PV210 Bezpečnostní analýza síťového provozu

Pokročilé metody zpracovávající síťové toky

Mgr. Tomáš Jirsík

12. 11. 2014

Jednoduché metody zpracovávající informace o tocích I: shrnutí

Časové řady

Shlukování

Další metody

Shrnutí

- Agregace provozu v podobě síťových toků je dobrým stavebním kamenem jednoduchých metod.
- Metody zpracovávající záznamy o tocích jsou v porovnání s inspekcí paketů velmi rychlé.
- I jednoduché metody (např. detekce TCP SYN skenů) jsou v praxi velmi užitečné.

Pokročilé metody zpracovávající síťové toky

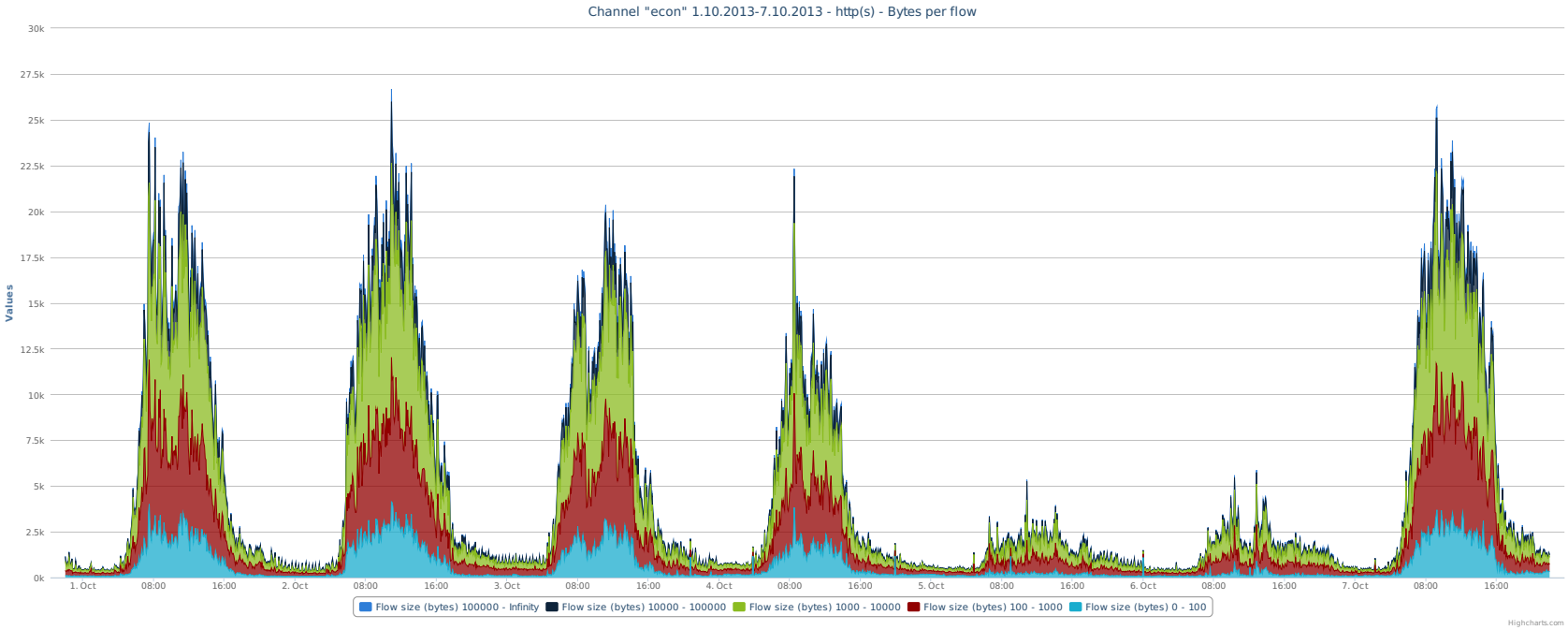
- Časové řady
  - Klouzavý průměr
  - Holt-Wintersova metoda
  - Analýza hlavních komponent
- Shlukování a hledání odlehlých pozorování
  - K-means
  - Local Outlier Factor
- Další metody

Časové řady

Využití v síťovém provozu

- Cíl - automatizovat detekci anomálií (pohled na graf a následné nalezení špiček).
- Hlavní myšlenka využití:
  - porovnání predikce a reálně naměřené hodnoty
  - pokud se hodnoty liší o více než je *“únosná míra”*, je detekována anomálie

