ANOMALY DETECTION

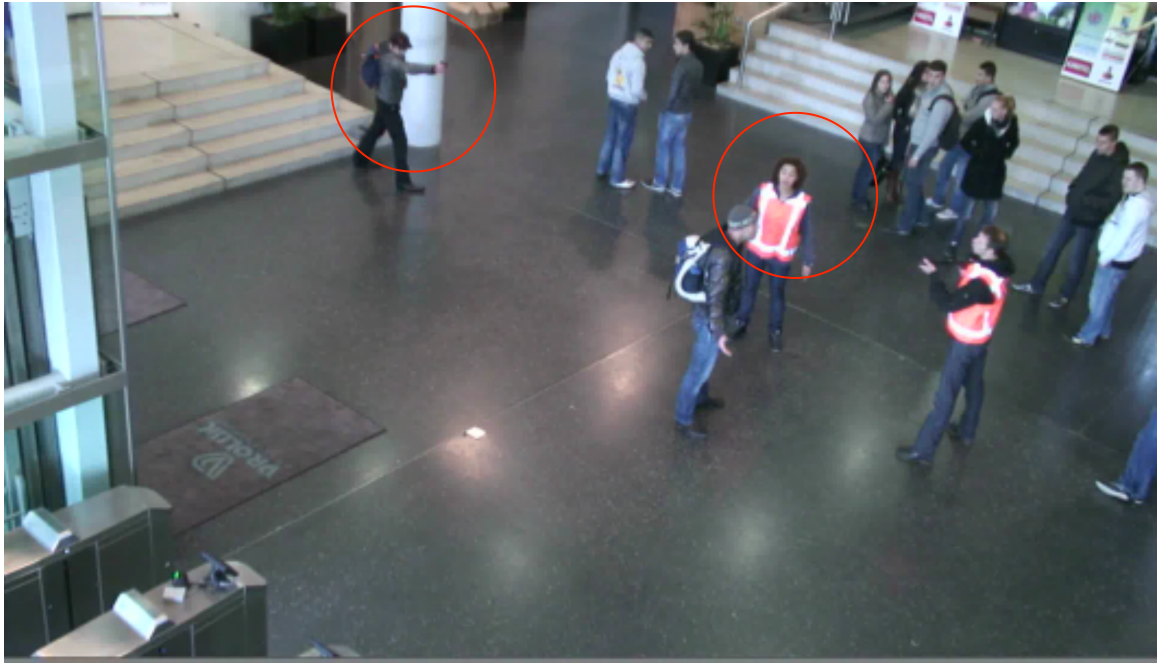
Piotr Zuraniewski

piotr.zuraniewski@tno.nl

TNO innovation for life

CAN DATA SCIENTISTS SAVE LIVES?

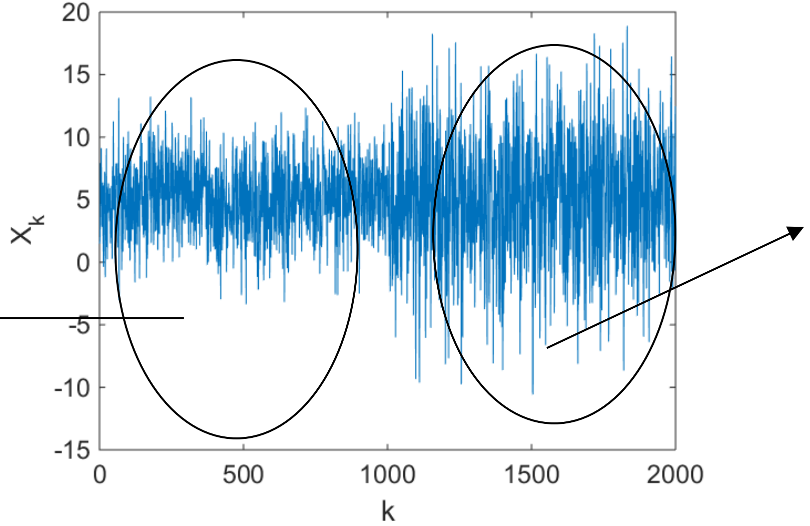
- You can do PhD in statistical methods in anomaly detection
  - I did at UvA ☺
- ...write papers on it, go to nice conferences and such
- But can you *actually* use it?
  - At TNO we do!
- If so, how does it work



