Anomaly Detection with Extreme Value Theory

A. Siffer, P-A Fouque, A. Termier and C. Largouet
May 30, 2017
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
WannaCry ransomware used in widespread attacks all over the world

- 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

WannaCry ransomware used in widespread attacks all over the world

- 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
WannaCry ransomware used in widespread attacks all over the world

- 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
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)
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)
Basic scheme

Data $\rightarrow$ Algorithm $\rightarrow$ 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)
Inherent problem

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

  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

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

- 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

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

○ 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 $\mathbb{P}(X \epsilon > z_q) < q$?
**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

Drawbacks: manual step, distribution assumption
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

![Graph showing daily payment by credit card (€)]

- Frequency
- Daily payment by credit card (€)

Probability estimation
Inspection of extreme events

![Histogram of daily payment by credit card (€)]

Probability estimation?
Extreme Value Theory

Main result (Fisher-Tippett-Gnedenko, 1928)

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

- Heavy tail
- Exponential tail
- Bounded tail

\[(X > x)\]
Main result (Fisher-Tippett-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

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

$$\frac{M_n - a_n}{b_n} \xrightarrow{d} \text{EVD}(\gamma)$$
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 $Y_j = X_{k_j} - t$ when $X_{k_j} > t$
- 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 $Y_j$ (→ 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 $Y_j$ (→ 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**

$X_1; X_2; : : : ; X_n$

Calibration

$q_t$

$z_q$(stream)

$X_i > n$

$X_i > z_q$

trigger alarm

yes

no

$X_i > t$

yes

update model

no

drop
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

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

(initial batch)

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

\[ q \]

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

\[ X_i > t \rightarrow \text{update model} \]

\[ \text{drop} \]

![Histogram](histogram.png)
(initial batch)

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

\[ q \]

\[ X_i > n \]

\[ X_i > z_q \text{ trigger alarm} \]

\[ t \]

\[ \text{yes} \quad \text{update model} \]

\[ \text{no} \quad \text{drop} \]
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

$X_1, X_2 \ldots X_n \rightarrow$ CALIBRATION

$q$

$X_i > n$:
- $X_i > z_q$ trigger alarm
- $X_i > t$ yes, update model; no, drop

Graph with $q$ and $t$ values
Streaming Peaks-Over-Threshold (SPOT) algorithm

(initial batch)

\[ X_1, X_2 \ldots X_n \]

\[ q \]

\[ q \]

CALIBRATION

\[ X_i > t \]

yes

update model

no

drop

\[ z_q \]

\[ t \]

\[ 0 \]

\[ 0.10 \]

\[ 0.20 \]

\[ 0.30 \]

20 40 60 80 100 120

16
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

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

(stream)

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

(initial batch)

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

(q)

(stream)

\[ X_{i>n} \rightarrow X_i > z_q \]

<table>
<thead>
<tr>
<th>( q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
</tr>
<tr>
<td>0.20</td>
</tr>
<tr>
<td>0.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( z_q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0.10</td>
</tr>
<tr>
<td>0.20</td>
</tr>
<tr>
<td>0.30</td>
</tr>
</tbody>
</table>

\( X_i > z_q \) trigger alarm
- yes
- no

\( X_i > t \) update model
- yes
- drop
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

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

\[ q \]

(stream)

\[ X_i > n \rightarrow X_i > z_q \]

\[ t \quad z_q \]

TRIGGER ALARM

YES

update model

no

drop
Streaming Peaks-Over-Threshold (SPOT) algorithm

(initial batch)

\[ X_1, X_2 \ldots X_n \]

\( \rightarrow \) CALIBRATION

\( \rightarrow \)

\( q \)

(stream)

\( X_i > n \)

\( X_i > z_q \rightarrow \) TRIGGER ALARM

YES

NO

\( X_i > t \)
Streaming Peaks-Over-Threshold (SPOT) algorithm

(Initial batch)

\[ X_1, X_2 \ldots X_n \]

\[ X_i > n \rightarrow \text{Calibration} \rightarrow \]

\[ q \]

(Threshold)

\[ X_i > z_q \rightarrow \text{Trigger alarm} \rightarrow \text{Yes, update model} \]

\[ X_i > t \rightarrow \text{No} \]

(Stream)

\[ X_i > n \]

\[ t \]

\[ z_q \]
Streaming Peaks-Over-Threshold (SPOT) Algorithm

(initial batch)

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

(q)

(1)

X \rightarrow \text{TRIGGER ALARM}

\[ X_i > n \]

\[ X_i > z_q \]

\[ X_i > t \]

\[ x \rightarrow \text{UPDATE MODEL} \]

\[ x \rightarrow \text{DROP} \]

(stream)

\[ X \rightarrow \text{TRIGGER ALARM} \]

\[ X \rightarrow \text{UPDATE MODEL} \]

\[ X \rightarrow \text{DROP} \]
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]

- Number of observations vs. Relative error
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]
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]
- Preprocessing step: raw .pcap → NetFlow format (only metadata)
AN EXAMPLE TO DETECT NETWORK SYN SCAN

- 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]
AN EXAMPLE TO DETECT NETWORK SYN SCAN

- 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)
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
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 $\Rightarrow \Pr(X > z_q) < q$
- False Positive regulator
SPOT SPECIFICATIONS

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

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

- 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
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
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?
EDF STOCK PRICES
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 thresholds 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

Our solution: Building dynamic threshold with a probabilistic meaning

Future: Adapt the method to higher dimensions

Application to detect network anomalies

But a general tool to monitor online time series in a blind way
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
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

Future: Adapt the method to higher dimensions
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
— **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