

Unsupervised streaming cyber-analytics

Niall Adams

Department of Mathematics and Data Science Institute
Imperial College London

April 2018

**Imperial College
London**

Contents

1. Some views on building cyber-analytics
2. Some methods:
 - ▶ “Future labelling”
 - ▶ Conditional Behaviour monitoring
 - ▶ aspects of data fusion
3. Conclusion

Some Collaborators:

- ▶ **ETH:** Dean Bodenham
- ▶ **Imperial, Mentat Innovations:** Christoforos Anagnostopoulos
- ▶ **Imperial:** Marina Evangelou, Elizabeth Riddle-Workman, Jack Hogan, Josh Plasse, Jordan Noble.

Funding:

- ▶ EPSRC, BAE Systems, QinetiQ, Imperial College

My research

I focus on developing methodology that is suited to data analysis problems arising in cyber-security, and similar areas.

The idea is to **complement** existing defences:

- ▶ Design statistical and machine learning procedures to
 - ▶ Model “normal” (historic) behaviour
 - ▶ As a basis for **anomaly detection**
 - ▶ **targeted** on aspects known to be suspicious, or at least **unusual** activity

Various types of (automatically collected) data available (to me) :

- ▶ Netflow - summaries of connections between devices
- ▶ Authentication events
- ▶ DNS - requests for resolving device names
- ▶ Web proxy and cache
- ▶ Host-based events (e.g. WLS)

Processing such data raises challenges:

- ▶ Data volume - often **huge**
- ▶ data velocity (fast, e.g. Imperial sees $\sim 20K$ Netflow events per second)
- ▶ Heterogeneity (some examples later)
- ▶ Temporal variation
- ▶ latent variable - human versus machine
- ▶ timeliness
- ▶ **Dearth** of labels (and signatures), and data at “lower conceptual level” than labels.
 - ▶ Perhaps availability of labels different at Cisco?

I am **not** trying to provide complete engineered solutions for these problems, but to understand the principles of designing algorithms. Full deployment, analytics to data, requires immense infrastructure and large and talented teams (**like you!**).

The **research** aspect relates to building models that can capture the underlying characteristics of the data, either

- ▶ In batch
- ▶ Sequentially

There are pros and cons to both, as well as **significant** *mathematical* and *computational* challenges.

A key issue is where to focus the analytic effort:

Graph



Clique



Edge: Session



Edge: NETFLOW



Edge: Packet



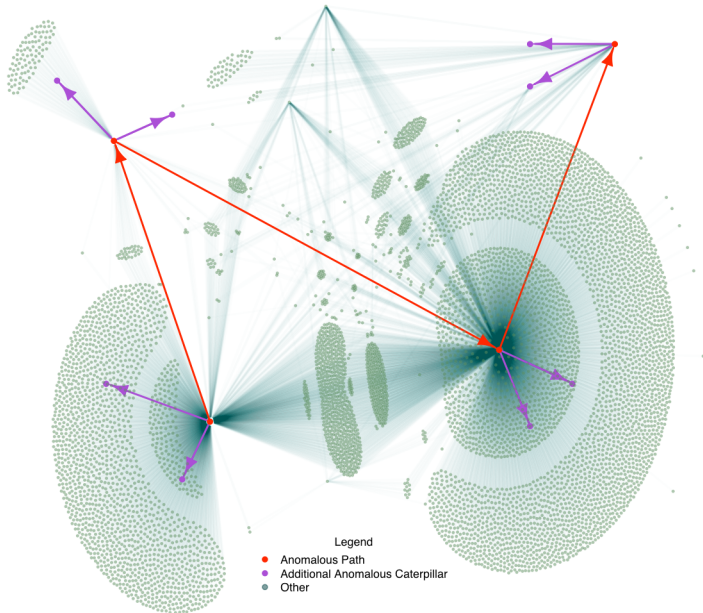
Given the set of challenges, the vision of large joint models over these diverse data sets appears out of reach. Instead, we think a complete data analytics system will

- ▶ Build analytics at different levels - batteries of analytics operating in an *unsupervised* manner, with reduced load for parameter setting
- ▶ Combine the weak anomalies detected by different analytics into a stronger conclusion (anomalies can happen for legitimate reasons, need to contain false positives, etc)
- ▶ Filter the total data down to a **manageable quantity for analysts** - aiming for **human-in-the-loop** security
- ▶ Providing visualisation and customisable analytics

It should be clear that network analysts are central to this formulation. **Quis custodiet ipsos custodes?** Need to have special controls with analyst-centric approaches.