ANOMALY DETECTION USING STATISTICAL METHODS

- Anomaly detection can be about finding „changepoint": moment from which **current statistical description** of data sample is **no longer valid**

part of data described well by given model X



data structure changed – model X not valid anymore

HOW TO: LEARN BASELINE, MODEL IT, SEE IF NEW DATA CONFORMS WITH BASELINE MODEL

- Collect ‘baseline’ data

- Using baseline data, find statistical description which you want to trace
  - Example: ‘mean value of X is 30’

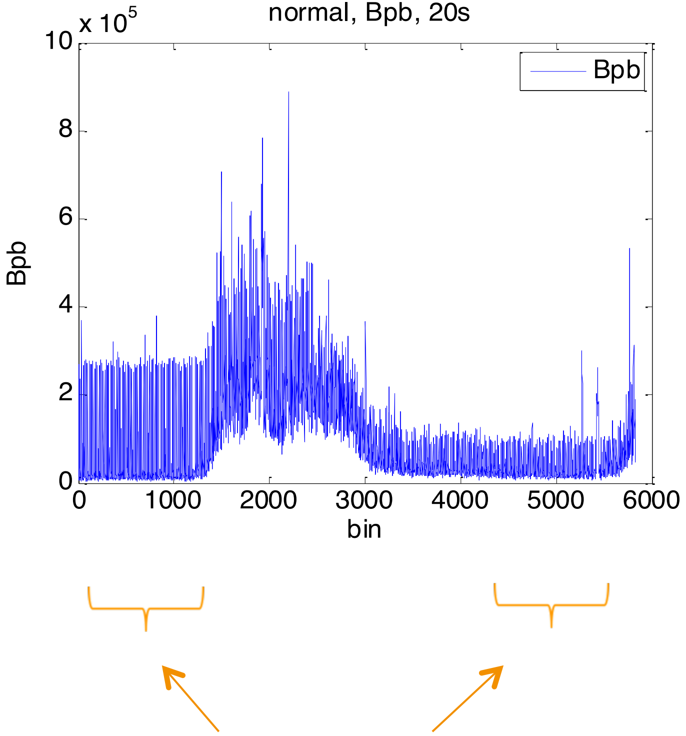
- Construct appropriate test statistics and apply it continuously to newly collected data
  - New data still conforms with baseline description? OK, no anomaly
  - Changepoint detected? Anomaly !

- Mind trade-off between false alarm ratio and miss probability – theoretical results help to control it

COLLECT ‘BASELINE’ DATA

- Collect clean (baseline) data reflecting „normal” situation
- Human expertise needed to confirm no known abnormalities are present

- Example: CyberAttack Detection (CAD) TNO project
  - Goal: prevent cyber espionage/data exfiltration
  - Use case: DNS protocol abused for data transfer
  - Baseline data: number of bytes carried by DNS in 20s time bins
  - „Clean data” collection occurred to be not so clean and revealed misconfigured Windows machine sending excessive DNS traffic



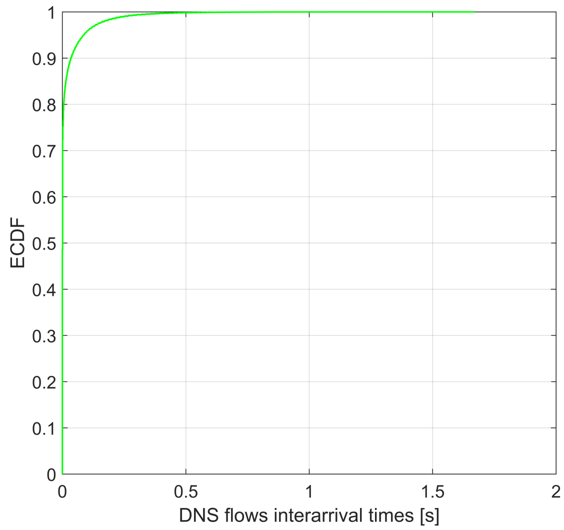
The same part of the day, different behaviour

