Machine Learning Approaches to Network Anomaly Detection

Tarem Ahmed, Boris Oreshkin and Mark Coates
Department of Electrical and Computer Engineering
McGill University
Montreal, QC, Canada
Email: tarem.ahmed@mail.mcgill.ca, boris.oreshkin@mail.mcgill.ca, coates@ece.mcgill.ca

***Abstract*— Networks of various kinds often experience anomalous behaviour. Examples include attacks or large data transfers in IP networks, presence of intruders in distributed video surveillance systems, and an automobile accident or an untimely congestion in a road network. Machine learning techniques enable the development of anomaly detection algorithms that are non-parametric, adaptive to changes in the characteristics of normal behaviour in the relevant network, and portable across applications. In this paper we use two different datasets, pictures of a highway in Quebec taken by a network of webcams and IP traffic statistics from the Abilene network, as examples in demonstrating the applicability of two machine learning algorithms to network anomaly detection. We investigate the use of the block-based One-Class Neighbour Machine and the recursive Kernel-based Online Anomaly Detection algorithms.**

I. Introduction

A network anomaly is a sudden and short-lived deviation from the normal operation of the network. Some anomalies are deliberately caused by intruders with malicious intent such as a denial-of-service attack in an IP network, while others may be purely an accident such as an overpass falling in a busy road network. Quick detection is needed to initiate a timely response, such as deploying an ambulance after a road accident, or raising an alarm if a surveillance network detects an intruder.

Network monitoring devices collect data at high rates. Designing an effective anomaly detection system consequently involves extracting relevant information from a voluminous amount of noisy, high-dimensional data. It is also important to design distributed algorithms as networks operate under bandwidth and power constraints and communication costs must be minimised.

Different anomalies exhibit themselves in network statistics in different manners, so developing general models of normal network behaviour and of anomalies is difficult. Model-based algorithms are also not portable across applications, and even subtle changes in the nature of network traffic or the monitored physical phenomena can render the model inappropriate. Non-parametric, learning algorithms based on machine learning principles are therefore desirable as they can learn the nature of normal measurements and autonomously adapt to variations in the structure of "normality".

A. Related Work and Contribution

Most methods of network anomaly detection are based on network traffic models. Brutlag uses as an extension of the Holt-Winters forecasting algorithm, which supports incremental model updating via exponential smoothing [1]. Hajji uses a Gaussian mixture model, and develops an algorithm based on a stochastic approximation of the Expectation-Maximization (EM) algorithm to obtain estimates of the model parameters [2]. Yamanishi et al. also assume a hierarchical structure of Gaussian mixtures in developing the "SmartSifter" tool, but uses different algorithms for updating the model parameters [3]. They use a variant of the Laplace law in the discrete domain, and a modified version of the incremental EM algorithm in the continuous domain. They test their algorithm to detect network intrusion on the standard ACM KDD Cup 1999 dataset. Lakhina et al. apply Principal Component Analysis (PCA) to separate IP network data into disjoint "normal" and "anomalous" subspaces, and signal an anomaly when the magnitude of the projection onto the anomalous subspace exceeds a threshold [4]–[6]. Huang et al. build on Lakhina's centralised PCA method of anomaly detection from [6], and develop a framework where local PCA analysis and stochastic matrix perturbation theory is used to develop an adaptive, distributed protocol [7].

Researchers have recently begun to use machine learning techniques to detect outliers in datasets from a variety of fields. Gardener et al. use a One-Class Support Vector Machine (OCSVM) to detect anomalies in EEG data from epilepsy patients [8]. Barbará et al. have proposed an algorithm to detect outliers in noisy datasets where no information is available regarding ground truth, based on a Transductive Confidence Machine (TCM) [9]. Transduction is an alternative to induction, in that instead of using all the data points to induce a model, one is able to use a small subset of them to estimate unknown properties of test points. Ma and Perkins present an algorithm using support vector regression to perform online anomaly detection on timeseries data in [10]. Ihler et al. present an adaptive anomaly detection algorithm that is based on a Markov-modulated Poisson process model, and use Markov Chain Monte Carlo methods in a Bayesian approach to learn the model parameters [11].