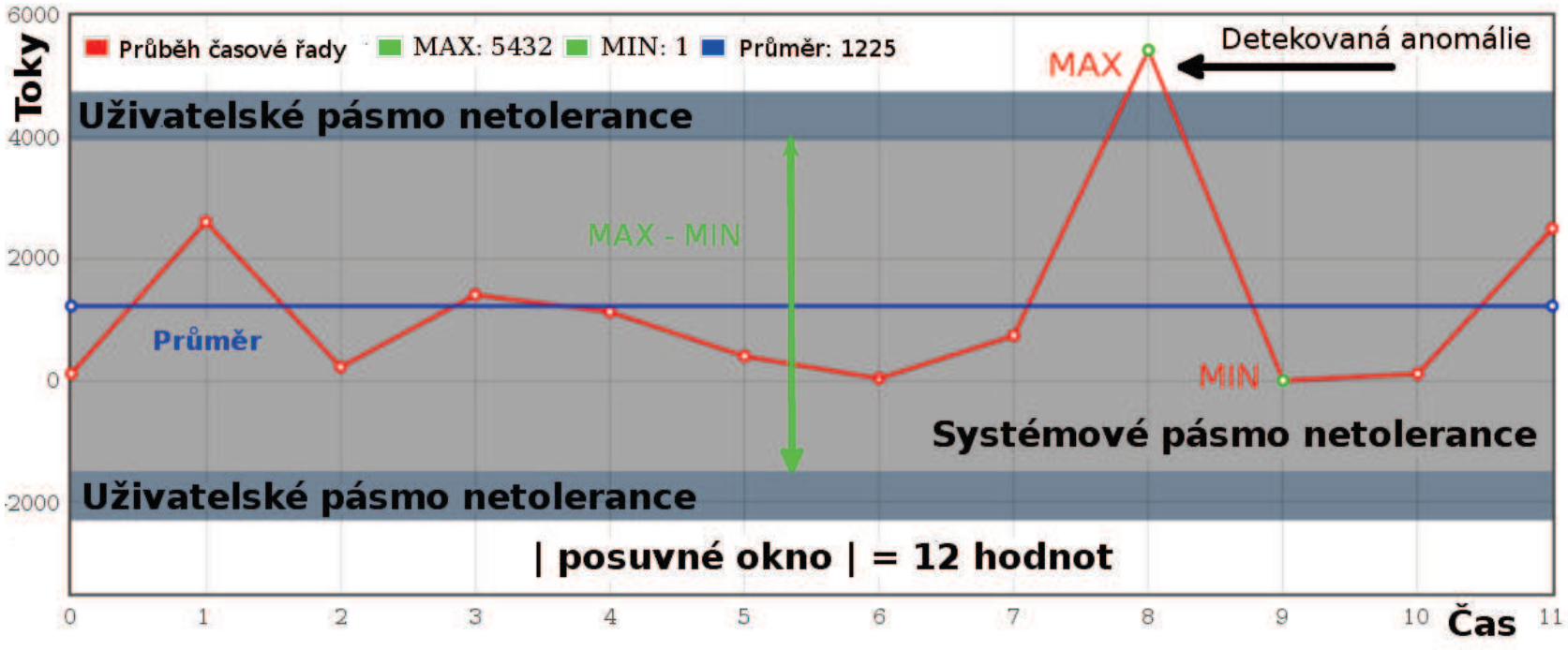
Časové řady


Channel "econ" 1.10.2013-7.10.2013 - http(s) - Bytes per flow
Values
30k
27.5k
25k
22.5k
20k
17.5k
15k
12.5k
10k
7.5k
5k
2.5k
0k
1. Oct
08:00
16:00
2. Oct
08:00
16:00
3. Oct
08:00
16:00
4. Oct
08:00
16:00
5. Oct
08:00
16:00
6. Oct
08:00
16:00
7. Oct
08:00
16:00
Flow size (bytes) 100000 - Infinity
Flow size (bytes) 10000 - 100000
Flow size (bytes) 1000 - 10000
Flow size (bytes) 100 - 1000
Flow size (bytes) 0 - 100
Highcharts.com

Časové řady

Klouzavý průměr

Klouzavý průměr



Časové řady

Holt-Wintersova metoda

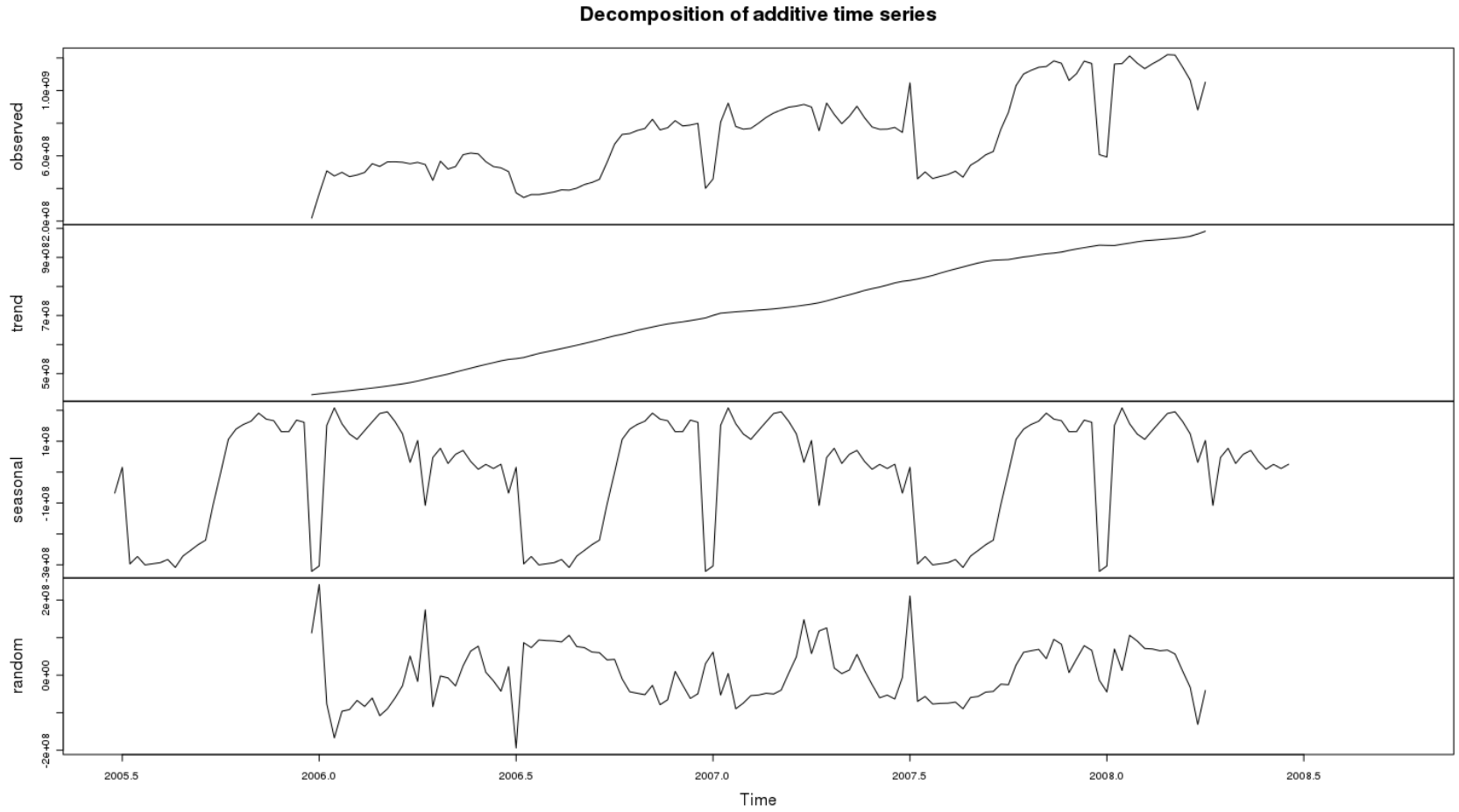
Úvod

- V literatuře také jako trojité exponenciální vyhlazování (*triple exponential smoothing*).
- Navržena v roce 1957 Holtem, v roce 1960 vylepšena Wintersem.
- Původní využití **predikce *časové řady***.
- Časová řada je posloupnost pozorování jedné nebo více náhodných veličin uspořádaná v čase.