USING BASELINE DATA, FIND STATISTICAL DESCRIPTION WHICH YOU WANT TO TRACE

- Use clean (baseline) data to define a statistical measure (along with its properties) to which you will compare new data

- It can be simple:
  - CAD example: mean number of bytes of „legal" DNS traffic per 20s bin between 3pm and 4pm is 201660

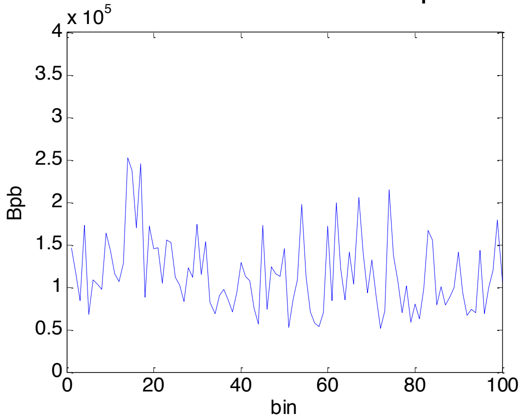
- ...or more involved
  - Empirical cumulative distribution function of „legal" DNS flows interarrival times has the following shape

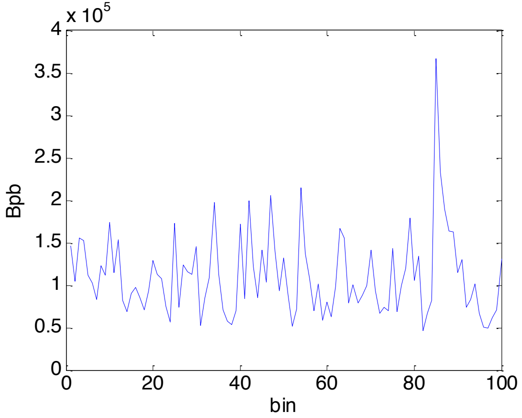


CONSTRUCT APPROPRIATE TEST STATISTICS AND APPLY IT TO NEWLY COLLECTED DATA

- Test statistics applied (continuously) to new data tells you what is a probability that your statistical description, derived from baseline data, is still fine
- Example:
  - probability of observing values > 2.5 * 10^5 is 10% *) → no alarm
  - probability of observing values > 3.5 * 10^5 is 0.1% *) → alarm

