Anomaly Detection in Computer Networks

Jaromír Navrátil

June 10, 2015

Outline

- Motivation.
- The data.
- Solution workflow + prototype demonstration.
- Prototype.
- Future challenges.
- Conclusion.

Motivation

- Company produces enterprise firewalls.
- Business Intelligence application:
  - Interactive visualization of firewall logs.
  - Used by domain experts.
  - ML module for anomaly detection.

The data

- Each proxy logs all relevant information of every request.
- BI have logs in DB aggregated by minute, hour or day.
- Example representation for ML:

The data

- Each proxy logs all relevant information of every request.
- BI have logs in DB aggregated by minute, hour or day.
- Example representation for ML:
  - Goal is to detect anomalous clients based on hourly sums of downloads, uploads and requests.
  - Entity is a group of examples.
  - Example is client-day.
  - Attributes are download-hour, upload-hour, request-hour.

entity	class	0-down	0-up	0-req	...	23-req
10.0.0.10	2015-05-25	6000	45000	65	...	4
10.0.0.10	2015-05-26	9500	42000	45	...	5
10.0.0.12	2015-05-25	40	30	1	...	2

Solution workflow + prototype demonstration

- Obtain data from Business Intelligence DB.
- Transform data to representation suitable for ML.
- Random Forest to create distance matrix.
- Agglomerative clustering to obtain clusters (i.e. classes).
- Cross-validation to obtain incorrectly classified examples.
  - K-fold cross-validation of binary Random Forest.
- Present top N incorrectly classified examples.

Prototype

- Business Intelligence application:
  - AngularJS client
  - NodeJS server
  - PostgreSQL DB + perl wrapper
- Machine Learning module:
  - NodeJS script
  - Random Forest
  - Agglomerative clustering (single/average/complete linkage)

Future challenges

- Agglomerative clustering is slow - n^3 naive implementation, $n^2 \log n$ optimized.
- K-means algorithm is unusable in arbitrary space.
- Incorporate anomaly confidence.

Conclusion

- It appears to work.
- Rigorous experiments are required.