An example of a machine learning approach to network anomaly detection is the time-based inductive learning machine (TIM) of Teng et al. [12]. Their algorithm constructs a set of rules based upon usage patterns. An anomaly is signalled when the premise of a rule occurs but the conclusion does not follow. Singliar and Hauskrecht use a support vector machine to detect anomalies in road traffic [13]. They use statistics collected by a sophisticated network of sensors including microwave loops and lasers, and design a detector for road traffic incidents.

Our objective in this paper is to show the applicability and need for learning algorithms in detecting anomalous behaviour in a distributed set of network measurements. From the wide variety of machine learning techniques available, we choose the One Class Neighbor Machine (OCNM) proposed by Muñoz and Moguerza in [14], and the recursive Kernel-based Online Anomaly Detection (KOAD) algorithm that we developed in [15]. We present our case via two examples: sequences of images from Transports Quebec's camera network, and IP timeseries data from the Abilene backbone network. We demonstrate that both algorithms are effective in detecting anomalies and motivate the development of more advanced, fully adaptive and fully distributed, learning algorithms.

B. Organization of Paper

The rest of this paper is organized as follows. Section II defines the problem we address. Section III describes the Transports Quebec and Abilene datasets and Section IV reviews the OCNM and KOAD algorithms. Section V presents our results and Section VI summarises our conclusions and discusses the need for distributed, learning algorithms for network anomaly detection.

II. Problem Statement

The anomaly detection problem can be formulated as follows. A continuous stream of data points $\mathbf{x} \in \mathbb{R}^k$ constitutes a collection of measurements $\{\mathbf{x}_t\}_{t=1}^T$ governed by a probability distribution P. Although measurements correspond to certain physical events in the event space $\mathcal{S}$, the mapping $f : \mathcal{S} \rightarrow \mathbb{R}^k$ between them may not be known. We assume that $\mathcal{S}$ can be divided into two subspaces corresponding to normal and anomalous physical conditions. In many practical situations it is of interest to infer the membership of an event in a particular subspace using the corresponding measurement. As the probability distribution P governing measurements is unknown, some mechanism should facilitate learning its volumetric representation from the collection $\{\mathbf{x}_t\}_{t=1}^T$. A general approach to the aforementioned problem of learning such a representation consists of constructing a Minimum Volume Set (MVS) with probability mass $\beta \in (0, 1)$ with respect to distribution P for a volume measure ξ [16]:

$$G_\beta^* = \arg\min\{\xi(G) : P(G) \geq \beta, G \text{ measurable}\}. \quad (1)$$

For a recent review of practical methods for estimating G_β^*, see [17]. Online estimation of minimum volume sets satisfying (1) allows the identification of high-density data regions where the mass of P is concentrated. Data points lying outside these regions and corresponding physical events are then declared anomalous.

Real multidimensional data exhibit distributions which are highly sparse. Moreover, distributions of raw data may lack invariance with respect to generating events. That is, physical events pertaining to the same region in $\mathcal{S}$ may generate measurements in completely different regions of $\mathbb{R}^k$. Therefore, it is often desirable to reduce the dimensionality of raw data via some feature extraction mechanism $g : \mathbb{R}^k \rightarrow \mathbb{R}^l$ where $l < k$, that is robust to sparsity and variance induced by the transition $f : \mathcal{S} \rightarrow \mathbb{R}^k$. We then construct a minimum volume set from the features and not from the raw data.

III. Data

We use two different datasets to advocate the applicability of machine learning algorithms to network anomaly detection.

1) Transports Quebec dataset: Transports Quebec maintains a set of webcams over its major roads [18]. These cameras record still images every five minutes. We collected images recorded by 6 cameras over a period of four days (Sep. 30 to Oct. 03, 2006) on Quebec's Autoroute 20. Each 5-minute interval constitutes a *timestep*.

Anomaly detection in a sequence of images relies mainly on the extraction of appropriate information from the sequence. There are two fundamental reasons for this. First, the large dimensionality inherent to image processing leads to dramatic increase in implementation costs. Second, large variation in operating conditions such as brightness and contrast (which are subject to the time of day and weather conditions) and colour content in the images (which is subject to season), can cause abrupt and undesirable changes in the raw data.