  *)numbers are not exact here – concept illustration only

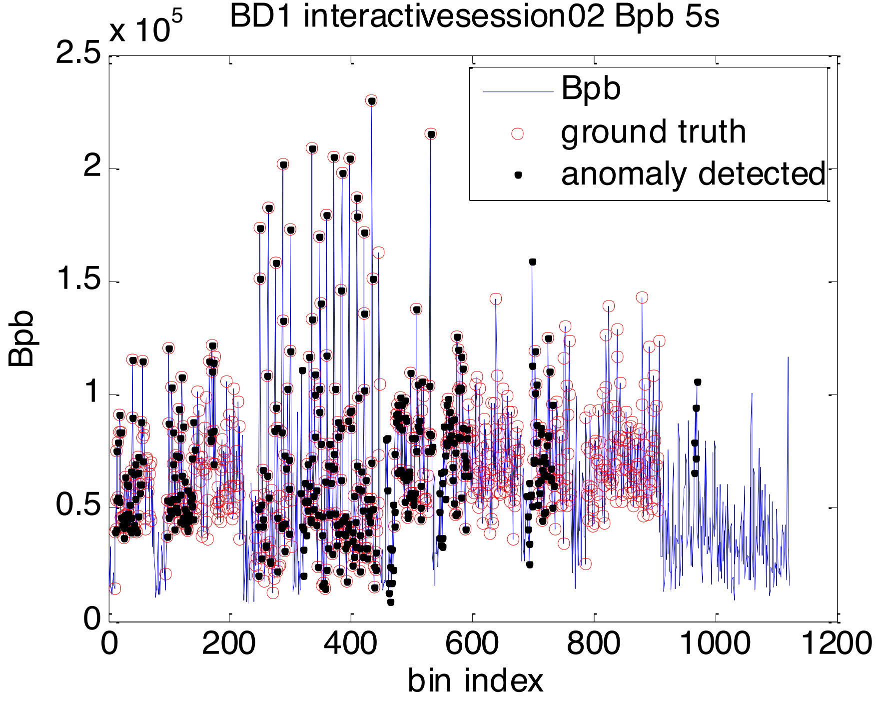




MIND TRADE-OFF BETWEEN FALSE ALARM RATIO AND MISS PROBABILITY

- Most often we cannot have 100% detection and 0% false alarms
- Trade-off: if we want to detect more, we have to accept higher false alarm rate
- Relation is usually complex and nonlinear
- Theoretical results help to control it

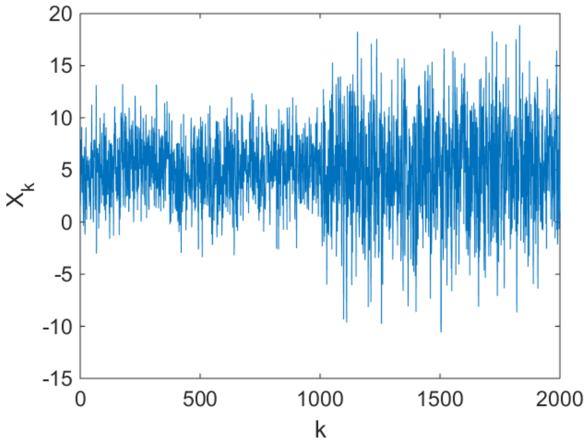
- CAD example again:
  - some time bins when system was under attack were missed (empty red circle)...
  - ...but „event-wise” we were successful in detection (note, however, one false alarm)



SORRY: NO GENERAL „APPROPRIATE” TEST STATISTICS EXISTS

- There is no „one-size-fits-all” method (good news: job for data scientists ☺)
- However, frequently, Cumulative Sum (CUSUM) method works reasonably well

- General idea: we want to detect if in observed sample $X_0, X_1, \ldots, X_n$
  - $X_0, X_1, \ldots, X_{k0-1}$ are independent, distributed according to *f(x)*,
  - $X_{k0}, X_{k0+1}, \ldots, X_n$ are independent, distributed according to *g(x)*
  - Changepoint k_0 may be at time $k = 1, 2, \ldots, n$