Předpoklady

- ekvidistantní časový interval.
- časová řada může být rozložena na tři komponenty: *základnu, lineární trend a sezónní trend*.

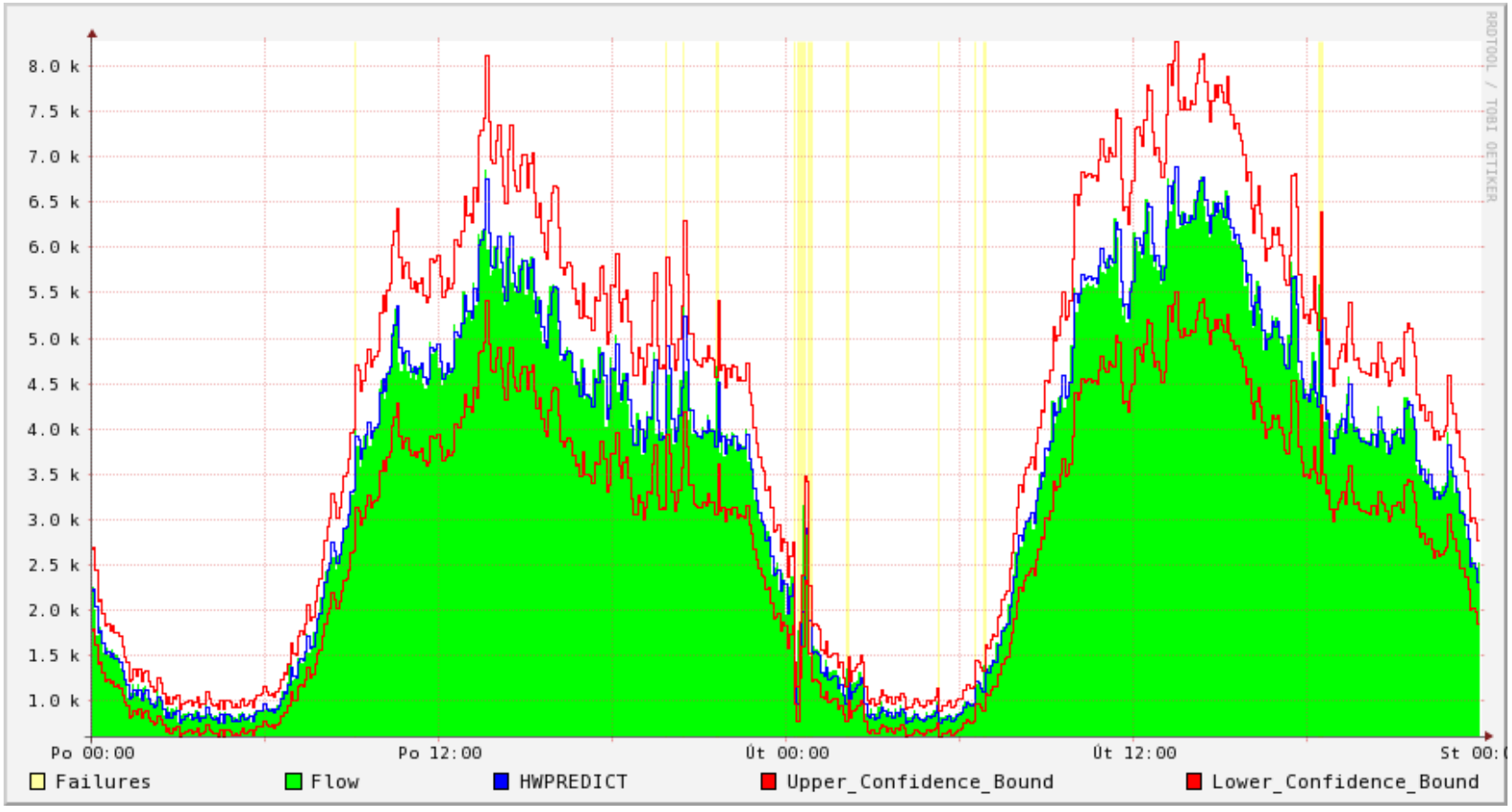
Předpoklady



Využití v síťovém provozu

- Metoda je vhodná i pro síťový provoz a toky.
- Časové řady objemových charakteristik mnohých služeb totiž vykazují toto chování (příklad):
  - **Trend**: postupné zvyšování požadavků na nějakou službu v čase.
  - **Sezónnost**: nejvíce požadavků se objevuje dopoledne, odpoledne méně a v noci obvykle minimum.
  - **Sezónní proměnlivost**: ve špičce je zaznamenána velká fluktuace počtu požadavků, kdežto v noci ne.
  - Postupný vývoj všech předcházejících složek v čase: vliv letního času na počty požadavků v průběhu dne.