We decided to use the discrete wavelet transform (DWT) to process the images. The DWT is known for its ability to extract spatially localised frequency information. We perform the two-dimensional DWT on every image and average the energy of transformation coefficients within each subband to achieve approximate shift invariance of the feature extractor. We expect that the appearance of a novel image in the sequence will manifest itself as a sudden change in the power in the frequency content of the vector of subband intensities. At each timestep, we construct a *wavelet feature vector* from each image obtained by each camera node.

2) Abilene dataset: The Abilene backbone network in the US contains 11 core routers. A *backbone flow* consists of packets entering the Abilene network at one particular core router and exiting at another. The data constitute a timeseries of the *entropies* of the 4 main packet header fields (source IP address, destination IP address, source port number and destination port number) in each of $11 \times 11 = 121$ backbone flows pertaining to each timestep. The entropy for each backbone flow, at each timestep, for each header field, is computed after constructing an empirical histogram of the relevant header field distribution for that backbone flow during that timestep. The four component entropies are finally concatenated to obtain an entropy timeseries of the 121 backbone flows. Physical anomalous phenomena cause changes in the distributions of packets belonging to the responsible backbone flow, and Lakhina et al. showed in [6] that these changes are best captured by changes in the entropies. The duration of a timestep is again five minutes, and the length of the Abilene timeseries is 2016 timesteps pertaining to one week (Dec. 15 to 21, 2003).

IV. Anomaly Detection

A. One-Class Neighbor Machine (OCNM)

The OCNM algorithm proposed by Muñoz and Moguerza provides an elegant means for estimating minimum volume sets [14]. It assumes a sample set $\mathcal{S}$ comprising T, F-dimensional data points, $\{\mathbf{x}_t\}_{t=1}^{T}$. The algorithm requires the choice of a sparsity measure, denoted by g. Example choices of a sparsity measure are the k-th nearest neighbour Euclidean distance and the average of the first k nearest-neighbour distances. The OCNM algorithm sorts the values of the g measure for the set of points $\mathcal{S}$, and subsequently identifies those points that lie inside the minimum volume set (MVS) with the smallest sparsity measure g, up to a specified fraction μ of the number of points in $\mathcal{S}$.

If the k-th nearest neighbour distance function is used as the sparsity measure, the OCNM algorithm involves calculating the distance from every point $\mathbf{x}_t$ to every *other* point in the sample set. As each point is F-dimensional and there are T timesteps, the complexity is $O(T^2F)$.

B. Kernel-based Online Anomaly Detection (KOAD)

Consider a set of multivariate measurements $\{\mathbf{x}_t\}_{t=1}^{T}$. In an appropriately chosen feature space $\mathcal{F}$ with an associated kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, the features corresponding to the normal traffic measurements should *cluster*. Then it should be possible to describe the region of normality using a relatively small *dictionary* of *approximately* linearly independent elements $\{\phi(\tilde{\mathbf{x}}_j)\}_{j=1}^{M}$ [19]. Here $\{\tilde{\mathbf{x}}_j\}_{j=1}^{M}$ represent those $\{\mathbf{x}_t\}_{t=1}^{T}$ that are entered into the dictionary and we expect the size of the dictionary (M) to be much less than T, leading to computational and storage savings. Feature vector $\phi(\mathbf{x}_t)$ is said to be *approximately* linearly dependent on $\{\phi(\tilde{\mathbf{x}}_j)\}_{j=1}^{M}$ with approximation threshold ν, if the projection error δ_t satisfies the following criterion:

$$\delta_t = \min_{\mathbf{a}} \left\| \sum_{j=1}^{M} a_j \phi(\tilde{\mathbf{x}}_j) - \phi(\mathbf{x}_t) \right\|^2 < \nu. \tag{2}$$

where $\mathbf{a} = \{a_j\}_{j=1}^{M}$ is the optimal coefficient vector.

The Kernel-based Online Anomaly Detection (KOAD) algorithm operates at each timestep t on a measurement vector $\mathbf{x}_t$. It begins by evaluating the error δ_t in projecting the arriving $\mathbf{x}_t$ onto the current dictionary (in the feature domain). This error measure δ_t is then compared with two thresholds ν_1 and ν_2, where $\nu_1 < \nu_2$. If $\delta_t < \nu_1$, we infer that $\mathbf{x}_t$ is sufficiently linearly dependent on the dictionary, and represents normal traffic. If $\delta_t > \nu_2$, we conclude that $\mathbf{x}_t$ is far away from the realm of normal behaviour, and immediately raise a "Red1" alarm to signal an anomaly.