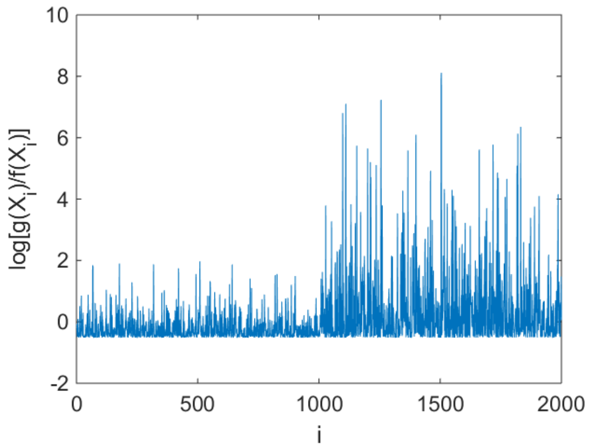
CUSUM RESULT AND ITS DRAWBACK

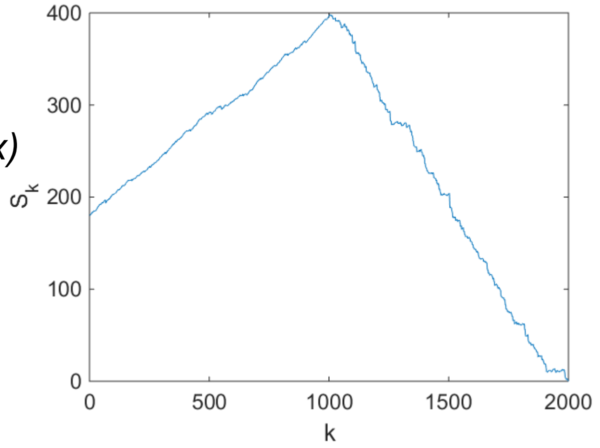
- Consider ratio $g(X_i)/f(X_i)$
  - Very informally: chances that X_i was sampled from *g(x)* vs. *f(x)*
  - If X_i really originates from *g(x)* this ratio (and its log) should be „large"
- Take the maximum over S_k to localize potential changepoint, where:

$$S = \max_{k=1,2,\ldots,n} S_k = \max_{k=1,2,\ldots,n} \sum_{i=k}^{n} \log \frac{g(X_i)}{f(X_i)}$$

- If this *test statistic* S is larger than a threshold b>0, raise an alarm

- CUSUM has obvious drawback: you have to fully specify distributions *f(x)* and *g(x);*
  - This can be far from reality; what to do then?



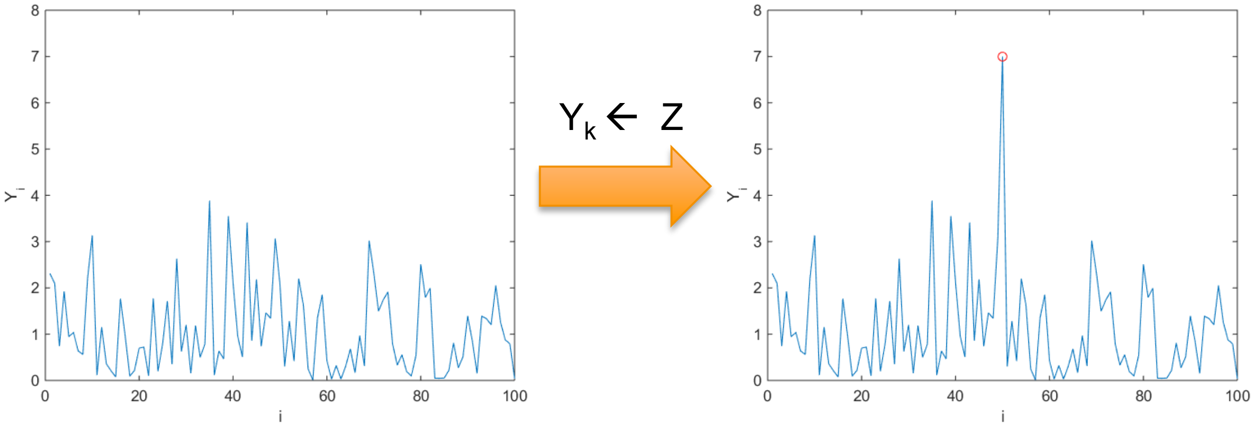


BOOTSTRAP

- Bootstrap methods are widely used is statistics
  - Useful especially in relatively small sample set-ups
  - ... and/or situations when experiment cannot be repeated
- General idea: create artificially more data (realizations) by drawing with replacement from original sample, keeping size of each realization equal to original one
  - Example: from original sample $\{x_1, x_2, x_3, x_4, x_5\}$ we can get something like
    - $\{x_3, x_2, x_5, x_4, x_5\}$
    - $\{x_2, x_1, x_5, x_1, x_1\}$
    - $\{x_4, x_4, x_2, x_1, x_3\}$
    - ....
- Number of such realizations can be arbitrary high, giving opportunity for (a kind of) large sample analysis