Vizualizace předpovědi Holt-Wintersovou metodou



Holt-Wintersova metoda podrobně

- Mějme časovou řadu: $y_1 ... y_{t-1}, y_t, y_{t+1}$, t je aktuální čas.
- Předpovídaná hodnota v časové řadě se vypočítá takto:

$$y_{t+1} = a_t + b_t + c_{t+1-m}$$

- m označuje periodu sezónního trendu (např. počet pozorování za den).
- Základna:
  $a_t = \alpha(y_t - c_{t-m}) + (1 - \alpha)(a_{t-1} + b_{t-1})$
- Lineární trend (*sklon*):
  $b_t = \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1}$
- Sezónní trend:
  $c_t = \gamma(y_t - a_t) + (1 - \gamma)c_{t-m}$

Holt-Wintersova metoda podrobně – pokr.

- α, β, γ jsou parametry adaptace. $0 < \alpha, \beta, \gamma < 1$.
- Hodnoty blíže k 1 povedou k rychlejší adaptaci, hodnoty blíže k 0 k opaku (větší důraz bude kladen na historii).
- Důležité je vhodně zvolit tyto parametry a to je docela problematické.
- Pozor na *otrávení učení*

- Tuto „otrockou" práci může za nás dělat někdo jiný (až na volbu parametrů). Např. aplikace **R**, konkrétně funkce *HoltWinters()* (`http://www.r-project.org/` nebo v balících vaší oblíbené linuxové distribuce) nebo nástroj `rrdtool`.

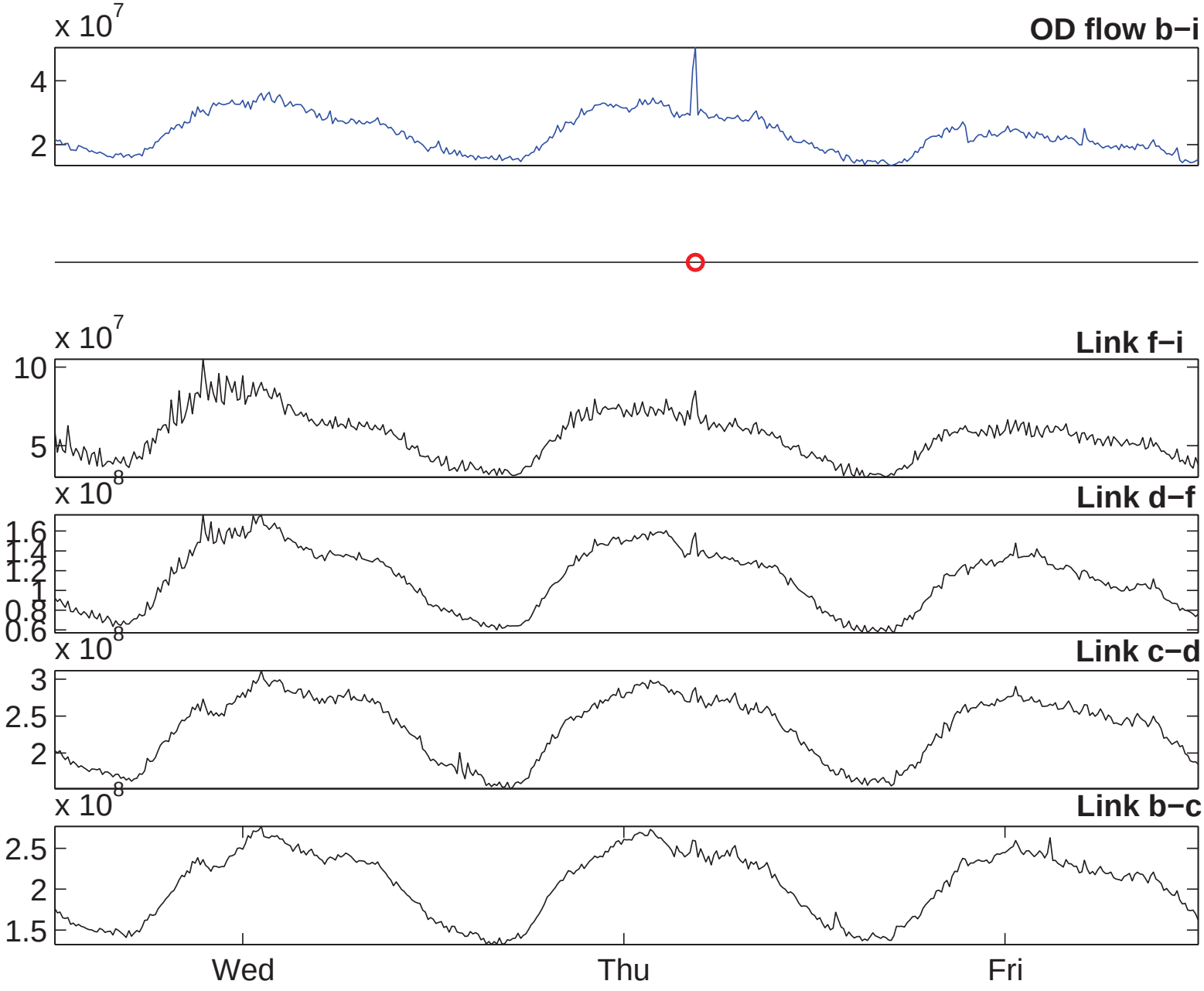
Časové řady

Analýza základních komponent
Principal component analysis (PCA)

Úvod do PCA

- Cíl: rozdělit skupinu vzájemně korelovaných pozorování na vzájemně nekorelované
- Využití ortogonální transformace
- První základní komponenta vysvětluje co nejvíce variability v datech, další obsahuje co nejvíce zbylé variability atd...
- Využití pro oddělení anomálního provozu od normálního.

Vizualizace PCA



Shluková analýza

- Vícerozměrná statistická metoda používaná ke klasifikaci objektů.
- Jednotky náležící do stejné skupiny jsou si podobnější než objekty z různých skupin.
- Je možné provádět jak na množině objektů, z nichž každý musí být popsán prostřednictvím stejného souboru znaků, které má smysl v dané množině sledovat, tak na množině znaků, které jsou charakterizovány prostřednictvím určitého souboru objektů, nositelů těchto znaků.