If $\delta_t > \nu_1$, we infer that $\mathbf{x}_t$ is sufficiently linearly independent from the dictionary to be considered an unusual event. It may indeed be an anomaly, or it may represent an expansion or migration of the space of normality. In this case, we do the following: raise an "Orange" alarm, keep track of the contribution of the relevant input vector $\mathbf{x}_t$ in explaining subsequent arrivals for ℓ timesteps, and then make a firm decision on it.

At timestep $t + \ell$, we re-evaluate the error δ in projecting $\mathbf{x}_t$ onto dictionary $\mathcal{D}_{t+\ell}$ corresponding to timestep $t+\ell$. Note that the dictionary may have changed between timesteps t and $t+\ell$, and the value of δ at this re-evaluation may consequently be different from the δ_t at timestep t. If the value of δ after the re-evaluation is found to be less than ν_1, we lower the orange alarm and keep the dictionary unchanged.

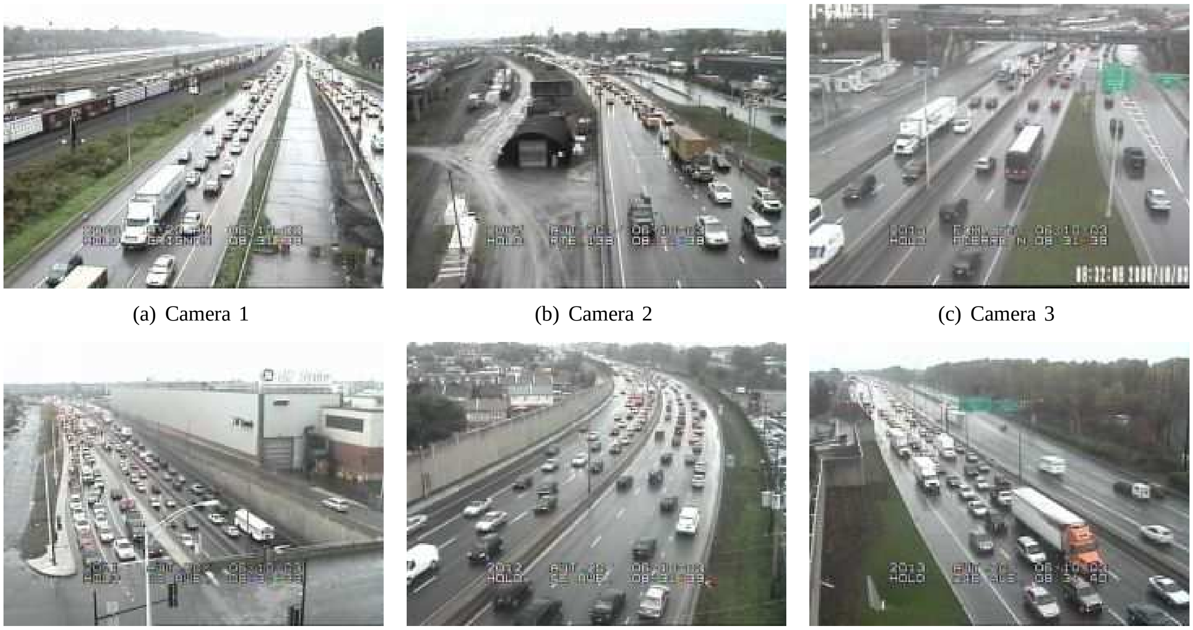
If the value of δ is found instead to be greater than ν_1 after the re-evaluation at timestep $t + \ell$, we perform a secondary "usefulness" test to resolve the orange alarm. The usefulness of $\mathbf{x}_t$ is assessed by observing the kernel values of $\mathbf{x}_t$ with $\mathbf{x}_i$, $i = t + 1, \ldots, t + \ell$. If a kernel value is high (greater than a threshold d), then $\phi(\mathbf{x}_t)$ is deemed close to $\phi(\mathbf{x}_i)$. If a significant *number* of the kernel values are high, then $\mathbf{x}_t$ cannot be considered anomalous; normal traffic has just migrated into a new portion of the feature space and $\mathbf{x}_t$ should be entered into the dictionary. Contrarily if almost all kernel values are small, then $\mathbf{x}_t$ is a reasonably isolated event, and should be heralded as an anomaly. We evaluate:

