Sequential Anomaly Detection in Wireless Sensor Networks and Effects of Long Range Dependant Data

Shanshan Zheng
Department of Electrical and Computer Engineering and the Institute of Systems Research, University of Maryland, College Park, Maryland, USA

John S. Baras
Department of Electrical and Computer Engineering and the Institute of Systems Research, University of Maryland, College Park, Maryland, USA

Abstract: Anomaly detection is important for the correct functioning of wireless sensor networks. Recent studies have shown that node mobility along with spatial correlation of the monitored phenomenon in sensor networks can lead to observation data that have long range dependency, which could significantly increase the difficulty of anomaly detection. In this paper, we develop an anomaly detection scheme based on multi-scale analysis of the long range dependent traffic to address this challenge. In this proposed detection scheme, the discrete wavelet transform is used to approximately de-correlate the traffic data and capture data characteristics in different time scales. The remaining dependencies are then captured by a multi-level hidden Markov model in the wavelet domain. To estimate the model parameters, we develop an online discounting Expectation Maximization (EM) algorithm, which also tracks variations of the estimated models over time. Network anomalies are detected as abrupt changes in the tracked model variation scores. Statistical properties of our detection scheme are evaluated numerically using long range dependent time series. We also evaluate our detection scheme in malicious scenarios simulated using the NS-2 network simulator.

Keywords: Anomaly detection; Hidden Markov Model; Long range dependency; Wavelet decomposition.

Subject Classifications: 62L12; 62F03; 62F15.

1. INTRODUCTION

A wireless sensor network consists of a set of spatially scattered sensors that can measure various properties of the environment, formulate local and distributed inferences, and make responses to events or queries. It can be deployed to monitor and protect critical infrastructure assets, such as power grids, automated railroad control, water and gas distribution, etc. However, due to the often unattended operating environment, it could be easy for attackers to compromise sensors and conduct malicious behaviors. Anomaly detection is thus critical to ensure the effective functioning of sensor networks.

Anomaly detection methods can be generally classified in two categories: signature-based and statistics-based (see Dewaele et al., 2007). Signature-based detection methods use attack signatures

Address correspondence to John S. Baras, Department of Electrical and Computer Engineering and the Institute of Systems Research, University of Maryland, College Park, MD, 20742, USA; Tel: +1 (301)-405-6606; Fax: +1 (301)-314-8486; E-mail: baras@umd.edu

to identify anomalies. The attack signatures are collected based on historical observations under the same attack, thus it can not be applied to detect unknown anomalies. Statistics-based detection methods overcome this drawback by only modeling normal network traffic and treat everything that falls outside the normal scope as anomalies. A typical statistics-based anomaly detection usually consists of the following steps: first, collect network measurements and model the normal traffic as a reference, then, apply a decision rule to detect whether current network traffic deviates from the reference. In the decision rule, some statistical distance between the analyzed traffic and the reference is computed, then it is decided whether the distance is large enough to trigger an alarm. Traditional anomaly detection methods assume that the network measurements are either independent or short range dependent. However, recent studies have shown that node mobility along with spatial correlation of the monitored phenomenon in sensor networks can lead to Long Range Dependent (LRD) traffic (see Wang and Akyildiz, 2009), which can cause high false alarms if using traditional methods.

In this paper, we develop an anomaly detection scheme based on multi-scale analysis of the long range dependent traffic. Since network anomalies can take a large variety of forms, e.g., they can be caused by different MAC layer misbehaviors, various routing layer attacks and many others, these anomalies may show statistically abnormal signals in different time scales (see Zhang et al., 2008). The length of the time interval over which network measurements are collected can influence the results. Therefore, analyzing network traffic in different time scales is necessary for anomaly detection. Discrete Wavelet Transform (DWT) is a useful tool for multi-scale analysis of network traffic due to its capability of capturing data characteristics over different time scales and frequencies. Furthermore, it can approximately de-correlate autocorrelated stochastic processes. Most of the literature work on using DWT for anomaly detection would use the first order or second order statistics (mean or variance) of the wavelet coefficients for anomaly detection, e.g., they detect abrupt changes in the mean or variance of the wavelet coefficients as anomalies (see Barford et al., 2002; Zuraniewski and Rincon, 2006). In contrast, we build a probabilistic model for the wavelet coefficients through a multi-level hidden Markov model (HMM), in an effort to capture the remaining dependency among the transformed data and thus better model the network traffic and improve detection accuracy. We design a forward-backward decomposition scheme and an online discounting Expectation Maximization (EM) algorithm to estimate model parameters. The online EM algorithm can also track the structure changes of the estimated HMMs by evaluating a model variation score, which is defined as the symmetric relative entropy between the current estimated model and a previous estimated one. Network anomalies are then detected as abrupt changes in the tracked model variation scores . To evaluate the proposed anomaly detection schemes, we provide extensive simulations, including numerical experiments using two types of well defined LRD models, namely, the Fractional Gaussian Noise (FGN) and the Autoregressive Fractionally Integrated Moving Average (ARFIMA) model. We also conducted experiments using Network Simulator 2 (NS-2) for anomaly detection in wireless sensor networks.

