i(t)$, que l'on note $i_1(t), \ldots, i_M(t)$ et tels que : $N_{i_1(t)}(t) \geq N_{i_2(t)}(t) \geq \cdots \geq N_{i_M(t)}(t)$. On note $\mathcal{T}(t) = \{i_1(t), \ldots, i_M(t)\}$ ce classement d'adresses IP. Notons que l'on ne garde pour la suite que les éléments de $\mathcal{T}(t)$ ainsi que les valeurs correspondantes $\{N_i(t),\ i \in \mathcal{T}(t),\ t = 1, \ldots, P\}$.

Création des séries temporelles censurées Pour chaque adresse IP i sélectionnée à l'étape précédente ($i \in \bigcup_{t=1}^{P} \mathcal{T}(t)$), on construit la série temporelle $(X_i(t))_{1 \leq t \leq P}$. Cette série est censurée, puisqu'il se peut qu'à un instant t, i ne soit pas dans l'ensemble $\mathcal{T}(t)$ et que l'on ne dispose donc plus de la valeur $N_i(t)$ correspondante. Dans ce cas, $X_i(t)$ prend la valeur $N_{i_M(t)}(t)$. On note par ailleurs $x_i(t)$ l'état de censure de $X_i(t)$:

$$(X_i(t), x_i(t)) = \begin{cases} (N_i(t), 1), & \text{si } i \in \mathcal{T}_M(t) \\ (\min\limits_{j \in \mathcal{T}_M(t)} N_j(t), 0), & \text{sinon.} \end{cases}$$

On définit ensuite les bornes supérieure $\overline{X_i(t)} = X_i(t)$ et inférieure $\underline{X_i(t)} = X_i(t)x_i(t)$ de $X_i(t)$.

Test de détection de changement On utilise un test non-paramétrique de détection de changement pour données censurées inspiré de celui proposé par Gombay et Liu (2000). C'est un test de rang non-paramétrique utilisant une fonction de score (que l'on note A) qui généralise le test de rang de Wilcoxon aux données censurées.

On veut tester, pour chaque i :

(H_0) : « $(N_i(t))_{1 \leq t \leq P}$ sont des variables aléatoires i.i.d »contre

(H_1) : « il existe un entier r tel que $(N_i(1), \ldots, N_i(r))$ et $(N_i(r+1), \ldots, N_i(P))$ ont une distribution différente. »

Pour tout $s,\ t \in 1, \ldots, P$, on définit les quantités suivantes :

$$A_{s,t} = \mathbb{1}\left(\underline{X_i(s)} > \overline{X_i(t)}\right) - \mathbb{1}\left(\underline{X_i(t)} > \overline{X_i(s)}\right) \ ; \tag{1}$$

$$U_s = \sum_{t=1}^{P} A_{s,t},\ \ s = 1, \ldots, P; \tag{2}$$

$$S_t = \left(\sum_{s=1}^{t} U_s\right) / \left(\sum_{s=1}^{P} U_s^2\right)^{1/2},\ \ t = 1, \ldots, P\ . \tag{3}$$

On utilise alors la quantité

$$W_P = \max_{1 \leq t \leq P} |S_t| \tag{4}$$

comme statistique de test.

Sous (H_0), il est montré par exemple dans le Théorème 1 de Gombay et Liu (2000) que sous certaines conditions,

$$W_P \xrightarrow{\mathcal{D}} B^\star = \sup_{0<t<1} |B(t)|,\ \text{quand } P \to \infty, \tag{5}$$

où $\{B(t),\ t \in [0,1]\}$ est un pont Brownien et $\mathcal{D}$ désigne la convergence en loi.

On associe à cette statistique de test une p-valeur : $Pval(W_P)$, où, voir par exemple Billingsley (1968),

$$Pval(b) = 2\sum_{j \geq 1} (-1)^{j-1} e^{-2j^2 b^2}\ ,\ \text{pour tout } b > 0. \tag{6}$$

Sélection des données à envoyer au collecteur central La sonde M_k choisit les d séries temporelles censurées ayant les plus petites p-valeurs et les transmet au collecteur central.