Shluková analýza

K-means

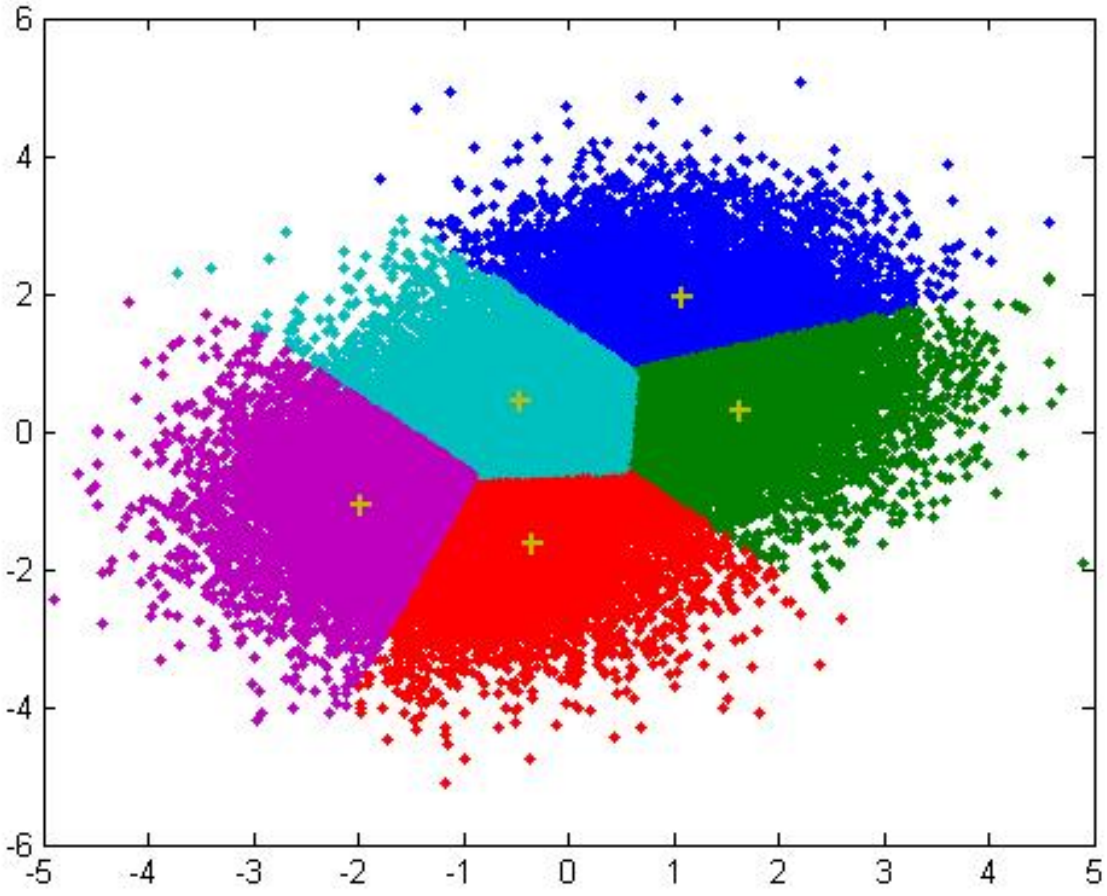
K-means

- Vícerozměrná statistická metoda používaná ke klasifikaci objektů.
- Využití ke klasifikaci provozu na normální a anomální
- Cíl: rozdělit pozorování na k skupin, tak aby minimalizovaly

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \| \tag{1}$$

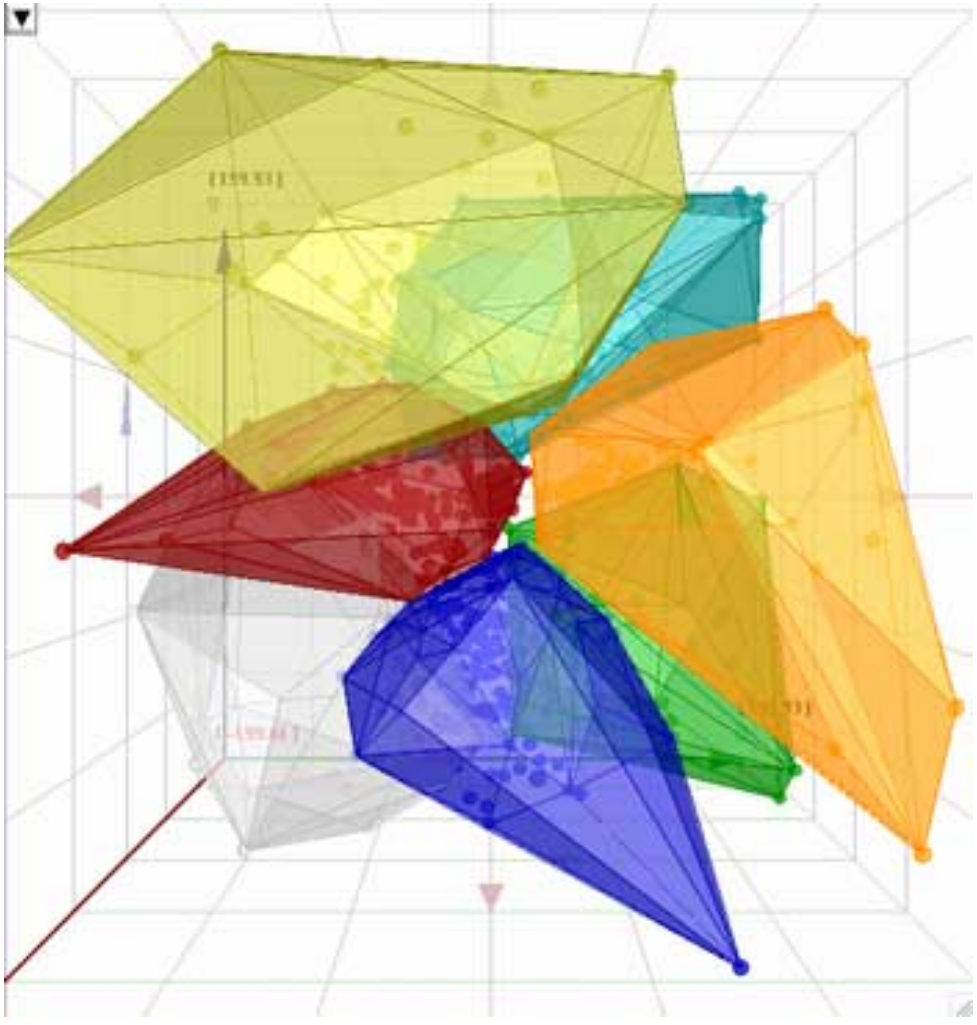
kde μ_i je průměr ve skupině s_i

K-means



https://www.youtube.com/watch?v=_aWzGGNrcic

K-means



K-means

Nevýhody metody:

