Machine Learning Algorithms for Anomaly Detection in Optical Networks

Tarem Ahmed and Mark Coates
McGill University
tarem.ahmed@mail.mcgill.ca, coates@ece.mcgill.ca

1. Introduction

- We define an anomaly as a rare and short-lived event.
- Anomaly detection involves extracting relevant information from high-dimensional and high-rate network data.
- Nature and normal behaviour of networks change over time. New types of anomalies are regularly discovered. Hence, adaptive and learning algorithms are desired.

Our Contribution:
- We demonstrate the applicability of Machine Learning approaches to anomaly detection in optical networks.
- We present two algorithms:
  - Kernel-based Online Anomaly Detection (KOAD);
  - One-Class Neighbour Machine (OCNM).
- We test the algorithms on a timeseries of *entropies* of the IP packet header fields, from the Abilene network.

2. Kernel-based Online Anomaly Detection (KOAD): Introduction

- Recursive algorithm for online anomaly detection [1], [2].
- Incrementally learns a *dictionary* of input vectors that spans the *region of normality* in a chosen feature space.
- An anomaly is flagged immediately upon encountering a deviation from the norm.
- The dictionary maintained is dynamic and incorporates changes in the normal behaviour of the given network.

Initialization:
- Sequence of multivariate measurements: $\{\mathbf{x}_t\}_{t=1:T}$.
- Choose feature space with associated kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle \quad \text{where} \quad \varphi : \mathbf{x} \in \mathbb{R}^n \rightarrow \varphi(\mathbf{x}) \in H^\infty$$

- Then feature vectors corresponding to normal traffic measurements should *cluster*.

The Dictionary:
- Should be possible to describe *region of normality* in feature space using sparse *dictionary*, $D = \{\tilde{\mathbf{x}}_j\}_{j=1}^{M}$
- Feature vector $\varphi(\mathbf{x}_t)$ is said to be *approximately linearly independent* on $\{\varphi(\tilde{\mathbf{x}}_j)\}_{j=1}^{M}$ if [3]:

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

3. KOAD: The Algorithm

- At timestep *t*, evaluate δ_t, compare with ν_1, ν_2 where $\nu_1 < \nu_2$.
- If $\delta_t > \nu_2$, infer $\mathbf{x}_t$ far from normality: **Red1 Alarm.**
- If $\delta_t < \nu_1$, infer $\mathbf{x}_t$ close to normality: **Green.**
- If $\delta_t > \nu_1$, raise **Orange Alarm** and track the contribution of $\mathbf{x}_t$ in explaining the *l* subsequent arrivals.
- At timestep *t+l* resolve any **Orange Alarm** from timestep *t*. Done by performing a secondary *Usefulness Test* [2], and determining how many of the *l* subsequent arrivals lie close to $\mathbf{x}_t$. We distinguish between cases where:
  - $\mathbf{x}_t$ is an **isolated event** and a **potential anomaly**; or
  - $\mathbf{x}_t$ represents a **migration of region of normality**.

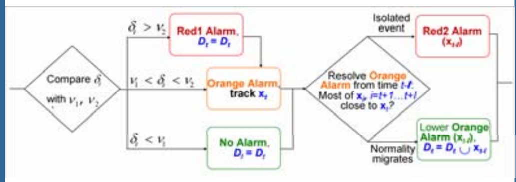

Fig. 1: Flow chart of operations performed at any timestep t by KOAD algorithm. For details, see [1].

4. One-Class Neighbor Machine (OCNM)

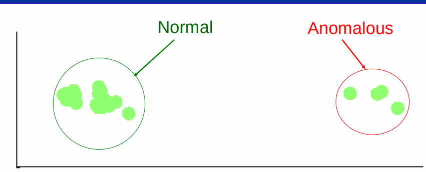

Fig. 2: 2-D Isomap of CHIN-LOSA backbone flow, *Entropy(srcIP)*.

- Region of normality should correspond to a Minimum Volume Set (MVS).
- OCNM for estimating MVS proposed in [4].
- Requires choice of sparsity measure, *g*.
  - Examples: *k*-th nearest-neighbour distance, average of first *k* nearest-neighbour distances.
- Sorts list of *g*, identifies points that lie inside MVS using pre-specified fraction μ.