The paper is organized as follows. Section 2 reviews the related work. Section 3 presents our wavelet-domain HMM for modeling the network traffic. The anomaly detection algorithm that is based on HMM structure changes is presented in Section 4. Numerical experiments are shown in Section 5 and simulation results using NS-2 are described in Section 6. Finally, Section 7 gives the conclusions.

2. Related Work

Network anomaly detection is an important problem and has attracted numerous research efforts. It usually involves modeling the normal traffic as a reference, and computing the statistical distance be-

tween the analyzed traffic and the reference.

The modeling of the normal traffic can be based on various statistical characteristics of the data. For example, Lakhina et al. (2005) proposed to use principal component analysis to identify an orthogonal basis along which the network measurements exhibit the highest variance. The principal components with high variance model the normal behavior of a network, whereas the remaining components of small variance are used to identify and classify anomalies. Scherrer et al. (2007) used a non-Gaussian long-range dependent process to model network traffic, which can provide several statistics such as the marginal distribution and the covariance to characterize the traffic. Zhang et al. (2009) proposed a spatio-temporal model using traffic matrices that specify the traffic volumes between origin and destination pairs in a network. Anomalies are detected by finding significant differences from historical observations. Besides these methods, the wavelet transform is another popular technique used for capturing network traffic, especially for the LRD traffic. Abry and Vitch (1998) proposed a wavelet-based tool for analyzing LRD time series and a related semi-parametric estimator for estimating LRD parameters. Barford et al. (2002) assume that the low frequency band signal of a wavelet transform represents the normal traffic pattern. They then normalize both medium and high frequency band signals to compute a weighted sum of the two signals. An alarm is raised if the variance of the combined signal exceeds a pre-selected threshold. Kim et al. (2004) used wavelet-based technique for de-noising and separating queueing delay caused by network congestions from various other delay variations. Zuraniewski and Rincon (2006) have combined wavelet transforms and change-point detection algorithms to detect the instants that the fractality changes noticeably. The key feature of these wavelet-based methods lies in the fact that the wavelet transform can turn the LRD that exists among the data samples into a short memory structure among the wavelet coefficients (see Abry et al., 2010). In our work, we build a wavelet-domain multi-level hidden Markov model for the LRD network traffic. The merit of our method is the model's mathematical tractability and its capability of capturing data dependency.

To measure the deviation of the analyzed traffic from the reference model, several statistical distances can be used, including simple threshold, mean quadratic distances, and entropy. Entropy is a measure of the uncertainty of a probability distribution. It can be used to compare certain qualitative differences of probability distributions. Gu et al. (2005) used a maximum entropy technique to estimate the reference traffic model and compute a distance measure related to the relative entropy of the analyzed network traffic with respect to the reference. Nychis et al. (2008) thoroughly evaluated entropy-based metrics for anomaly detection. In our detection scheme, we apply the symmetric relative entropy as a distance measure. The online EM algorithm can efficiently compute the symmetric relative entropy between the current HMM model and a previous estimated one. Anomalies are then detected as abrupt changes in the symmetric relative entropy measurements.

3. Wavelet domain hidden Markov model for long range dependent traffic

Wavelet transforms have been popular for analyzing autocorrelated time series due to their capability to compress multi-scale features and approximately de-correlate the time series. They can provide compact information about a signal at different locations in time and frequency. Our traffic model is in the wavelet domain. We build a Hidden Markov Model (HMM) for the wavelet transformed network measurements. The basic idea for transform domain model is that a linear invertible transform can often restructure a signal, generating transform coefficients whose structure is simpler to model.