$$\left[\sum_{i=t+1}^{t+\ell} \mathbb{I}(k(\mathbf{x}_t, \mathbf{x}_i) > d) \right] > \epsilon\ell, \tag{3}$$

where $\mathbb{I}$ is the indicator function and $\epsilon \in (0, 1)$ is a selected constant. In this manner, by employing the secondary "usefulness test", we are able to distinguish between an arrival that is an anomaly, from one that is a result of a change in the region of normality. If (3) evaluates true, then we lower the relevant orange alarm to green (no anomaly) and add $\mathbf{x}_t$ to the dictionary. If (3) evaluates false, we elevate the relevant orange alarm to a "Red2" alarm.

The KOAD algorithm also deletes obsolete elements from the dictionary as the region of normality expands or migrates, thereby maintaining a small dictionary. In addition, it incorporates exponential forgetting so that the impact of past observations is gradually reduced.

Assuming a dictionary containing m elements, the computational complexity of the KOAD algorithm is $\mathrm{O}(m^2)$ for every standard timestep, and $\mathrm{O}(m^3)$ on the rare occasions when an element removal occurs. The KOAD complexity is thus independent of time, making the algorithm naturally suited to online applications. Our experiments have shown that high sparsity levels are achieved in practice, and the dictionary size does not grow indefinitely. See [15] for details regarding the KOAD algorithm.



(a) Camera 1 (b) Camera 2 (c) Camera 3

(d) Camera 4 (e) Camera 5 (f) Camera 6

Fig. 1. Pictures from the Transports Quebec camera network corresponding to timestep $t = 368$. Congestion is evident in all the images at this timestep. Both OCNM and KOAD flagged this timestep as anomalous for most representative parameter settings, when run in a distributed fashion in each of the 6 nodes.

C. Monitoring Architecture

We propose two monitoring architectures: a distributed approach and a centralised approach. In the distributed architecture, the detection algorithms are locally run at each node. After each timestep, each node makes a local decision regarding the presence or absence of an anomaly at that timestep, and transmits a *binary* result to the Central Monitoring Unit (CMU). The CMU then makes a decision on the location of an anomaly in time and space, if at least n of the c nodes individually signalled an anomaly. The idea behind this n-out-of-c detection scheme is that in many applications, such as in a road network, *bona fide* anomalies such as an untimely traffic congestion are simultaneously evident to multiple nodes. Individual flags are likely to be caused by comparably less important and independent events such as a single camera malfunctioning.

In the centralised architecture, all measurements are communicated to the Central Monitoring Unit. The CMU then runs the detection algorithm. The centralised approach is often desirable and necessary to detect anomalies that exhibit themselves through distributed changes in the global measurement vector. The works of Lakhina et al. have shown that traffic volume distributions in large backbone IP networks exhibit a covariance structure, and detecting a break in this structure enables one to unearth a wide range of anomalies [4]–[6].

V. Experimental Results

A. Transports Quebec Data Analysis

In this section, we study the effectiveness of OCNM and KOAD in detecting unusual events in Quebec's road network. We apply OCNM and KOAD to image sequences from Transports Quebec webcam network. This is an example application of the distributed monitoring architecture described in Section IV-C. Our data consists of a series of 444 timesteps corresponding to daylight hours during the Sep. 30 to Oct. 03, 2006 period. We use the averaged energy of DWT coefficients (see Section III-.1) from 6 subbands from each of the 6 cameras (nodes). In our application with $c = 6$ cameras, we used $n = 3$ as the central decision rule.

We present illustrative images from the 6-camera network corresponding to a traffic congestion in Fig. 1. Given the normal flow of traffic during the length of our dataset, short periods of congestion constitute an example of a road network anomaly. This timestep was flagged in all 6 nodes by both OCNM and KOAD as anomalous, for most representative algorithm parameter settings.

