Anomaly Detection with Extreme Value Theory

A. Siffer, P-A Fouque, A. Termier and C. Largouet
April 26, 2017
Contents

Context

Providing better thresholds

Finding anomalies in streams

Application to intrusion detection

A more general framework
Context
GENERAL MOTIVATIONS

Massive usage of the Internet
General motivations

- Massive usage of the Internet
  - More and more vulnerabilities
General motivations

How the Carbanak cybergang stole $1bn
A targeted attack on a bank

— Massive usage of the Internet
  • More and more vulnerabilities
  • More and more threats

1 Tbps DDoS Attack
Powered By 150,000 Hacked IoT Devices
General motivations

How the Carbanak cybergang stole $1bn
A targeted attack on a bank

- Massive usage of the Internet
  - More and more vulnerabilities
  - More and more threats

- Awareness of the sensitive data and infrastructures

1 Tbps DDoS Attack
Powered By 150,000 Hacked IoT Devices
General motivations

How the Carbanak cybergang stole $1bn
A targeted attack on a bank

→ Massive usage of the Internet
  • More and more vulnerabilities
  • More and more threats

→ Awareness of the sensitive data and infrastructures

⇒ Network security: a major concern
IDS (Intrusion Detection System)

- Monitor traffic
- Detect attacks
IDS (Intrusion Detection System)
  - Monitor traffic
  - Detect attacks

Current methods: rule-based
  - Work fine on common and well-known attacks
  - Cannot detect new attacks
A Solution

- IDS (Intrusion Detection System)
  - Monitor traffic
  - Detect attacks

- Current methods: rule-based
  - Work fine on common and well-known attacks
  - Cannot detect new attacks

- Emerging methods: anomaly-based
  - Use the network data to estimate a normal behavior
  - Apply algorithms to detect abnormal events (attacks)
Overview

Basic scheme

data → ALGORITHM → alerts

Many "standard" algorithms have been tested.
Complex pipelines are emerging (ensemble/hybrid techniques).
Basic scheme

- Many "standard" algorithms have been tested
— Basic scheme

— Many "standard" algorithms have been tested
— Complex pipelines are emerging (ensemble/hybrid techniques)
Algorithms are not magic

- They give some information about data (scores)

- The thresholds are often hard-set
  - Expertise
  - Fine-tuning
  - Distribution assumption

Our idea: provide dynamic threshold with a probabilistic meaning
Algorithms are not magic

- They give some information about data (scores)
- But the decision often rely on a human choice

```python
if score > threshold then trigger alert
```
Inherent Problem

- Algorithms are not magic
  - They give some information about data (scores)
  - But the decision often rely on a human choice

  \[
  \text{if score} > \text{threshold then trigger alert}
  \]

- The thresholds are often hard-set
  - Expertise
  - Fine-tuning
  - Distribution assumption
### Inherent Problem

- **Algorithms are not magic**
  - They give some information about data (scores)
  - But the decision often rely on a human choice

  ```python
  if score>threshold then trigger alert
  ```

- **The thresholds are often hard-set**
  - Expertise
  - Fine-tuning
  - Distribution assumption

- **Our idea:** provide dynamic threshold with a probabilistic meaning
Providing better thresholds
My problem

How to set $z_q$ such that $P(X ≥ z_q) < q$?
How to set $z_q$ such that $\Pr(X_\varepsilon > z_q) < q$?
Solution 1: Empirical Approach

- **Drawbacks**: stuck in the interval, poor resolution.
Solution 1: Empirical Approach

Drawbacks: stuck in the interval, poor resolution
Solution 1: empirical approach

Drawbacks: stuck in the interval, poor resolution
Solution 2: Standard Model

Drawbacks: manual step, distribution assumption
Solution 2: Standard Model

Drawbacks: manual step, distribution assumption
SOLUTION 2: STANDARD MODEL

Solution 2: Standard Model

Drawbacks: manual step, distribution assumption
Different clients and/or temporal drift
## Results

<table>
<thead>
<tr>
<th>Properties</th>
<th>Empirical quantile</th>
<th>Standard model</th>
</tr>
</thead>
<tbody>
<tr>
<td>statistical guarantees</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>easy to adapt</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>high resolution</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Inspection of extreme events

Daily payment by credit card (€)

Frequency
Extreme Value Theory

Main result (Fisher-Tippett-Gnedenko, 1928)

The extreme values of any distribution have nearly the same distribution (called Extreme Value Distribution).

\[ (X > x) \]

heavy tail
exponential tail
bounded tail
Main result (Fisher-Tippet-Gnedenko, 1928)