Example

From colleagues (formerly!) at Los Alamos (see [2]):



2: Methods

- ▶ Streaming change point detection
- ▶ Conditional behaviour modelling

Future labelling

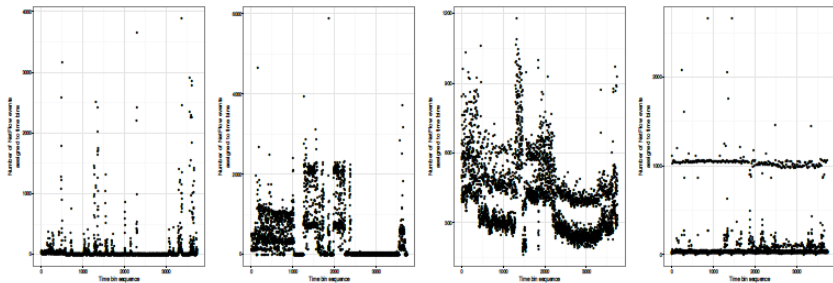
In the absence of good labels, it is difficult to access the full power of supervised methods.

As in intermediate position, consider defining a label for the data based on “what happens next?”. That is, we are monitoring a quantity of interest in contiguous bins, and the label for bin i is the quantity in bin $i + 1$. That is, can we learn to predict future behaviour?

This kind of reasoning:

- ▶ Provides a labelling, where we can define the response for whatever we are interested in
- ▶ may be difficult to implement in a production system! (sorry)
- ▶ Give an opening for data fusion and combination methods.

Consider the following plots: the number of Netflow events, y , associated with specific devices in 5 minute bins.



- ▶ Diverse behaviours here: bursts, multiple processes and more
- ▶ **IDEA:** in **absence of labels**, build model for y_{t+1} from features derived from bin t .
- ▶ Construct prediction interval, or p-value, for anomaly detection.
- ▶ Quite general: could operate in host based sensor, network traffic, etc.

Features

Very diverse range of possibilities for feature construction, depending on the data. Here we consider Netflow (see [11] for details).

Features for bin t :

- ▶ Time
 - ▶ Indicators for the bin time, time of day, day of week.
- ▶ Events
 - ▶ Summary statistics related to the number and properties of events in the bin
- ▶ Characteristics
 - ▶ Summary statistics related to Netflow features, such as packet and byte load
- ▶ Nature
 - ▶ Summary statistics related to ports and protocols

We generated a set of 35 feature variables.

Empty bins **crudely** encoded with missing value indicators.

Of course, many other things possible:

- ▶ More complicated characterisation of contiguous bins, Markov representations etc
- ▶ Identifying neighbours
- ▶ More formal time-series modelling
- ▶ Peer-group ideas
- ▶ etc

Models

This is a challenging modelling problem, because of the heterogeneity of the data.

We tried many modelling approaches, including:

- ▶ Poisson and (zero-inflated) linear models
- ▶ Regression trees
- ▶ Random Forests
- ▶ Quantile regression
- ▶ Quantile regression forests (QRF)

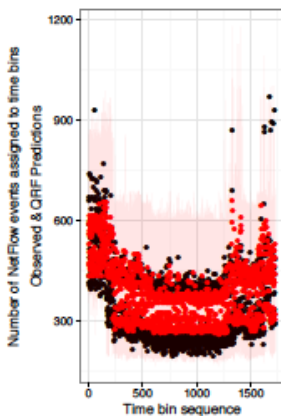
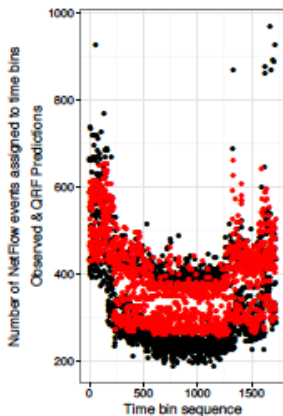
In terms of performance, we sought to determine whether any of these methods could beat a *naive* benchmark – predict $y_t + 1$ as y_t .