Fig. 2(top panel) shows the results of wavelet analysis of the image sequence from one camera. We selected one of the six cameras for preliminary assessment of feature extraction quality. It can be seen that the high-frequency components of the feature vector show the expected variation in the vicinity of traffic jams. Abrupt changes to the position of a camera generate sudden spikes in the feature vector components. Fig. 2 (middle panel) shows the distance measures obtained using the OCNM algorithm with k set to 50 and using $\mu = 0.90$ to signal the 10% outliers. Fig. 2(bottom panel) shows the variations in KOAD projection error δ_t. We ran the KOAD algorithm here with the thresholds $\nu_1 = 0.25$ and $\nu_2 = 0.50$, a Gaussian kernel having standard deviation 0.01, and default settings for all other parameters (see [15]) which includes the orange alarm being resolved after 20 timesteps (i.e. $\ell = 20$). We begin our analysis of Fig. 2 at $t = 51$ with the previous

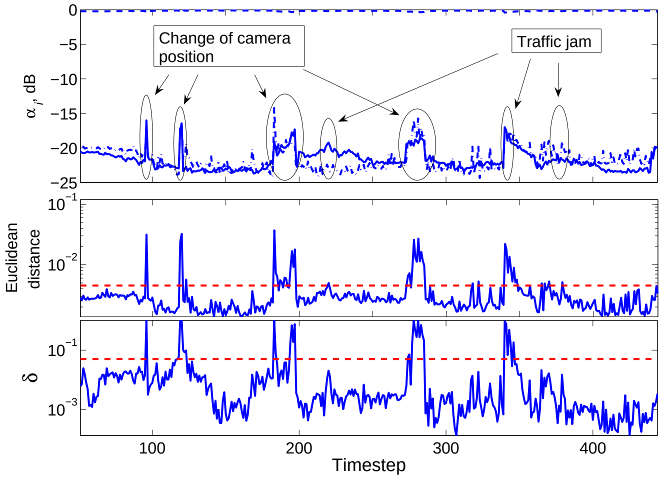


Fig. 2. Top panel: Annotated plot of average wavelet coefficients in subbands evolving through time. Approximation coefficients are shown by dashed line. Solid and dashed lines indicate transform levels 1 and 6 corresponding to highest and lowest frequency content of the DWT, respectively. Middle panel: OCNM using kth nearest-neighbour distance, with dashed line indicating 90% MVS threshold. Bottom panel: KOAD projection error δ_t with dashed line indicating lower threshold ν_1. Transports Quebec dataset, Camera 6.

timesteps constituting the training period in this application.

Fig. 3 presents the receiver operating characteristics (ROC) curves showing the variation in the probability of detection (P_D) with the probability of false alarms (P_{FA}), for the OCNM and KOAD algorithms applied to the Transports Quebec dataset. In our experiments, we used OCNM with the nearest-neighbour parameter k set to 50, and varying μ from 0.50 to 0.95 to signal between 50% and 5% outliers. We ran KOAD with the thresholds set to $\nu_1 = 0.00001, \nu_2 = 0.05$, and using a Gaussian kernel where the standard deviation of the kernel function is varied between 0.002 and 0.020. The other KOAD parameters are retained at their default values (see [15]), with an orange alarm resolved after 20 timesteps (i.e. $\ell = 20$).

Although our experiments were performed on a limited data set, this result provides a preliminary assessment of the anomaly detection algorithms based on wavelet feature extraction mechanism and machine learning data clustering approaches. It can be clearly seen from Fig. 3 that the KOAD detector outperforms the OCNM detector.

B. Abilene Network Data Analysis

In this subsection we present the results of applying OCNM and KOAD to the Abilene dataset. Here we want to also detect those anomalies that cause sudden changes in the overall distribution of traffic over the network, as opposed to affecting a single link, during a particular timestep. Thus in this application we implement the centralised architecture proposed in Section IV-C. For discussions on the wide range of anomalies seen in IP networks, refer to the works of Lakhina et al. [4]–[6]. Here we also compare our results with those obtained by Lakhina et al. using the PCA subspace method of anomaly detection.