3.1. Wavelet domain hidden Markov model

In wavelet transform (decomposition), the measurements $x(t), t = 1, \ldots, N$ are decomposed into multiple scales by a weighted sum of a certain orthonormal basis functions,

$$x(t) = \sum_{k=1}^{N} a_{L,k}\phi_{L,k}(t) + \sum_{m=1}^{L}\sum_{k} d_{m,k}\psi_{m,k}(t),$$

where $\phi_{L,k},\ \psi_{m,k}$ are the orthonormal basis, $a_{L,k},\ d_{m,k}$ are the approximation and detail coefficients. The approximation coefficients $a_{L,k}$ provide the general shape of the signal, while the detail coefficients $d_{m,k}$ from different scales provide different levels of details for the signal content, with $d_{1,k}$ providing the finest details and $d_{L,k}$ providing the coarsest details. The locality and multi-resolution properties enable the wavelet transform to efficiently match a wide range of signal characteristics from high-frequency transients and edges to slowly varying harmonics.

In our work, we apply the Discrete Wavelet Transform (DWT) to the network traffic. A DWT is a wavelet transform for which the basis functions are discretely sampled. DWT can be explained using a pair of quadrature mirror filters. Efficient methods have been developed for decomposing a signal using a family of wavelet basis functions based on convolution with the corresponding quadrature mirror filters. However, the wavelet transform cannot completely decorrelate real-world signals, i.e., a residual dependency always remains among the wavelet coefficients. A key factor for a successful wavelet-based algorithm is an accurate joint probability model for the wavelet coefficients (see Crouse et al., 1998). A complete model for the joint probability density function would be too complicated, if not impossible, to obtain in practice, while modeling the wavelet coefficients as independent is simple but disregards the inter-coefficient dependencies. To make a balance between the two extremes, we use a hidden Markov model to capture the remaining dependency among the wavelet coefficients. It is based on two properties of the wavelet transform (see Crouse et al., 1998; Mallat and Zhong, 1992): first is the *Clustering* property, meaning that if a particular wavelet coefficient is large/small, the adjacent coefficients are very likely to also be large/small; second is the *Persistence* property, meaning that large/small values of wavelet coefficients tend to propagate across scales.

For the wavelet coefficients $d_{j,k},\ j = 1, \ldots, L$ and $k = 1, \ldots, n_j$, where L is the decomposition level and n_j is the number of wavelet coefficients in scale j, we assume that each $d_{j,k}$ is associated with a hidden state $s_{j,k}$. We then use a hidden Markov model to characterize the wavelet coefficients through the factorization

$$\begin{aligned} &P(\{d_{1,i}, s_{1,i}\}_{i=1}^{n_1}, \ldots, \{d_{L,i}, s_{L,i}\}_{i=1}^{n_L}) \\ &= p(s_{L,1}) \prod_{j=2}^{n_L} p(s_{L,j}|s_{L,j-1}) \prod_{i=1}^{L-1} p(s_{i,1}|s_{i+1,1}) \prod_{i=1}^{L-1}\prod_{j=2}^{n_i} p(s_{i,j}|s_{i,j-1}, s_{i+1,\lceil j/2\rceil}) \prod_{i=1}^{L}\prod_{j=1}^{n_i} p(d_{i,j}|s_{i,j}). \quad (3.1) \end{aligned}$$

This factorization involves three main conditional independence assumptions: first, conditioned on the states at the previous coarser scale $i + 1$, the states at scale i form a first order Markov chain; second, conditioned on the corresponding state at the previous coarser scale $i+1$, i.e., $s_{i+1,\lceil j/2\rceil}$, and the previous state at the same scale, i.e., $s_{i,j-1}$, the state $s_{i,j}$ is independent of all states in the coarser scales; third, the wavelet coefficients are independent of everything else given their hidden states. The three independence assumptions are critical for deriving the inference algorithms for this wavelet domain HMM. Fig. 1 illustrate a hidden Markov model for a 3-level wavelet decomposition.