2.2 Agrégation dans le collecteur central

Le collecteur central construit alors les séries temporelles suivantes à partir des données envoyées par les sondes :

$$\underline{Z_i(t)} = \sum_{k=1}^{K} \underline{X_i^{(k)}(t)} \quad \text{et} \quad \overline{Z_i(t)} = \sum_{k=1}^{K} \overline{X_i^{(k)}(t)} \ ,$$

où $(\underline{X_i^{(k)}(t)}, t = 1, \ldots, P)$ et $(\overline{X_i^{(k)}(t)}, t = 1, \ldots, P)$ sont les séries temporelles associées à l'adresse IP i créées dans la sonde M_k. Le même test de détection de ruptures que celui utilisé dans les sondes est alors appliqué aux séries temporelles $\underline{Z_i}$ et $\overline{Z_i}$, l'adresse IP i étant alors déclarée comme attaquée au niveau $\alpha \in (0, 1)$ lorsque : $Pval(W_P) < \alpha$.

2.3 Résultats

Afin d'évaluer les performances de l'algorithme proposé, plusieurs séries de simulations ont été effectuées. Dans un premier temps une trace de données réelles provenant d'un fournisseur d'accès internet a été utilisée. Celle-ci contenant des données centralisées, elles ont été réparties sur $K = 15$ sondes virtuelles. Cette trace contenait des attaques de type SYN Flooding vers 4 adresses IP identifiées. Le TopRank distribué obtient de bons résultats comparé à une méthode plus simple utilisant la correction de Bonferroni, et les performances sont à peine dégradées par rapport au TopRank non décentralisé, pour une économie notable de trafic échangé, voir figure 1.

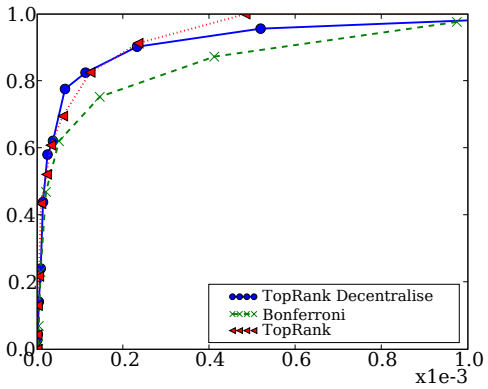

Fig. 1 – Courbes COR pour TopRank distribué ('○'), Bonferroni ('x'), TopRank ('◁').

Par ailleurs, du trafic synthétique a été généré (figure 2) et injecté dans une topologie réseau générée dans le but de refléter un réseau d'opérateur (aux caractéristiques proches du réseau académique américain Abilene). Les performances suivent les mêmes tendances qu'avec les expériences précédentes.

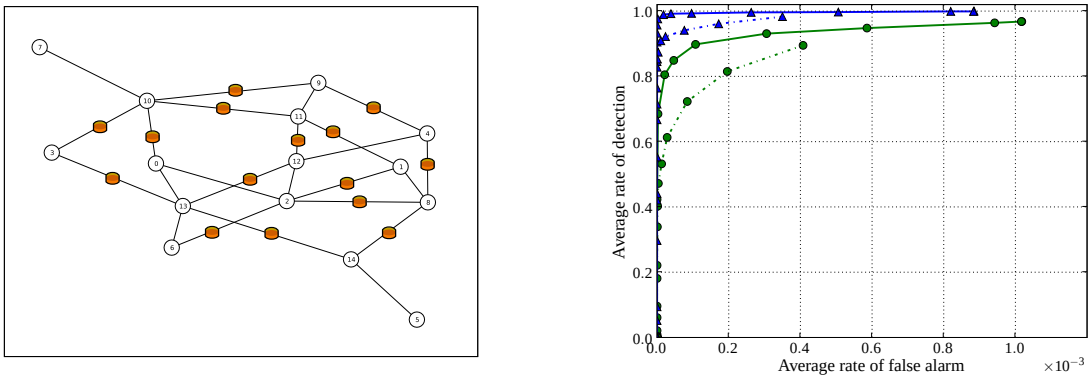

Fig. 2 – Gauche : topologie réseau dans laquelle le trafic a été injecté. Droite : Courbes COR pour le TopRank distribué (lignes pleines) and *Bonferroni* (pointillés), pour des intensités d'attaques $\eta = 1.2$ ("•"), 1.5 ("△").