The benchmark is quite **hard to beat** – best effort (when considering performance over 60 devices) came from random forests and quantile regression forests. Assessed with MSE, MAE and other standard forecasting performances.

But **prediction is not enough**, we need to generate a measure of surprise, like a prediction interval or p-value.

QRF performs reasonably well here, we simply model the median, and some extreme quantiles to compute something with the properties of a prediction interval.

For example ...



- ▶ Multiple processes reasonably handled.
- ▶ Some amount of time change handled.
- ▶ Prediction interval is conservative (empirically).

Data Fusion

LANL recently released a large set of data, comprising WLS logs and NETFLOW events. Will use this to consider the idea above, but now in the context of data fusion.

Consider monitoring a device, where the binary response variable is “netflow connection to a new computer, in the next 5 minute bin”.

The feature variables can be constructed from NETFLOW, WLS or both. Much ingenuity is possible here, and we have only tried simple things so far, on a random sample of 500 devices.

WLS		NetFlow	
Name	Description	Name	Description
<i>Time</i>	Bin no. (1-288)	<i>Time</i>	Bin no. (1-288)
<i>WorkingHours</i>	Working hours indicator	<i>WorkingHours</i>	Working hours indicator
<i>Events</i>	No. of events	<i>OutDegree</i>	No. of unique outward communications
<i>UniqueEvents</i>	No. of unique event types	<i>InDegree</i>	No. of unique inward communications
<i>Logons</i>	No. of successful log-ons	<i>TotalOut</i>	Total no. of outward communications
<i>FailedLogons</i>	No. of failed log-ons	<i>TotalIn</i>	Total no. of inward communications
<i>Users</i>	No. of users active	<i>BytesOut</i>	Total no. of bytes sent
<i>Processes</i>	No. of processes started	<i>BytesIn</i>	Total no. of bytes received
		<i>PacketsOut</i>	Total no. of packets sent
		<i>PacketsIn</i>	Total no. of packets received
		<i>NewEvents</i>	No. of new events in the current bin
		<i>UDP</i>	No. of communications using the UDP protocol

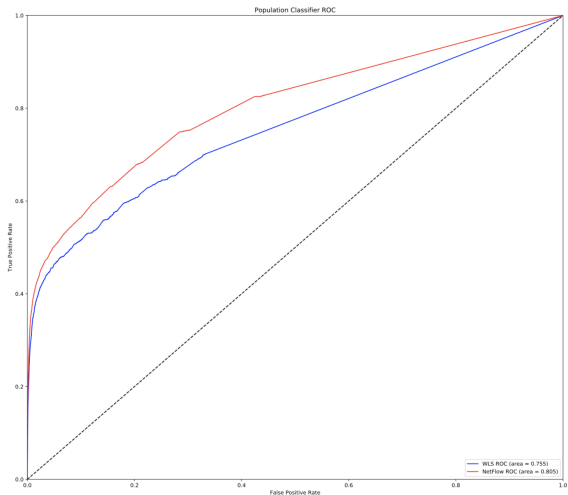


Figure 1: ROC Analysis of Classifiers. Separate classifiers were trained using WLS features and NetFlow features on the population of computers over the course of day 1 and tested on data from day 2. The NetFlow classifier appears to perform better, as expected, considering the ‘new event’ being predicted is a new NetFlow communication. As none of the WLS features relate to communication activity, the performance of the WLS classifier is impressive