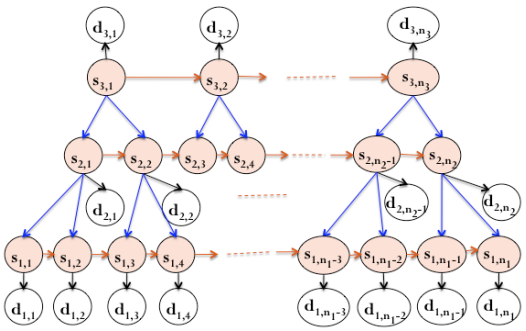

Figure 1: HMM for 3-level wavelet decomposition

3.2. Estimating model parameters using an Expectation-Maximization (EM) algorithm

Denote the set of wavelet coefficients and their hidden states by $\mathcal{D} = \{\{d_{L,i}\}_{i=1}^{n_L}, \ldots, \{d_{1,i}\}_{i=1}^{n_1}\}$ and $\mathcal{S} = \{\{s_{L,i}\}_{i=1}^{n_L}, \ldots, \{s_{1,i}\}_{i=1}^{n_1}\}$ respectively, where n_i is the number of wavelet coefficients in the i^{th} scale. The parameters of the HMM include the following three probability distributions: the first is the initial probability for the state $s_{L,1}$, i.e., $\pi_k = P(s_{L,1} = k),\ k \in \mathcal{K}$, where $\mathcal{K}$ represents the domain of the hidden states; the second is the two types of state transition probabilities, i.e.,

$$\pi^{i,1}_{k_1|k_2} = P(s_{i,1} = k_1 | s_{i+1,1} = k_2) \text{ for } i < L,$$
$$\pi^{i}_{k_1|k_2,k_3} = P(s_{i,j} = k_1 | s_{i,j-1} = k_2, s_{i+1,\lceil j/2 \rceil} = k_3);$$

and the third is the conditional probability of the wavelet coefficient given its hidden state, i.e., $P(d_{i,j}|s_{i,j} = k)$, which can be modeled by a mixture Gaussian distribution. For simplicity and presentation clarity, we use a single Gaussian distribution to capture $P(d_{i,j}|s_{i,j} = k)$, i.e, $P(d_{i,j}|s_{i,j} = k) \sim \mathcal{N}(\mu^i_k, \sigma^i_k)$, where μ^i_k and σ^i_k are the mean and the standard deviation for the state k in the i^{th} scale. The extension to mixture Gaussian distributions is straightforward. The model parameters, denoted by $\boldsymbol{\theta} = \{\pi_k, \pi^{i,1}_{k_1|k_2}, \pi^{i}_{k_1|k_2,k_3}, \mu^i_k, \sigma^i_k\}$, can be estimated from the real data using the maximum likelihood criterion. Due to the intractability of direct maximization of the likelihood function, we apply an Expectation Maximization (EM) algorithm. The EM algorithm provides an maximum likelihood estimation of model parameters by iteratively applying an E-step and an M-step. In the E-step, the expected value of the log likelihood function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = E_{\mathcal{S}|\mathcal{D},\boldsymbol{\theta}^{(t)}}[\log P_{\boldsymbol{\theta}}(\mathcal{S}, \mathcal{D})]$ is computed. Then in the M-step, the parameters that maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ are computed, i.e., $\boldsymbol{\theta}^{(t+1)} = arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.

To implement the two steps, we define the following posterior probabilities,

$$\gamma^{i,j}_k = P(s_{i,j} = k|\mathcal{D}), \gamma^{i,1}_{k_1,k_2} = P(s_{i,1} = k_1, s_{i+1,1} = k_2|\mathcal{D}), \text{ for } i < L$$
$$\gamma^{i,j}_{k_1,k_2,k_3} = P(s_{i,j} = k_1, s_{i,j-1} = k_2, s_{i,\lceil j/2 \rceil} = k_3|\mathcal{D}).$$

According to equation (3.1), maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ using the Lagrange multiplier method leads to the following estimation of $\boldsymbol{\theta}$,

$$\pi_k = \gamma^{L,1}_k, \pi^{i,1}_{k_1|k_2} = \frac{\gamma^{i,1}_{k_1,k_2}}{\sum_{l\in\mathcal{K}} \gamma^{i,1}_{l,k_2}},\ \pi^{i}_{k_1|k_2,k_3} = \frac{\sum_{j=2}^{n_i} \gamma^{i,j}_{k_1,k_2,k_3}}{\sum_{l\in\mathcal{K}} \sum_{j=2}^{n_i} \gamma^{i,j}_{l,k_2,k_3}},$$
$$\mu^i_k = \frac{\sum_{j=1}^{n_i} \gamma^{i,j}_k d_{i,j}}{\sum_{j=1}^{n_i} \gamma^{i,j}_k},\ (\sigma^i_k)^2 = \frac{\sum_{j=1}^{n_i} \gamma^{i,j}_k (d_{i,j} - \mu^i_k)^2}{\sum_{j=1}^{n_i} \gamma^{i,j}_k}.$$

The computation of the posterior probabilities $\gamma(\cdot)$ is more involved. Using a brute force computation by direct marginalization will take $O(N \cdot |\mathcal{K}|^N)$ operations, where $|\mathcal{K}|$ represents the cardinality of set $\mathcal{K}$ and N is the length of the input signal. However, by exploiting the sparse factorization in equation (3.1) and manipulating the distributive property of '+' and '$\times$', we are able to design a forward-backward decomposition algorithm to compute these posterior probabilities with computational complexity $O(N \cdot |\mathcal{K}|^{L+1})$, where L is the wavelet decomposition level and much smaller than N.