3 Tests d'homogénéité et de détection de changement pour données multivariées

Un test d'homogénéité et un test de détection de ruptures qui en est dérivé est proposé ici. Contrairement au test de détection de rupture utilisé dans le TopRank distribué, pour lequel des statistiques sur des données unidimensionnelles étaient calculées pour chacune des séries temporelles vues par les différentes sondes sans exploiter les relations entre les données, on propose ici un test pour données multi-variées qui prend en compte ces structures de dépendance.

3.1 Test d'homogénéité

De nombreuses méthodes non-paramétriques existent dans la littérature pour le test d'homogénéité à deux échantillons dans un cadre multi-dimensionnel. Citons par exemple Schilling (1986) ou Henze (1988) qui proposent un test reposant sur un algorithme utilisant un arbre des plus proches voisins. Plus récemment, des méthodes à noyau comme le *Maximum Mean Discrepancy* [Gretton *et al.* (2008)] ou l'analyse discriminante de Fisher dans un espace de Hilbert à noyau reproduisant (KFDA) [Harchaoui *et al.* (2007)] ont été proposées.

Notre méthode utilise un test inspiré de l'approche proposée par Wei et Lachin (1984) ; celle-ci a pour avantage d'être adaptée à des données censurées, et permet donc un filtrage comme dans le TopRank. Comme pour l'algorithme du TopRank distribué, une statistique basée sur le test de rang de Wilcoxon est calculée dans chaque dimension. Cependant, on fait intervenir dans la statistique finale une matrice de covariance des statistiques marginales, ce qui permet de prendre en compte une éventuelle relation de dépendance entre les différentes sondes et finalement d'améliorer la puissance de détection.

Soit $\{\mathbf{X}(t) = (X^{(1)}(t), \ldots, X^{(K)}(t))', t = 1, \ldots, n\}$ le vecteur de données de dimension K (où K est le nombre de sondes), où t est l'indice temporel et la quantité en exposant le numéro de sonde. On veut tester, lorsque l est connu :

(H_0) : « $(\underline{\mathbf{X}}(t), \overline{\mathbf{X}}(t))_{1 \leq t \leq n}$ sont des variables aléatoires i.i.d »contre

(H_1) : « $\big((\underline{\mathbf{X}}(1), \overline{\mathbf{X}}(1)), \ldots, \ (\underline{\mathbf{X}}(l), \overline{\mathbf{X}}(l))\big)$ et $\big((\underline{\mathbf{X}}(l+1), \overline{\mathbf{X}}(l+1)), \ldots, (\underline{\mathbf{X}}(n), \overline{\mathbf{X}}(n))\big)$ ont une distribution différente. »

Pour prendre en compte les censures introduites par une étape de filtrage identique à celle utilisée dans l'algorithme du TopRank Distribué, on a introduit de même les quantités $\underline{X}$ et $\overline{X}$.

Définissons alors :

$$U_n^{(k)}(l) = \frac{1}{n^{3/2}} \sum_{i=1}^{l} \sum_{j=l+1}^{n} \Big(\mathbb{1}(\underline{X}^{(k)}(i) \geq \overline{X}^{(k)}(j)) - \mathbb{1}(\overline{X}^{(k)}(j) \geq \underline{X}^{(k)}(i)) \Big) \ ,$$

$$= \frac{1}{n^{3/2}} \sum_{i=1}^{l} \sum_{j=l+1}^{n} h((\underline{X}^{(k)}(i), \overline{X}^{(k)}(i)), (\underline{X}^{(k)}(j), \overline{X}^{(k)}(j))) \ , \quad (7)$$

où l'on a défini $h((x_1, y_1), (x_2, y_2)) = \mathbb{1}(x_1 \geq y_2) - \mathbb{1}(x_2 \geq y_1)$,

