Online Anomaly Detection for Optical Networks

Tarem Ahmed and Mark Coates

McGill University

tahmed@mail.mcgill.ca, coates@tsp.ece.mcgill.ca

Introduction

What is an anomaly?

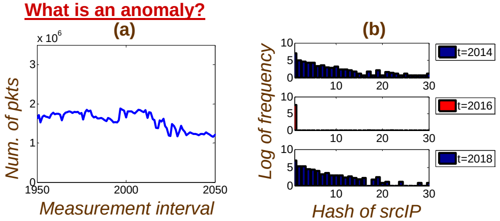

Fig. 1: Fairly stable behavior over time of **(a)** total number of packets in network, but **(b)** drastic change in distribution of source IP within a particular flow at *t* = 2016. Data from NYC core router in Abilene backbone network.

- Network anomalies span wide variety of classes/types. Need online and intelligent, anomaly detection method.
- We propose an **online**, **learning** algorithm, based on the Kernel Recursive Least-Squares (KRLS) algorithm.
- We test on data from Abilene backbone network, and compare with offline, block-based algorithm based on Principal Component Analysis (PCA) [1].

The Architecture

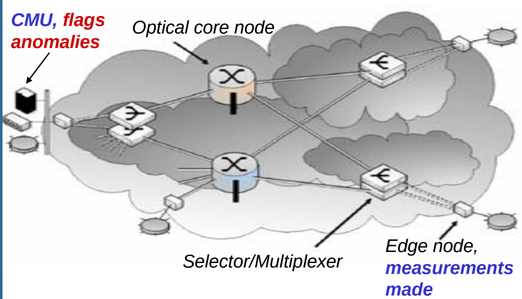

Fig. 2: Traffic statistics collected (in **distributed** manner) at edge nodes, every pre-determined measurement interval. Stats sent to **central monitoring unit (CMU)** which runs **online anomaly detection algorithm**, **raises alarms**.

- Collect following packet header information at edge nodes: **{src edge node, dst edge node, srcIP, dstIP}**.
- **Flow** defined as **{src edge node, dst edge node}** pair.
- $\mathbf{x}_t$ is **Flow Vector**, defined as vector giving number of packets (or bytes) in each flow, normalized, at time *t*.

The Key Idea

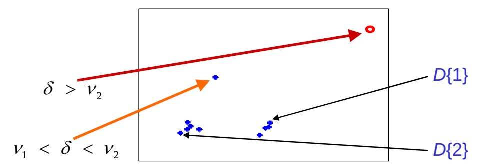

Fig. 3: Simplified depiction of space spanned by 2 sample dictionary elements, ***D*{1}** and ***D*{2}**. δ is distance metric, ν_1 and ν_2 are thresholds where $\nu_1 < \nu_2$. **Anomaly declared** when $\delta > \nu_2$, ***D* expanded** when $\nu_1 < \delta < \nu_2$.

Objective: Build a **dictionary** of flow vectors $D = \{\tilde{\mathbf{x}}_j\}_{j=1}^{m}$, such that mapping onto feature space, $\{\varphi(\tilde{\mathbf{x}}_j)\}_{j=1}^{m}$, forms an ***approximately* linearly independent** basis. φ represents mapping from input space to feature space [2].

The Detection Algorithm

•**Initialize** at t = 1, by entering $\mathbf{x}_1$ into dictionary.

•**Iterate** for t = 2,3,…

Step 1: New data arrive. Evaluate δ_t, the **degree** of **linear dependence** of $\varphi(\mathbf{x}_t)$ on the dictionary at time *t* [2]:

$$\delta_t = \min_{a} \left\| \sum_{j=1}^{m_{t-1}} a_j \phi(\tilde{\mathbf{x}}_j) - \phi(\mathbf{x}_t) \right\|^2 \qquad (1)$$

Step 2: Compare δ_t with thresholds ν_1 and ν_2, where $\nu_1 < \nu_2$:

- If $\delta_t > \nu_2$, new input vector is **very** far away, conclude anomaly. Raise **red alarm**, **no change** to dictionary.
- If $\nu_1 < \delta_t < \nu_2$, new input vector not sufficiently explained by dictionary. **Add** $\mathbf{x}_t$ to dictionary, raise **orange alarm**.
- If $\delta_t < \nu_1$, new input vector falls within normal subspace. **No alarm**, **no change** to dictionary.