- Rozdělí data do skupin, ale neurčí, zda je skupina anomální, či ne.
- Určit počet skupin, do kterých se mají data rozdělit.

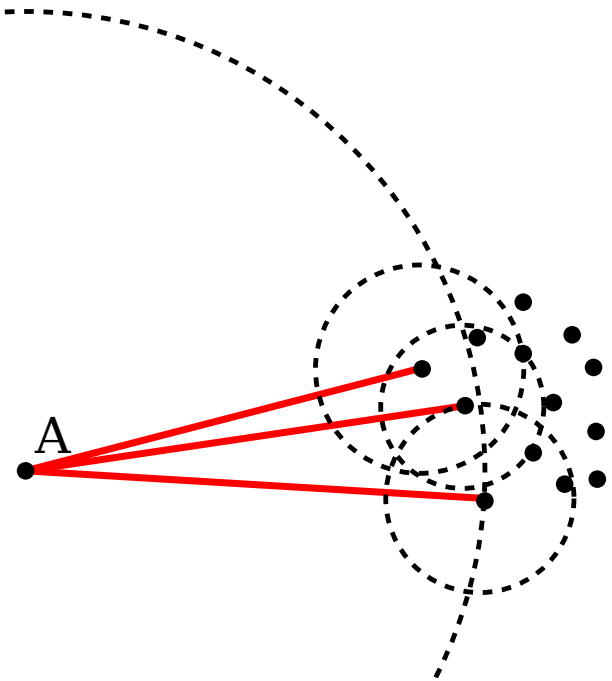
Shluková analýza

Local Outlier Factor (LOF)

Princip

Definice odlehlého pozorování:

- *An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.*[1]

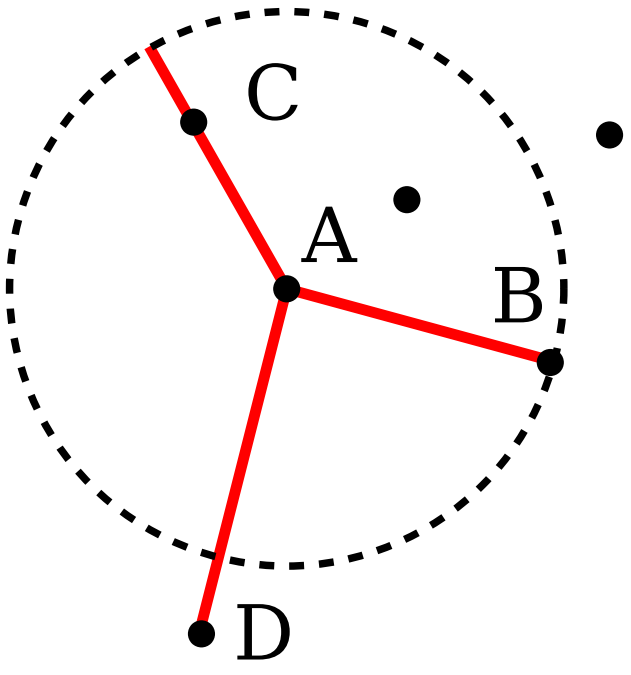


[1] *Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander LOF: Identifying Density-Based Local Outliers*

Výpočet

Dosažitelná vzdálenost:
(Reachability Distance)