$U_n^{(k)}$ est à une constante de normalisation près similaire à l'expression (3) de TopRank. Elle est calculée pour tout moniteur $k \in \{1 \ldots K\}$. Soit Σ la matrice de covariance définie pour tout $k, k' = 1, \ldots, K$ par :

$$\Sigma_{k,k'} = t_0 \cdot (1 - t_0) \cdot \mathrm{Cov} \Big(F^{(k)}(\underline{X}^{(k)}(1) - \overline{G}^{(k)}(\overline{X}^{(k)}(1)); F^{(k')}(\underline{X}^{(k')}(1)) - \overline{G}^{(k')}(\overline{X}^{(k')}(1)) \Big) \ , \quad (8)$$

où t_0 est tel que $l = \lfloor t_0 \cdot n \rfloor$, F (resp. G) est la fonction de répartition de $\overline{X}(1), \ldots, \overline{X}(n)$ (resp. $\underline{X}(1), \ldots, \underline{X}(n)$) et $\overline{G} = 1 - G$.

Si on pose $\mathbf{U}_n(t_0) = (U_{n,1}(t_0), \ldots, U_{n,k}(t_o))'$, alors

$$S(t_0) = \mathbf{U}_n(t_0) \cdot \Sigma^{-1} \cdot \mathbf{U}_n(t_0)' \quad (9)$$

est la statistique de test que l'on propose. Sous l'hypothèse H_0, nous avons montré que $S(t_0)$ converge en loi vers un χ^2 à K degrés de liberté lorsque $n \to \infty$.

3.2 Test de rupture

Avec les mêmes notations que précédemment, on veut cette fois ci tester :

(H_0) : « $(\underline{\mathbf{X}}(t), \overline{\mathbf{X}}(t))_{1 \leq t \leq n}$ sont des variables aléatoires i.i.d »contre

(H_1) : « il existe un entier r inconnu tel que
$\big((\underline{\mathbf{X}}(1), \overline{\mathbf{X}}(1)), \ldots, \ (\underline{\mathbf{X}}(r), \overline{\mathbf{X}}(r))\big)$ et $\big((\underline{\mathbf{X}}(r+1), \overline{\mathbf{X}}(r+1)), \ldots, (\underline{\mathbf{X}}(n), \overline{\mathbf{X}}(n))\big)$ ont une distribution différente. »

La statistique de test que nous proposons devient alors :

$$W_n = \sup_{0<t<1} |\mathbf{U}_n(\lfloor t \cdot n \rfloor) \cdot \Gamma^{-1} \cdot \mathbf{U}_n(\lfloor t \cdot n \rfloor)'| \ , \tag{10}$$

où

$$\Gamma_{k,k'} = \frac{3 \cdot \Sigma_{k,k'}}{t_0(1-t_0)} \ , 1 \leq k, k' \leq K$$

est cette fois le terme général de la matrice Γ.

Nous avons démontré que, sous H_0, lorsque $n \to \infty$,

$$W_n \xrightarrow{\mathcal{D}} \frac{1}{3} \sup_{0<t<1} \sum_{i=1}^{K} B_i^2(t) \ , \tag{11}$$

où $\{B_i(t), \ t \in [0,1]\}$ sont des ponts Browniens indépendants. On peut, en utilisant le résultat suivant [Kiefer (1959)], calculer les p-valeurs $Pval(W_n)$ associées à W_n, avec :

$$Pval(b) = \frac{4}{\Gamma(\frac{K}{2}) 2^{\frac{K}{2}} b^{\frac{K}{2}}} \sum_{m=1}^{\infty} \frac{(\gamma_{(K-2)/2,m})^{K-2} \exp[-(\gamma_{(K-2)/2,m})^2]/2b}{[J_{K/2}(\gamma_{(K-2)/2,m})]^2} \ , \tag{12}$$

où J_ν est une fonction de Bessel de première espèce et $\gamma_{\nu,m}$ est le m^e zéro positif de J_ν.

4 Perspectives

À court terme, les efforts se porteront sur l'évaluation et la validation expérimentale des tests d'homogénéité et de changement présentés dans la partie 3 Les algorithmes seront testés dans des simulations sur des données synthétiques ou des datasets courants. Selon disponibilité, on effectuera aussi une évaluation sur des jeux de données réelles. Les résultats seront soumis à des conférences avec actes (NIPS, ECML).

Dans un horizon plus lointain, on envisage une étude sur une décentralisation plus poussée des algorithmes de détection de changement. Dans ce cas de décentralisation, les sondes adoptent une stratégie collaborative. On doit alors explorer les différentes possibilités de collaboration, leur robustesse en cas de défaillance, la nature et la quantité des échanges entre les capteurs qui sont liés aux contraintes de communication qui seraient imposées...