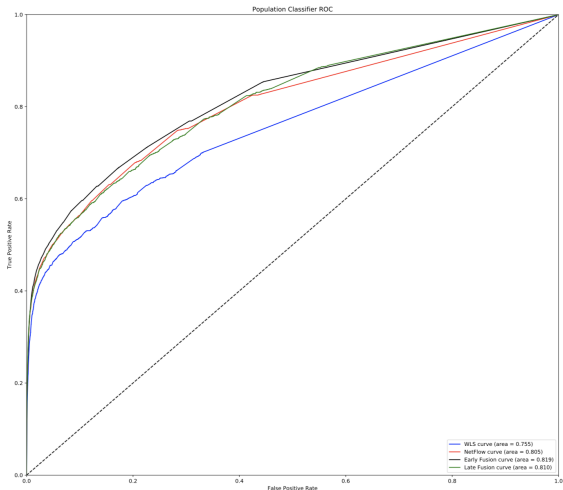


Figure 2: Early vs. Late Fusion. We compare two methods of combining the data sets. For early fusion, we include all the features together when building a classifier. This allows the random forest model to learn interactions between features in the two data sets. For late fusion, we train two separate classifiers and then combine the outputs by averaging their predictions. The early fusion classifier displays the highest performance.

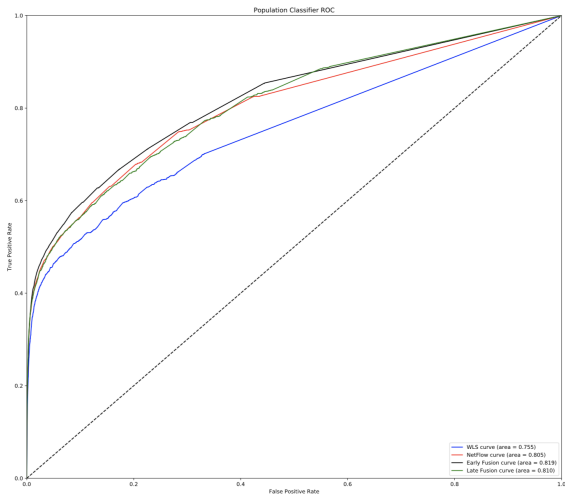
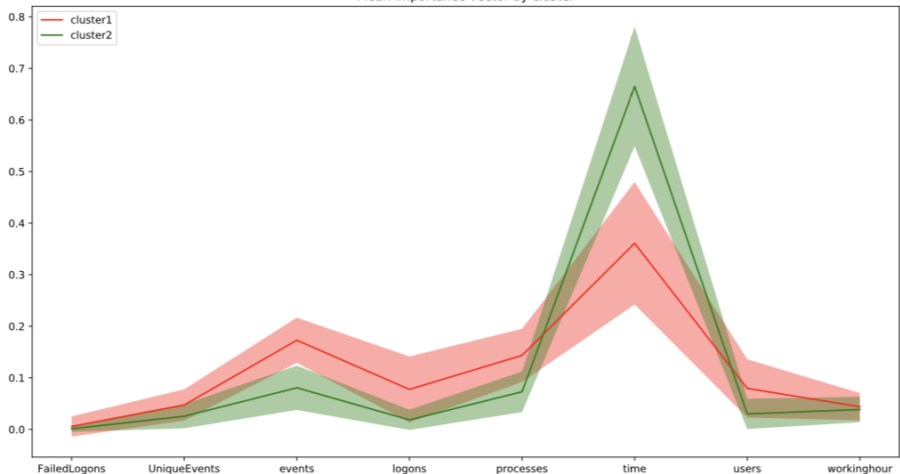


Figure 2: Early vs. Late Fusion. We compare two methods of combining the data sets. For early fusion, we include all the features together when building a classifier. This allows the random forest model to learn interactions between features in the two data sets. For late fusion, we train two separate classifiers and then combine the outputs by averaging their predictions. The early fusion classifier displays the highest performance.