$$RD_k(A, B) = \max\{\text{ k-distance}(B), d(A, B)\}$$



Výpočet

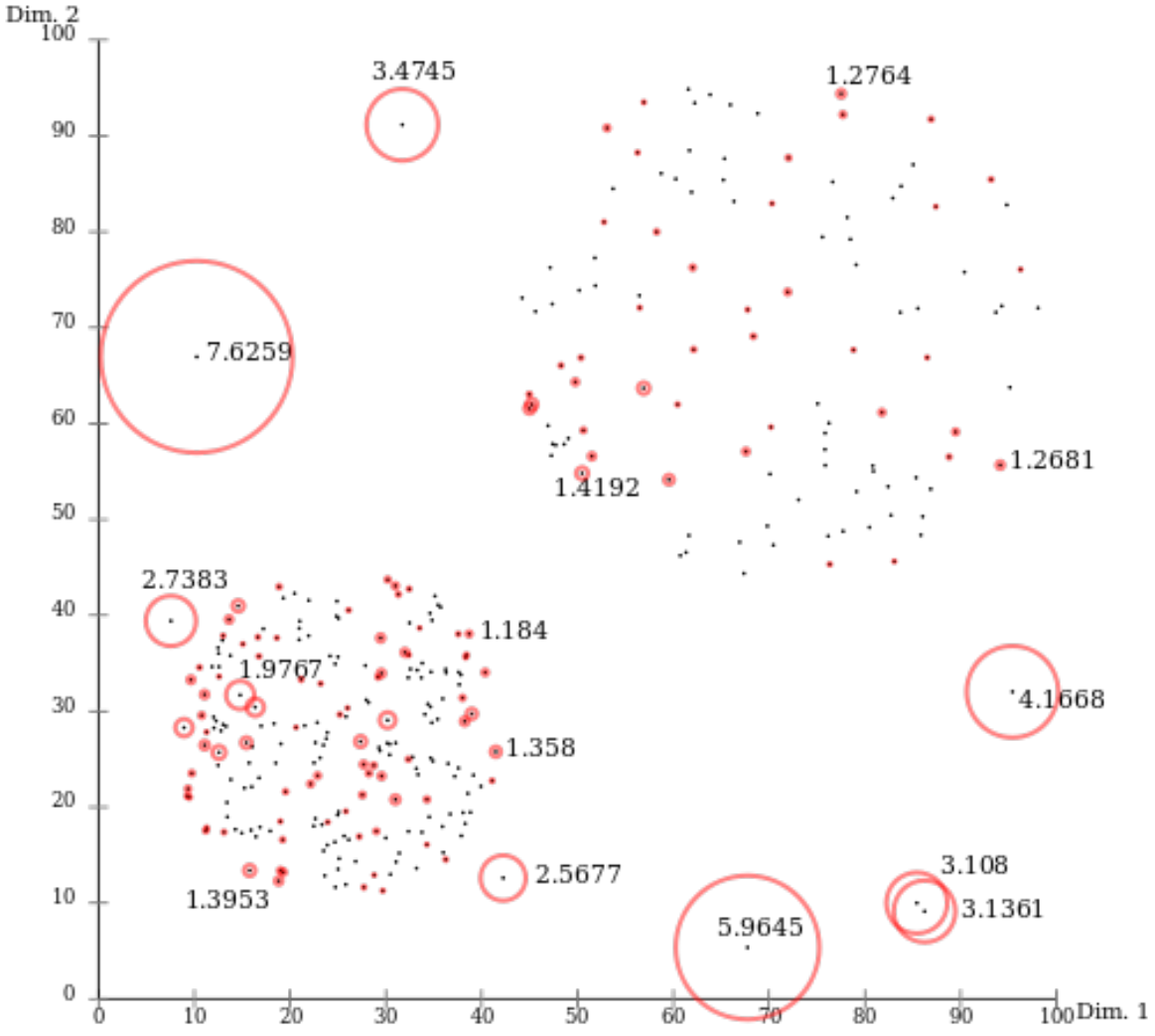
Místní hustota dosažitelnosti:
(Local Reachability Density)

$$lrd(A) = 1/\left(\frac{\sum_{B \in N_k(A)} RD_k(A,B)}{|N_k(A)|}\right)$$

Local Outlier Factor:

$$\mathrm{LOF}_k(A) = \frac{\sum_{B \in N_k(A)} \frac{lrd(B)}{lrd(A)}}{|N_k(A)|} = \frac{\sum_{B \in N_k(A)} lrd(B)}{|N_k(A)|} / lrd(A)$$

Proč použít Local Outlier Factor?



Využito v projektu MINDS

Minnesota Intrusion Detection System (MINDS)

- Navrženo pro velké sítě (Univerzita v Minnesotě, 2002).
- Pracuje s NetFlow daty v 10minutovém časovém okně.
- Nejprve je provedena **feature extraction** – pro každý tok jsou spočítány klíčové položky.
- Cílem detekce je najít spojení s „vyčnívajícími” (outliers) položkami = anomálie síťového provozu.
- Detekce vyžívá **Local Outlier Factor** (LOF).
- LOF stojí na předpokladu, že vlastnost „vyčnívání” není binární, že lze vyjádřit míru „vyčnívání“.
- *Local* – míra vyčnívání je počítána jen v uzavřeném sousedství objektu.

MINDS: extrakce položek

- Používají se dva typy položek.
- ***Time window**-based* jsou extrahovány z toků za posledních *t* sekund.
- Z principu není možné zachytit anomálie trvající déle než *t* sekund (např. pomalé skenování portů).
- ***Connection window**-based* z posledních *n* toků odchozích z různých zdrojů/příchozích na různé cíle.

MINDS: typy položek

Time window-based features:

- *count-dest*, number of flows to unique destination IP addresses inside the network in the last *t* seconds from the same source,
- *count-src*, number of flows from unique source IP addresses inside the network in the last *t* seconds to the same destination,
- *count-serv-src*, number of flows from the source IP to the same destination port in the last *t* seconds
- *count-serv-dest*, number of flows to the destination IP address using same source port in the last *t* seconds.