The extreme values of any distribution have nearly the same distribution (called Extreme Value Distribution)
Main result (Fisher-Tippett-Gnedenko, 1928)

The extreme values of any distribution have nearly the same distribution (called Extreme Value Distribution)
Let $X_1, X_2, \ldots, X_n$ a sequence of i.i.d. random variables with

$$S_n = \sum_{i=1}^{n} X_i \quad \text{and} \quad M_n = \max_{1 \leq i \leq n} (X_i)$$
Let $X_1, X_2, \ldots X_n$ a sequence of i.i.d. random variables with

$$S_n = \sum_{i=1}^{n} X_i \quad M_n = \max_{1 \leq i \leq n} (X_i)$$

Central Limit Theorem

$$\frac{S_n - n\mu}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$
Let $X_1, X_2, \ldots, X_n$ a sequence of i.i.d. random variables with

$$S_n = \sum_{i=1}^{n} X_i \quad M_n = \max_{1 \leq i \leq n} (X_i)$$

Central Limit Theorem

$$\frac{S_n - n \mu}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

FTG Theorem

$$\frac{M_n - a_n}{b_n} \xrightarrow{d} \text{EVD}(\gamma)$$
A MORE PRACTICAL RESULT

Second theorem of EVT (Pickands-Balkema-de Haan, 1974)

The excesses over a high threshold follow a Generalized Pareto Distribution (with parameters $\gamma, \sigma$)
A more practical result

- Second theorem of EVT (Pickands-Balkema-de Haan, 1974)

  The excesses over a high threshold follow a Generalized Pareto Distribution (with parameters $\gamma, \sigma$)

- What does it imply?
  - we have a model for extreme events
  - we can compute $z_q$ for $q$ as small as desired
How to use EVT

- Get some data $X_1, X_2 \ldots X_n$
- Set a high threshold $t$ and retrieve the excesses $Y_j = X_{k_j} - t$ when $X_{k_j} > t$
How to use EVT

→ Get some data $X_1, X_2 \ldots X_n$

→ Set a high threshold $t$ and retrieve the excesses $Y_j = X_{kj} - t$ when $X_{kj} > t$

→ Fit a GPD to the $Y_j$ (⇒ find parameters $\gamma, \sigma$)
How to use EVT

- Get some data \(X_1, X_2 \ldots X_n\)

- Set a high threshold \(t\) and retrieve the excesses \(Y_j = X_{k_j} - t\) when \(X_{k_j} > t\)

- Fit a GPD to the \(Y_j\) (\(\rightarrow\) find parameters \(\gamma, \sigma\))

- Compute \(z_q\) such as \(\mathbb{P}(X > z_q) < q\)
How to use EVT

- Get some data $X_1, X_2 \ldots X_n$
- Set a high threshold $t$ and retrieve the excesses $Y_j = X_{k_j} - t$ when $X_{k_j} > t$
- Fit a GPD to the $Y_j$ ($\rightarrow$ find parameters $\gamma, \sigma$)
- Compute $z_q$ such as $\mathbb{P}(X > z_q) < q$
Finding anomalies in streams
Streaming Peaks-Over-Threshold (SPOT) algorithm

\[ X_1, X_2, \ldots, X_n \]

Calibration

\[ q_t(z, q_{\text{stream}}) \]

\[ X_i > n \]

Trigger alarm: yes, no

\[ X_i > t \]

Update model: yes, no, drop

16
(initial batch)

$X_1, X_2 \ldots X_n$
**Streaming Peaks-Over-Threshold (SPOT) algorithm**

\(X_1, X_2 \ldots X_n\) → **Calibration**

\(q\)

\(X_i > n\) \(\rightarrow\) **yes**

\(X_i > z_q\) \(\rightarrow\) **trigger alarm**

\(t\) \(\rightarrow\) **yes**

\(z\) \(\rightarrow\) **drop**

<table>
<thead>
<tr>
<th>q</th>
<th>zq (stream)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>20</td>
</tr>
<tr>
<td>0.20</td>
<td>40</td>
</tr>
<tr>
<td>0.30</td>
<td>60</td>
</tr>
<tr>
<td>0.40</td>
<td>80</td>
</tr>
<tr>
<td>0.50</td>
<td>100</td>
</tr>
<tr>
<td>0.60</td>
<td>120</td>
</tr>
</tbody>
</table>

16
Streaming Peaks-Over-Threshold (SPOT) Algorithm

\[ X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION} \]