5. Data

Data collection:
- 11 core routers, 121 *backbone flows*
- 4 main pkt headers collected: *(srcIP, dstIP, srcPort, dstPort)*

Fig. 3: Abilene weathermap. Source: Indiana University.

Data processing:
- Construct header histogram
- Calculate header *entropies* for each backbone flow, at each timestep
- Variations in entropies (distributions) reveal many anomalies [5].

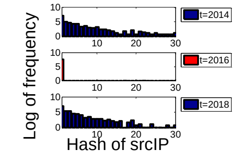

Fig. 4: Example anomaly. Distribution of srcIP exhibits sudden and short-lived change.

6. Results

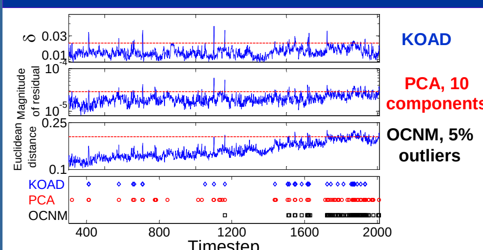

Fig. 5: Top three panels show the variation in the KOAD δ_t, the PCA magnitude of residual with 10 principal components assigned to the normal subspace, and the OCNM k-th nearest neighbour Euclidean distance, versus time. Bottom panel compares the anomalies flagged by each algorithm,.

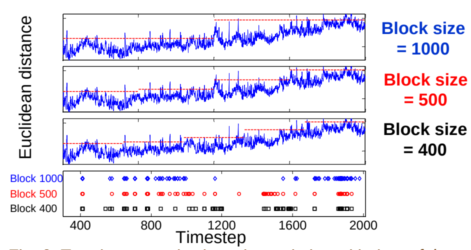

Fig. 6: Top three panels show the variation with time of the OCNM k-th nearest-neighbour Euclidean distance, using block-sizes of 1000, 500 and 400 timesteps. Bottom panel compares the anomalies flagged in each case.

7. Discussion

- We validate recursive **KOAD** and block-based **OCNM** against the block-based Principal Component Analysis **(PCA)** anomaly detection method from [6].
- **KOAD** is run using a Gaussian kernel, **PCA** with 10 principal components assigned to the normal subspace, and **OCNM** using k = 50 and μ = 0.95.
- The spikes in Fig. 5(a-c) indicate that all three algorithms signal an overlapping set of anomalies.
- Fig. 5(c) indicates that the **OCNM** *k*-th nearest-neighbour distance metric exhibits upward trend. Suggests that 2000-timestep block size is too large.
- Fig. 6 compares **OCNM** results for various block sizes.

8. Conclusions and Future Work

- Preliminary results indicate the potential of Machine Learning approaches in quick anomaly detection.
- Computations must be distributed to minimize communication costs.
- Complexity must be independent of time for online application. **KOAD** complexity is, **OCNM** is not.

9. Acknowledgements

Thanks to Anukool Lakhina for providing the Abilene dataset. This research was supported by the Canadian Natural Sciences and Engineering Research Council of (NSERC) and industrial and government partners, through the Agile All-Photonic Networks (AAPN) research network.

10. References

[1] T. Ahmed, M. Coates and A. Lakhina, "Multivariate online anomaly detection using kernel recursive least squares," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, to appear.

[2] T. Ahmed, B. Oreshkin and M. Coates, "Machine learning approaches to network anomaly detection," in *Proc. USENIX Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML)*, Cambridge, MA, Apr. 2007.

[3] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least squares algorithm," *IEEE Trans. Signal Proc.*, vol. 52, No. 8, pp. 2275–2285, Aug. 2004.

[4] A. Muñoz and J. Moguerza, "Estimation of high-density regions using one-class neighbor machines," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol 28, num 3, pp 476--480, Mar. 2006.

[5] A. Lakhina, M. Crovella and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. ACM SIGCOMM*, Philadelphia, PA, Aug. 2005.