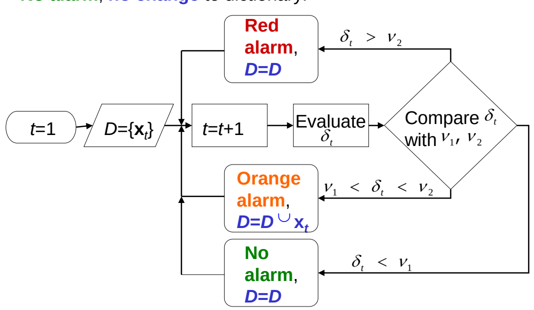

Fig. 4: Outline of online anomaly detection algorithm.

Results

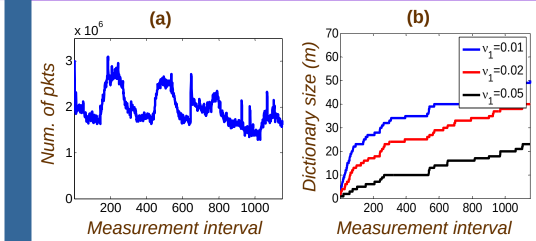

Fig. 5: **(a)** Variation in total number of packets in network; **(b)** growth in *D* for various values of ν_1, with $\nu_2 = 6\nu_1$.

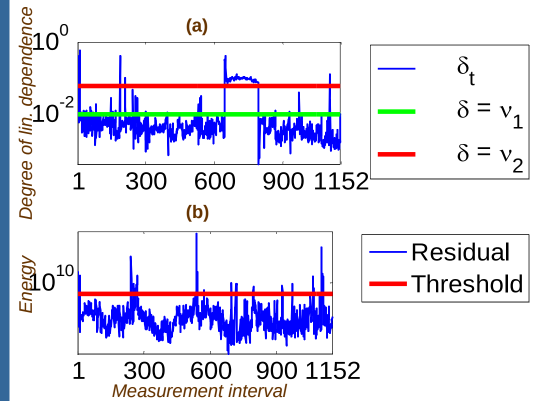

Fig. 6: Comparing (a) δ_t in proposed algorithm with $\nu_1 = 0.01$ and $\nu_2 = 6\nu_1$, with **(b)** energy in residual subspace using block-based PCA from [1]. Spikes represent anomalies.

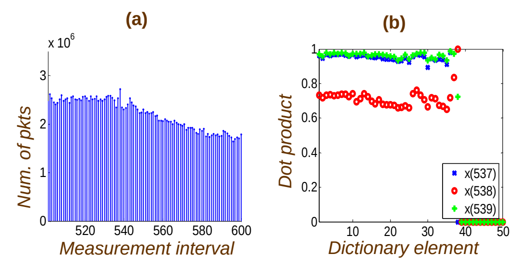

Fig. 7: Example anomaly at *t* = 538. Not easily seen in **(a)** timeseries of packets, but **(b)** obvious by observing inner product of $\mathbf{x}_t$ from each dictionary member.

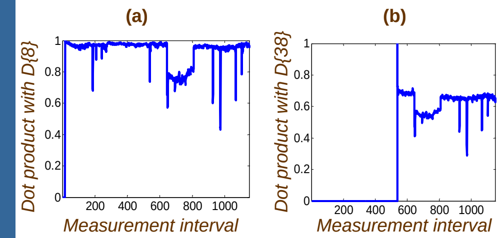

Fig. 8: Progression in time of inner product of $\mathbf{x}_t$ with **(a)** a normal dictionary member, and **(b)** an anomalous flow vector that was admitted to dictionary.

Conclusions and Future Work

- Algorithm is **recursive**, there is **no need to relearn** from scratch when new data arrive.
- **Storage requirement and complexity bounded** by O(m^2), i.e. independent of time.
- **Performance comparable to accepted offline, block-based PCA method** in [1].
- Work in progress includes **controlling dictionary** by enabling dropping of obsolete or anomalous elements; confirming anomaly in case of **orange alarm** if relevant $\mathbf{x}_t$ exhibits continually low inner product value to subsequent input vectors.
- Future work involves letting the **data determine the thresholds** ν_1 and ν_2.

Acknowledgements

R. Summerhill and M. Fullmer at Abilene provided access to raw data.

References

[1] A. Lakhina, M. Crovella and C. Diot, *Diagnosing Network-Wide Traffic Anomalies*, ACM SIGCOMM, Portland, OR, August 2004.

[2] Y. Engel, S. Mannor and R. Meir, *The Kernel Recursive Least Squares Algorithm*, IEEE Trans. on Sig. Proc., 52(8), pp.2275--2285, 2004.