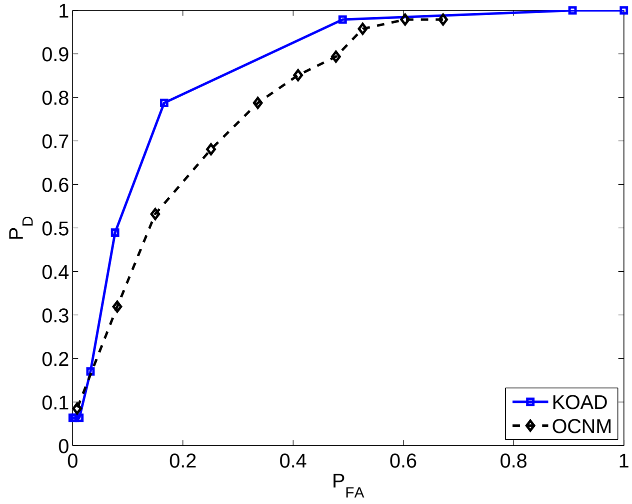


Fig. 3. ROC curves showing variation in probability of detection (P_D) with probability of false alarms (P_{FA}) for distributed anomaly detection based on wavelet feature extraction. Solid line shows KOAD while dashed line shows OCNM data clustering. Transports Quebec dataset.

Fig. 4(a) shows the variations in δ_t obtained using the KOAD algorithm with $\nu_1 = 0.01, \nu_2 = 0.02$, a Gaussian kernel of standard deviation 0.6, and default settings for all other parameters (see [15]). We start our analysis at $t = 301$, with the previous timesteps disregarded as constitute the training period in this application. Fig. 4(b) shows the magnitude of the energy in the residual components using the PCA subspace method of anomaly detection [6]. We used 10 principle components for the normal subspace, in accordance with [6]. Fig. 4(c) shows the distance measures obtained using OCNM with $k = 50$, together with the threshold indicating the 95% minimum volume set. The spike positions in Figs. 4(a-c) indicate the anomalies signalled by KOAD, PCA and OCNM, respectively. Fig. 4(d) isolates for comparison the positions of the anomalies detected by each individual algorithm.

It is evident from Fig. 4(c) that the OCNM k-th nearest neighbour distance metric experiences an upward trend during the one-week period. This phenomenon was observed for values of k that ranged from 10 to 200. Although the positions of the spikes (with respect to the immediate surrounding timesteps) in Fig. 4(c) largely correspond with those in Fig. 4(a-b), we see that most of the outliers signalled by OCNM lie in the latter part of the plot.

The increasing distance metric suggests that the space of normal traffic expands over the recorded time period. The KOAD algorithm is best able to detect anomalies in such a situation, as the dictionary in this algorithm is *dynamic*, with obsolete elements being removed and new, more relevant elements added as necessary. Indeed, we noticed in our experiments with this particular dataset that the dictionary members change significantly over the reported period.

Fig. 4(c) also argues the need for a sequential or block-based version of OCNM where outliers may be incrementally reported after every timestep or block of timesteps. When we ran OCNM on the first 1000 data points only, it flagged the

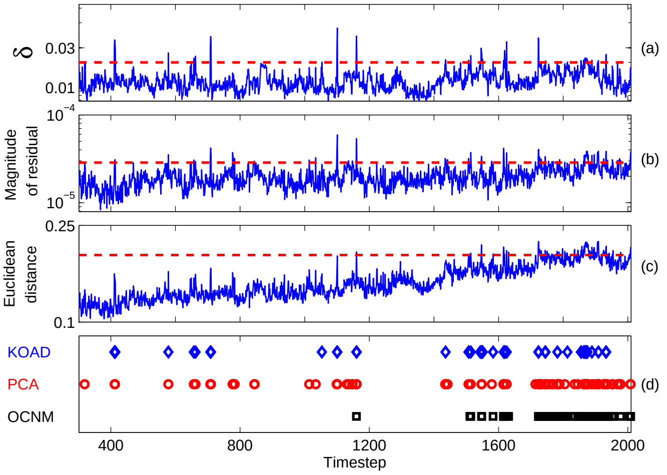


Fig. 4. (a) KOAD projection error δ, with dashed line indicating lower threshold ν_1; (b) magnitude of PCA projection onto the residual subspace, with dashed line indicating the associated Q-statistic threshold [6]; (c) OCNM using kth nearest-neighbour Euclidean distance, with dashed line indicating the 95% MVS threshold; and (d) positions at which anomalies are detected by (◇) KOAD, (o) PCA, and (□) OCNM, all as functions of time. Abilene dataset.

same anomalies as KOAD.