Mean importance vector by cluster



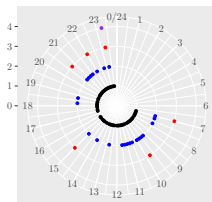
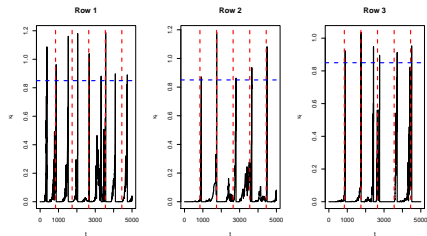
4: Parallel coordinates plot of the mean importance vector of each cluster \pm one standard deviation

Comments

- ▶ Interesting challenges about assessment of performance: report over population, or within device?
- ▶ Need to use a method that allows expiry of the connection list - which we have.
- ▶ Can think about surprise: which predictions were very wrong?
- ▶ How does the output of this process relate to the structure of the graph? To lateral movement?
- ▶ Can imagine many interesting response variables:
 - ▶ NETFLOW based: new server port, new IP/port pair, etc
 - ▶ WLS (Host) based: new user, new service, etcand hence a large suite of interacting analytics.

Back to multinomial, and more fusion

- ▶ Multinomial change detection: use a bound on KL to define a detector. Needs Monte Carlo calibration (one-off) to set control parameters (to fix ARL0).



- ▶ We can, and should, consider using procedures like this to enhance the feature vectors.
- ▶ For example, in considering a device generating new nodes, some of the estimates from the multinomial distribution of server ports, may provide an enhancement to performance.
- ▶ Of course, such “derived” features need to be rapidly computable. So, even if we are doing things with binned data, streaming methods may still be preferred in order to contain memory and compute burden.

4: Conclusion

- ▶ Cyber-analytics provides new challenges and opportunities for statistics and machine learning. The attackers are not going to go away!
- ▶ We are trying to develop tools that are well adapted to the nature of the problem - not trying to develop full solutions. I have tried to show some of our recent and current thinking - papers are available detailing this stuff (and more).

Thank you!

Questions

References

1. Haykin, S. (2002) 'Adaptive filter theory', 4th edition, Prentice Hall.
2. Neil, J., Storlie, C., Hash, C. and Brugh, A. 'Statistical detection of intruders within computer networks using scan statistics', in *Data analysis for network cyber-security*, Imperial College Press.
3. Bodenham, D.A. (2014). 'Adaptive estimation with change detection for streaming data'. PhD Thesis, Department of Mathematics, Imperial College London.
4. Anagnostopoulos, C. (2010) 'A statistical framework for streaming data analysis'. PhD Thesis, Department of Mathematics, Imperial College London.
5. Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M., Pavlidis, D.K. and Hand, D.J., Streaming Gaussian classification using recursive maximum likelihood with adaptive forgetting, *Stat.Anal. Data Mining*, 5(2) (2012), 139-166.
6. Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M. and Hand, D.J. Temporally adaptive estimation of logistic classifiers on data streams. *Adv. Data An. Classif.*, 3(3) (2009), 243-261.
7. Pavlidis, N.G., Tasoulis, D.K., Adams, N.M and Hand, D.J., -perceptron: an adaptive classifier for data streams, *Pattern Recog.*, 44(1) (2011), 78-96.
8. Alexander G. Tartakovsky (2014) 'Rapid Detection of Attacks in Computer Networks by Quickest Change-point Detection Methods'. In *Data analysis for network cyber-security*, Imperial College Press.
9. Bodenham, D.A. and Adams, N.M. (2016) 'Continuous changepoint monitoring of data streams using adaptive estimation'. *Statistics and Computing*, 27(5), 1257–1270.
10. Bodenham, D.A. and Adams N.M. (2014) 'Adaptive change detection for relay-like behavior', IEEE Joint Information and Security Informatics Conference, 2014.
11. Evangelou, M. and Adams N.M. (2016) 'Predictability of NetFlow data', 14th IEEE International Conference on Intelligence and Security Informatics - Cybersecurity and Big Data (IEEE ISI), IEEE, Pages: 67-72.
12. Page, E. (1954) 'Continuous inspection schemes', *Biometrika*, 41, 100-115.
13. Jiang, W., Shu, W. and Apley D.W. (2008) "Adaptive CUSUM procedures with EWMA-based shift estimators. *IEE Transactions*, 40(10), 992–1003.
14. Plasse, J. and Adams, N.M. (2018) "Streaming change point detection for transition matrices", *Technometrics*, in review.