\( q \) (initial batch)

\[ X_i > \alpha \rightarrow \text{trigger alarm} \]

\[ X_i > t \rightarrow \text{yes} \]

\[ \text{update model} \]

\[ \text{drop} \]
Streaming Peaks-Over-Threshold (SPOT) algorithm

(initial batch)

\(X_1, X_2 \ldots X_n\) → CALIBRATION

\(q\)

\(X_i > n \quad X_i > z_q\)

trigger alarm

yes

no

\(X_i > t\)

yes

update model

no

drop

16
Streaming Peaks-Over-Threshold (SPOT) algorithm

(initial batch)

$X_1, X_2 \ldots X_n \rightarrow$ CALIBRATION

$q$
Streaming Peaks-Over-Threshold (SPOT) algorithm

(initial batch)

$X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION}$

(stream)

$X_{i>n}$
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

\[ X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION} \]

(q)

(stream)

\[ X_i > z_q \]

\[ X_i > n \]

<table>
<thead>
<tr>
<th>t</th>
<th>z_q</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.30</td>
</tr>
<tr>
<td>40</td>
<td>0.20</td>
</tr>
<tr>
<td>60</td>
<td>0.10</td>
</tr>
<tr>
<td>80</td>
<td>0.00</td>
</tr>
<tr>
<td>100</td>
<td>0.00</td>
</tr>
<tr>
<td>120</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Trigger alarm:
- Yes
- No

Update model:
- Yes
- No
- Drop
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

\( X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION} \)

\( q \)

(stream)

\( X_i > n \rightarrow X_i > z_q \)

\( X_i > z_q \rightarrow \text{TRIGGER ALARM} \)

YES

**Calibration**

\( q_0 = \{0.10, 0.20, 0.30\} \)

\( z_q \)

\( t \)

\( X_i > z_q \)

\( X_i > t \)

yes

no

update model

drop
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

\[ X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION} \]

\[ q \]

\[ X_i > n \rightarrow X_i > z_q \rightarrow \text{TRIGGER ALARM} \]

(stream)

\[ X_i > n \]

\[ X_i > z_q \rightarrow \text{YES} \rightarrow X_i > t \rightarrow \text{NO} \]

Graph showing calibration with thresholds and decisions for streaming data.
**Streaming Peaks-Over-Threshold (SPOT) Algorithm**

(initial batch)

\[ X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION} \]

(stream)

\[ X_i > n \rightarrow X_i > z_q \rightarrow \text{TRIGGER ALARM} \]

\[ X_i > t \rightarrow \text{UPDATE MODEL} \]
(initial batch)

$X_1, X_2 \ldots X_n \rightarrow \text{CALIBRATION}$

(stream)

$X_i > n \rightarrow X_i > z_q$  

$X_i > t \rightarrow X_i > t$  

TRIGGER ALARM

UPDATE MODEL

DROP
Can we trust that threshold $z_q$?

- An example with ground truth: a Gaussian White Noise
  - 40 streams with 200,000 iid variables drawn from $\mathcal{N}(0, 1)$
  - $q = 10^{-3} \Rightarrow$ theoretical threshold $z_{th} \approx 3.09$
Can we trust that threshold $z_q$?

- An example with ground truth: a Gaussian White Noise
  - 40 streams with 200,000 iid variables drawn from $\mathcal{N}(0, 1)$
  - $q = 10^{-3} \Rightarrow$ theoretical threshold $z_{th} \approx 3.09$

- Averaged relative error

![Averaged relative error graph](image-url)
Application to intrusion detection
Lack of relevant public datasets to test the algorithms ...
Lack of relevant public datasets to test the algorithms...

KDD99? See [McHugh 2000] and [Mahoney & Chan 2003]
About the Data

- Lack of relevant public datasets to test the algorithms ...
- KDD99 ? See [McHugh 2000] and [Mahoney & Chan 2003]
- We rather use MAWI
  - 15 min a day of real traffic (.pcap file)
  - Anomaly patterns given by the MAWILab [Fontugne et al. 2010] with taxonomy [Mazel et al. 2014]
Lack of relevant public datasets to test the algorithms ...

KDD99? See [McHugh 2000] and [Mahoney & Chan 2003]

We rather use MAWI

- 15 min a day of real traffic (.pcap file)
- Anomaly patterns given by the MAWILab [Fontugne et al. 2010] with taxonomy [Mazel et al. 2014]

Preprocessing step: raw .pcap → NetFlow format (only metadata)
The ratio of SYN packets: relevant feature to detect network scan [Fernandes & Owezarski 2009]
An example to detect network SYN scan

- The ratio of SYN packets: relevant feature to detect network scan [Fernandes & Owezarski 2009]
The ratio of SYN packets: relevant feature to detect network scan [Fernandes & Owezarski 2009]

Goal: find peaks
Parameters: $q = 10^{-4}$, $n = 2000$ (from the previous day record)
SPOT RESULTS

- Parameters: $q = 10^{-4}$, $n = 2000$ (from the previous day record)
Do we really flag scan attacks?