VI. Conclusion

Our preliminary results of the application of machine learning techniques to network anomaly detection indicate their potential and highlight the areas where improvement is required. The non-stationary nature of the network measurements, be they network traffic metrics or recordings of physical phenomena, makes it imperative that the algorithms be able to adapt over time. To make the algorithms portable to different applications and robust to diverse operating environments, all parameters must be learned and autonomously set from arriving data. The algorithms must be capable of running in real-time despite being presented with large volumes of high-dimensional, noisy, distributed data. This means that the methods must perform sequential (recursive) calculations with the complexity at each timestep being independent of time. The computations must be distributed amongst the nodes in the network, and communication, which consumes network resources and introduces undesirable latency in detection, must be minimised.

The KOAD algorithm satisfies some of these requirements, being recursive in nature and adapting to changing data streams. However, in its current form it does not learn appropriate parameter settings, exhibits sensitivity to threshold choices, and has no natural distributed form. The distributed architecture we have described in this paper is a sub-optimal system. Our current and future research focuses on rectifying these deficiencies in the KOAD approach and exploring other promising learning-based alternatives to the network anomaly detection challenge.

Acknowledgements

The authors thank Sergio Restrepo for processing the Transports Quebec data and Anukool Lakhina for providing the Abilene dataset.

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and industrial and government partners, through the Agile All-Photonic Networks (AAPN) research network. The research was also supported by Mathematics of Information Technology and Complex Systems (MITACS) under the NSERC Network of Centres of Excellence (NCE) program.

References

[1] J. Brutlag, "Aberrant behavior detection in time series for network monitoring," in *Proc. USENIX System Admin. Conf. (LISA)*, New Orleans, LA, Dec. 2000.

[2] H. Hajji, "Statistical analysis of network traffic for adaptive faults detection," *IEEE Trans. Neural Networks*, vol. 16, no. 5, pp. 1053–1063, Sep. 2005.

[3] K. Yamanish, J.-I. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275–300, May 2004.

[4] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. ACM SIGMETRICS*, New York, NY, Jun. 2004.

[5] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004.

[6] ——, "Mining Anomalies Using Traffic Feature Distributions," in *Proc. ACM SIGCOMM*, Philadelphia, PA, Aug. 2005.

[7] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, and N. Taft, "In-network PCA and anomaly detection," in *Advances in Neural Information Processing Systems*, 19th ed., B. Schölkopf, J. Platt and T. Hoffman, Ed. Cambridge, MA: MIT Press, 2007, to appear.

[8] A. Gardner, A. Krieger, G. Vachtsevanos, and B. Litt, "One-class novelty detection for seizure analysis from intracranial EEG," *J. Machine Learning Research (JMLR)*, vol. 7, pp. 1025–1044, Jun. 2006.

[9] D. Barbará, C. Domeniconi and J. Rogers, "Detecting outliers using transduction and statistical testing," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, PA, Aug. 2003.

[10] J. Ma and S. Perkins, "Online novelty detection on temporal sequences," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Washington, DC, Aug. 2003.

[11] A. Ihler, J. Hutchins, and P. Smyth, "Adaptive event detection with time-varying Poisson processes," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, PA, Aug. 2006.

[12] H. Teng, K. Chen, and S. Lu, "Adaptive real-time anomaly detection using inductively generated sequential patterns," in *Proc. IEEE Comp. Soc. Symp. Research in Security and Privacy*, Oakland, CA, May 1990.

[13] T. Singliar and M. Hauskrecht, "Towards a learning traffic incident detection system," in *Proc. Workshop on Machine learning Algorithms for Surveillance and Event Detection*, Pittsburgh, PA, Jun. 2006.

[14] A. Muñoz and J. Moguerza, "Estimation of high-density regions using one-class neighbor machines," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 476–480, Mar. 2006.

[15] 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.

[16] M. Davenport, R. Baraniuk, and C. Scott, "Learning minimum volume sets with support vector machines," in *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, Maynooth, Ireland, Sep. 2006.

[17] C. Scott and R. Nowak, "Learning minimum volume sets," *J. Machine Learning Research (JMLR)*, vol. 7, pp. 665–704, Apr. 2006.

[18] Transports Quebec. Organization webpage. [Online]. Available: http://www.mtq.gouv.qc.ca/en/information/cameras/montreal/index.asp

[19] 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.