5 Publications

Les travaux décrits ci-dessus ont fait l'objet de communications scientifiques :

- Lung-Yut-Fong *et al.* (2009a) ;
- Lung-Yut-Fong *et al.* (2009c) ;
- Lung-Yut-Fong *et al.* (2009b) (article de journal en cours de revue) ;
- Exposé oral à l'atelier *Temporal Segmentation* de la conférence NIPS (*Neural Information Processing Systems*), Vancouver, décembre 2009. http://www.harchaoui.eu/zaid/workshops/nips09/index.html.

Références

Basseville, M. et Nikiforov, I. V. (1993). *Detection of Abrupt Changes : Theory and Applications*. Prentice-Hall.

Billingsley, P. (1968). *Convergence of probability measures*. Wiley, New York.

Gombay, E. et Liu, S. (2000). A nonparametric test for change in randomly censored data. *The Canadian Journal of Statistics*, 28(1):113–121.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. et Smola, A. (2008). A kernel method for the two-sample-problem. *Journal of Machine Learning Research*, 1:1–10.

Harchaoui, Z., Bach, F. et Moulines, E. (2007). Testing for homogeneity with kernel fisher discriminant analysis. *In NIPS*, Vancouver.

Henze, N. (1988). A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics*, pages 772–783.

Huang, L., Nguyen, X., Garofalakis, M., Jordan, M. I., Joseph, A. et Taft, N. (2007). In-network pca and anomaly detection. *In* Schölkopf, B., Platt, J. et Hoffman, T., éditeurs : *Advances in Neural Information Processing Systems 19*, pages 617–624. MIT Press, Cambridge, MA.

Kiefer, J. (1959). K-sample analogues of the Kolmogorov-Smirnov and Cramer-v. Mises tests. *The Annals of Mathematical Statistics*, 30(2):420–447.

Krishnamurthy, B., Sen, S., Zhang, Y. et Chen, Y. (2003). Sketch-based change detection : methods, evaluation, and applications. *In IMC '03 : Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 234–247, New York, NY, USA. ACM.

Lakhina, A., Crovella, M. et Diot, C. (2004). Diagnosing network-wide traffic anomalies. *In SIGCOMM '04 : Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 219–230, New York, NY, USA. ACM.

Lévy-Leduc, C. et Roueff, F. (2009). Detection and localization of change-points in high-dimensional network traffic data. *Annals of Applied Statistics*, 3(2):637–662.

Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G. et Lakhina, A. (2006). Detection and identification of network anomalies using sketch subspaces. pages 147–152. Proceedings of SIGCOMM.

Lung-Yut-Fong, A., Cappé, O., Lévy-Leduc, C. et Roueff, F. (2009a). Détection et localisation décentralisées d'anomalies dans le trafic internet. *In GRETSI*, Dijon, France.

Lung-Yut-Fong, A., Lévy-Leduc, C. et Cappé, O. (2009b). Distributed detecion/localization of change-points in high-dimensional network traffic data. *submitted*.

Lung-Yut-Fong, A., Lévy-Leduc, C. et Cappé, O. (2009c). Distributed detection/localization of network anomalies using rank tests. *In IEEE Workshop on Statistical Signal Procesing*, Cardiff, Wales, UK.

Paxson, V. (1999). Bro : A system for detecting network intruders in real-time. *Computer Network*, 31(23–24):2435–2463.

Roesch, M. (1999). Snort : Lightweight intrusion detection for networks. pages 229–238. Proceedings of LISA.

Schilling, M. (1986). Multivariate two-sample tests based on nearest neighbors. *Journal of the American Statistical Association*, pages 799–806.

Siris, V. A. et Papagalou, F. (2006). Application of anomaly detection algorithms for detecting syn flooding attacks. *Computer Communications*, 29(9):1433 – 1442. ICON 2004 - 12th IEEE International Conference on Network 2004.

Wei, L. et Lachin, J. (1984). Two-sample asymptotically distribution-free tests for incomplete multivariate observations. *Journal of the American Statistical Association*, 79(387):653–661.