- The main parameter $q$: a False Positive regulator

86% of scan flows detected with less than 4% of FP
Do we really flag scan attacks?

The main parameter $q$: a False Positive regulator

86% of scan flows detected with less than 4% of FP
Do we really flag scan attacks?

- The main parameter $q$: a False Positive regulator

- 86% of scan flows detected with less than 4% of FP
A more general framework
A single main parameter \( q \)
- With a probabilistic meaning \( \mathbb{P}(X > z_q) < q \)
- False Positive regulator
SPOT Specifications

- A single main parameter $q$
  - With a probabilistic meaning $\Rightarrow P(X > z_q) < q$
  - False Positive regulator

- Stream capable
  - Incremental learning
  - Fast ($\sim 1000$ values/s)
  - Low memory usage (only the excesses)
SPOT

- performs dynamic thresholding without distribution assumption
- uses it to detect network anomalies
Other things?

- SPOT
  - performs dynamic thresholding without distribution assumption
  - uses it to detect network anomalies
- But it could be adapted to
Other things?

- SPOT
  - performs dynamic thresholding without distribution assumption
  - uses it to detect network anomalies

- But it could be adapted to
  - compute upper and lower thresholds
Other things?

- SPOT
  - performs dynamic thresholding without distribution assumption
  - uses it to detect network anomalies
- But it could be adapted to
  - compute upper and lower thresholds
  - other fields
Other things?

- SPOT
  - performs dynamic thresholding without distribution assumption
  - uses it to detect network anomalies

- But it could be adapted to
  - compute upper and lower thresholds
  - other fields
  - drifting contexts (with an additional parameter) → DSPOT
A RECENT EXAMPLE

Thursday the 9th of February 2017
• 9h: explosion at Flamanville nuclear plant
• 11h: official declaration of the incident by EDF

What about the EDF stock prices?
A recent example

Thursday the 9th of February 2017
A recent example

Thursday the 9th of February 2017

- 9h: explosion at Flamanville nuclear plant
A recent example

- Thursday the 9th of February 2017
  - 9h: explosion at Flamanville nuclear plant
  - 11h: official declaration of the incident by EDF
Thursday the 9th of February 2017

- **9h**: explosion at Flamanville nuclear plant
- **11h**: official declaration of the incident by EDF

What about the EDF stock prices?
EDF stock prices

EDF stock price (€)

Time


EDF STOCK PRICES

The graph shows the EDF stock price (€) over time from 09:02 to 17:14. The prices fluctuate throughout the day, with a notable drop around 11:32.
Conclusion

Context: A great deal of work has been done to develop anomaly detection algorithms. Problem: Decision thresholds rely on either distribution assumption or expertise. Our solution: Building dynamic threshold with a probabilistic meaning.

Application to detect network anomalies. But a general tool to monitor online time series in a blind way.

Future: Adapt the method to higher dimensions.
Context: A great deal of work has been done to develop anomaly detection algorithms.
Context: A great deal of work has been done to develop anomaly detection algorithms

Problem: Decision thresholds rely on either distribution assumption or expertise
Context: A great deal of work has been done to develop anomaly detection algorithms

Problem: Decision thresholds rely on either distribution assumption or expertise

Our solution: Building dynamic threshold with a probabilistic meaning
Conclusion

- **Context**: A great deal of work has been done to develop anomaly detection algorithms
- **Problem**: Decision thresholds rely on either distribution assumption or expertise
- **Our solution**: Building dynamic threshold with a probabilistic meaning
  - Application to detect network anomalies
Context: A great deal of work has been done to develop anomaly detection algorithms.

Problem: Decision thresholds rely on either distribution assumption or expertise.

Our solution: Building dynamic threshold with a probabilistic meaning:
  - Application to detect network anomalies
  - But a general tool to monitor online time series in a blind way

Future:
Adapt the method to higher dimensions.
— **Context**: A great deal of work has been done to develop anomaly detection algorithms

— **Problem**: Decision thresholds rely on either distribution assumption or expertise

— **Our solution**: Building dynamic threshold with a probabilistic meaning
  - Application to detect network anomalies
  - But a general tool to monitor online time series in a blind way

— **Future**: Adapt the method to higher dimensions