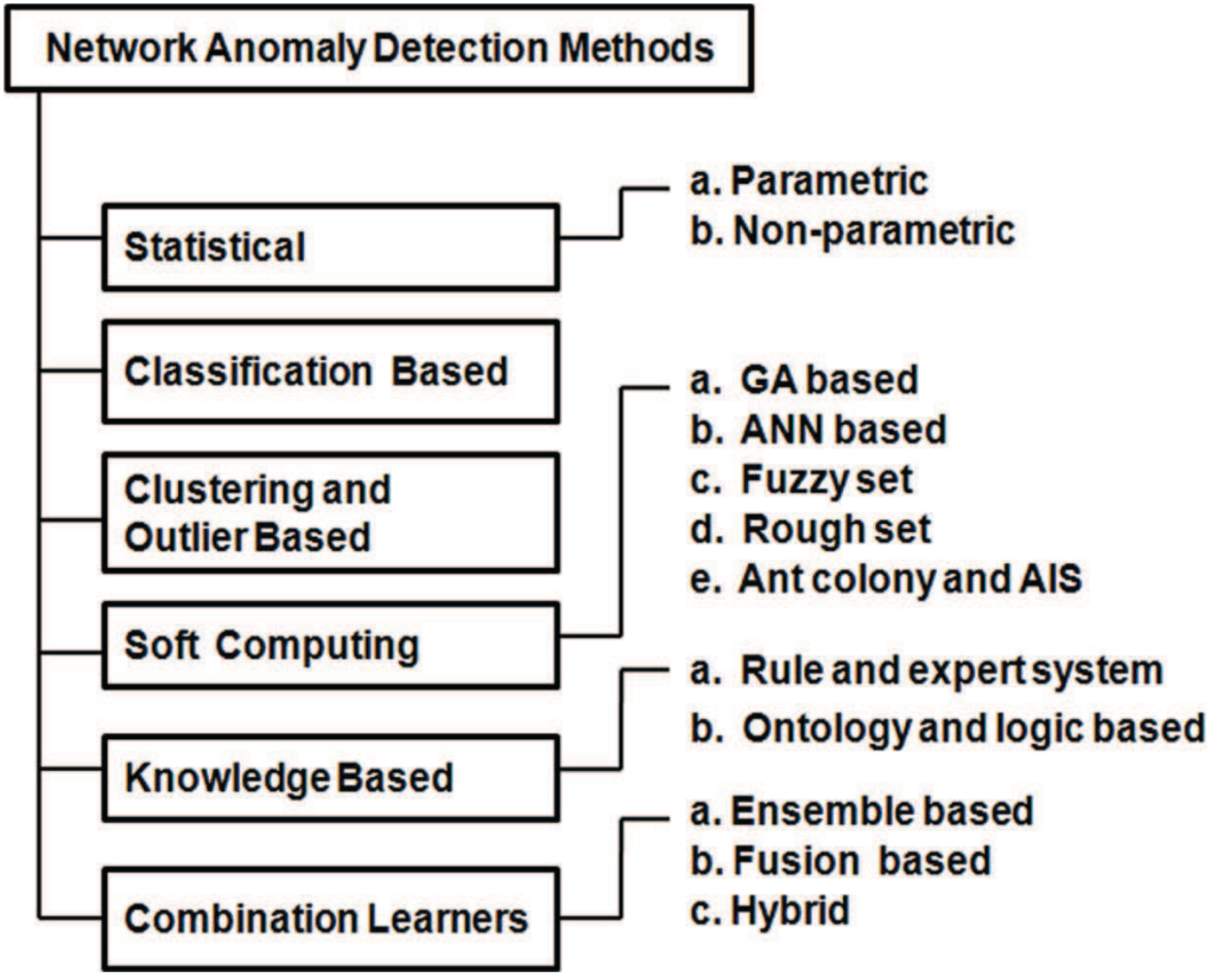
Connection window-based features:

- *count-dest-conn*, number of flows to unique destination IP addresses inside the network in the last *n* flows from the same source
- *count-src-conn*, number of flows from unique source IP addresses inside the network in the last *n* flows to the same destination
- *count-serv-src-conn*, number of flows from the source IP to the same destination port in the last *n* flows
- *count-serv-dest-conn*, number of flows to the destination IP address using same source port in the last *n* flows.

MINDS: zhodnocení

- Navrženo a experimentálně vyzkoušeno v roce 2002.
- Úspěšně detekován SQL červ *Slammer* v prvotním stádiu šíření, kdy tvořil jen 2 % celkového provozu.
- Snort na základě detekce signatur neodhalil tuto mutaci červu.
- Výpočetně náročné: LOF počítá vzdálenosti mezi každými dvěma body, tzn. $O(n^2)$.
- Pro velké sítě nutno použít vzorkování, to však může mít vliv na přesnost detekce.
- Zdrojové kódy MINDS však nejsou dostupné.

Přehled dalších metod



Entropie

- Nechť $X = \{n_i,\ i = 1, \ldots, N\}$, kde hodnota i nastává n_i krát v tomto vzorku. N je počet různých hodnot.
- Entropie vzorku (*sample entropy*) je pak definována takto:

$$H(X) = -\sum_{i=1}^{N} (\frac{n_i}{S}) \log_2(\frac{n_i}{S}),\ S = \sum_{i=1}^{N} n_i$$

- S je celkový počet pozorování.
- Hodnota entropie leží v intervalu $[0, log_2 N]$.
- Entropie je rovna nule, právě tehdy když jsou všechna pozorování stejná.
- Entropie je $log_2 N$, právě když jsou četnosti stejné, tj. $n_1 = n_2 = \cdots = n_N$
- *Relativní* entropie: $h(X) = \frac{H(X)}{H_{max}} = \frac{H(X)}{log_2 N}$

Výpočet entropie: příklad

- IP adresy v souboru: 10.0.0.1, 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.2, 10.0.0.1, 10.0.0.4, 10.0.0.1
- Pozorování: celkem 8 adres, 4 různé hodnoty.
- $N = 4, S = 8, X = \{n_1, n_2, n_3, n_4\} = \{4, 2, 1, 1\}, H_{max} = log_2 4 = 2$
- $\frac{n_1}{S} = \frac{4}{8}, \frac{n_2}{S} = \frac{2}{8}, \frac{n_3}{S} = \frac{1}{8}, \frac{n_4}{S} = \frac{1}{8}$
- $H(X) = -(\frac{4}{8} log_2 \frac{4}{8} + \frac{2}{8} log_2 \frac{2}{8} + \frac{1}{8} log_2 \frac{1}{8} + \frac{1}{8} log_2 \frac{1}{8}) = 1,75$
- $h(X) = \frac{H(X)}{H_{max}} = \frac{1,75}{2} = 0,875$

Shrnutí

- Dnes prezentované přístupy jsou metody pro detekce útoků/anomálií.
- Zajímají nás odlišnosti (anomálie) ve sledovaném vzorku.

- Pomocí predikce časových řad lze detekovat neočekávaná pozorování.
- Pomocí LOF lze najít „body”, které nespadají do žádného (i jinak hustého) shluku.
- Entropie udává míru neurčitosti (neuspořádanosti). Aplikace na síťový provoz vypadá slibně.

Další literatura

- *Introduction to Time Series Analysis*
  `http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm`
- Brutlag, J.: *Aberrant behaviour Detection in Time Series for Network Monitoring*, 2000
  `http://www.usenix.org/events/lisa00/full_papers/brutlag/brutlag_html/index.html`
- *Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander LOF: Identifying Density-Based Local Outliers*
  `http://www.it.iitb.ac.in/~deepak/deepak/courses/mtp/papers/LOF-identifying%20density-based%20local%20outliers.pdf`