BOOTSTRAP OUTLIER ('BOOTLIER') DETECTION METHOD IDEA

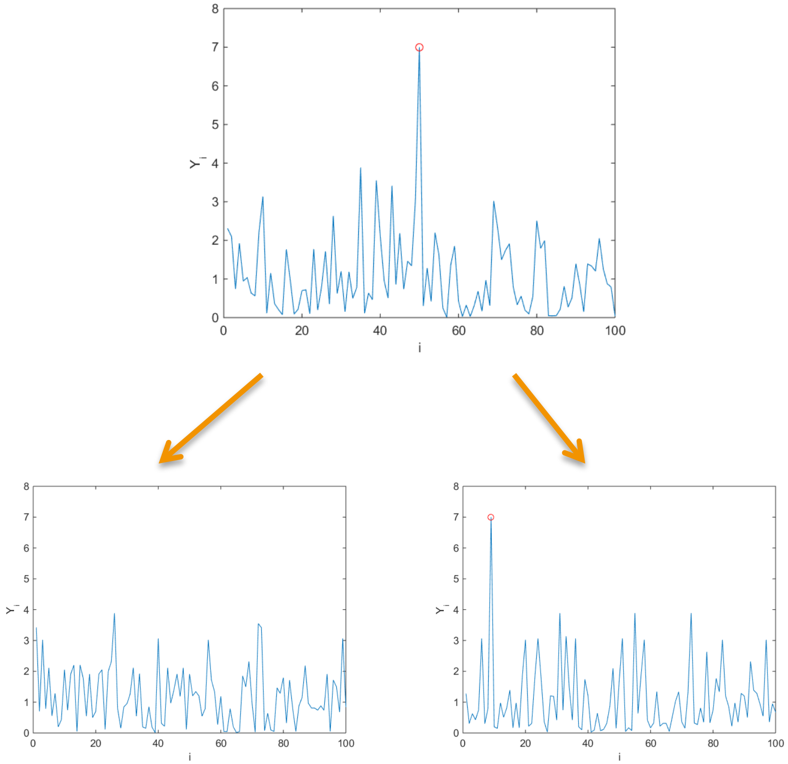
- Non-parametric, graphical method for detecting outlier(s) in data (Singh & Xie, 2003)
- Consider original sample **Y**={$Y_1,\ldots,Y_n$} which is independent, identically distributed (i.i.d.)
- To illustrate „bootlier" method, introduce outlier by replacing certain Y_k with large value Z (call new sample $\mathbf{Y_Z}$)



BOOTSTRAP OUTLIER ('BOOTLIER') DETECTION METHOD IDEA

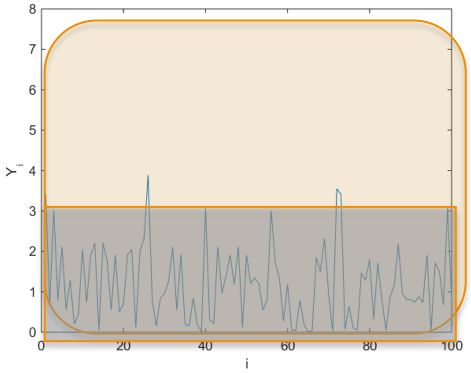
- Create bootstrap sample $\mathbf{Y}^* = \{Y^*_1,\ldots,Y^*_n\}$ by drawing from $\mathbf{Y}_\mathbf{Z}$ with replacement
- Two situations may happen:
  - bootstrap sample **does not** contain outlier Z
  - **...** or **does** contain outlier Z

- The chance $\mathbf{Y}^*$ does **not** contain Z is $(1\text{-}1/n)^n$
  - 1/e (or ~37%) for large n



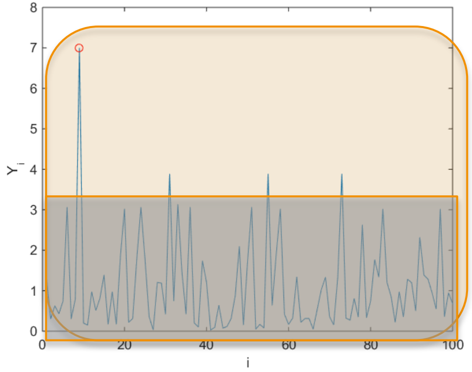
BOOTSTRAP OUTLIER ('BOOTLIER') DETECTION METHOD IDEA

- Calculate
  - mean ***M*** of $\{Y^*_1,\ldots,Y^*_n\}$
  - … and trimmed mean ***TM*** of $\{Y^*_1,\ldots,Y^*_n\}$
  - (***TM*** = mean without *k* most extreme values)
- If there is outlier in bootstrap sample, (***M – TM)*** difference is expected to be larger compared to case without outlier



M = 1.1928 TM = 1.0939

M – TM = 0.0989

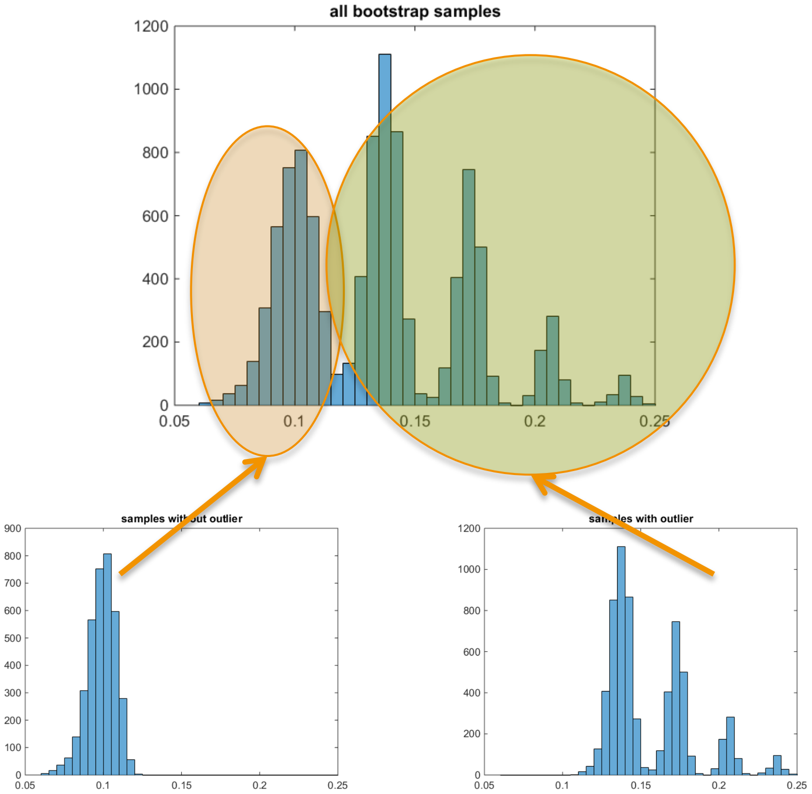


M = 1.1144 TM = 0.9665

M – TM = 0.1479

BOOTSTRAP OUTLIER ('BOOTLIER') DETECTION METHOD IDEA

- For large number of bootstrap samples we can create histogram of such differences (***M – TM)***
- Presence of outlier in original sample will make histogram for all bootstrap samples multimodal ('bumpy')
- Bootstrap samples without outlier contribute to first mode (small values)
- ...while bootstrap samples with oulier (perhaps present more than once!) create modes for larger values



SUMMARY

- Statistical anomaly detection is both exciting and applicable
  - Variety of methods exist, both for „nice” and „rough” cases
  - ...but sometimes we you have to make our own algorithm

- At TNO we use Anomaly Detection in various (BigData) projects
  - Cybersecurity
  - Communication networks monitoring also in SDN
  - Physical infrastructure monitoring (bridges, dams)
  - Behaviour monitoring
  - ...

THANK YOU FOR YOUR ATTENTION

TNO innovation for life