3.3. Forward-backward decomposition

Our algorithm extends the classical forward-backward decomposition algorithm for a one-level hidden Markov model to our multi-level case. The key point is to only maintain L appropriate hidden states in both the forward and backward variables for computational efficiency.

3.3.1. Forward decomposition

Defining the following variables,

$$\mathcal{S}_{i,j} = \{s_{L,\lceil 2^{i-L}j\rceil}, \ldots, s_{i+1,\lceil 2^{-1}j\rceil}, s_{i,j}, s_{i-1,2(j-1)}, \ldots, s_{1,2^{i-1}(j-1)},$$

$$\mathcal{D}_{i,j} = \{d_{L,k\leq\lceil 2^{i-L}j\rceil}, \ldots, d_{i+1,k\leq\lceil 2^{-1}j\rceil}, d_{i,k\leq j}, d_{i-1,k\leq 2(j-1)}, \ldots, d_{1,k\leq 2^{i-1}(j-1)},$$

we let the forward variable be $\alpha_{i,j} = P(\mathcal{S}_{i,j}, \mathcal{D}_{i,j})$. Let $[\alpha_{1,2^{h-1}j}] = f(h, \alpha_{h,j}), \forall h, j \in Z^+$ to define a dynamic programming algorithm. The pseudo-code for computing the forward variables using dynamic programming is shown in Table 1. Its correctness can be proved using the three conditional independence assumptions in our HMM. For simplicity and presentation clarity, in Table 1 we assume that the input data length N is a power of 2, and denote the conditional probability $P(d_{i,j}|s_{i,j})$ by $g_1(d_{i,j})$, and $P(s_{i,j}|s_{i,j-1}, s_{i+1,\lceil j/2\rceil})$ by $g_2(s_{i,j})$.

There is some implementation issue for the algorithm in Table 1, namely, the numerical under- or over-flow of $\alpha_{i,j}$ as $P(\mathcal{S}_{i,j}, \mathcal{D}_{i,j})$ becomes smaller and smaller with the increasing size of the observations $\mathcal{D}_{i,j}$. Therefore, it is necessary to scale the forward variables by positive real numbers to keep the numeric values within reasonable bounds. One solution is to use a scaled version $\bar{\alpha}_{i,j} = \frac{\alpha_{i,j}}{c_{i,j}}$, where $c_{i,j} = \sum_{\mathcal{S}_{i,j}} \alpha_{i,j}$. In this way, $\bar{\alpha}_{i,j}$ represents the probability $P(\mathcal{S}_{i,j}|\mathcal{D}_{i,j})$ and $c_{i,j}$ represents the probability $P(d_{i,j}|\mathcal{D}_{i,j}\backslash d_{i,j})$. It is straightforward to prove that both $c_{i,j}$ and $\bar{\alpha}_{i,j}$ do not depend on the number of observations. The algorithm for computing $(\bar{\alpha}_{i,j}, c_{i,j})$ can be obtained by adding a normalization step after each update of $\alpha_{i,j}$ for the algorithm in Table 1. A by-product of the *scaled* forward decomposition is that the log-likelihood $\log P(\mathcal{D})$ can be computed as

$$\log P(\mathcal{D}) = \sum_{i=1}^{L} \sum_{j=1}^{n_i} \log c_{i,j}.$$

3.3.2. Backward decomposition

Letting $\mathcal{D}_{i,j}^c = \mathcal{D} - \mathcal{D}_{i,j}$, we define the backward variable to be $\beta_{i,j} = P(\mathcal{D}_{i,j}^c|\mathcal{S}_{i,j})$. It can be computed using a similar dynamic programming algorithm as the one in Table 1. To avoid the numerical under- or over-flow problem, instead of computing $\beta_{i,j}$, we compute a scaled version $\bar{\beta}_{i,j}$ as is shown in

Table 1: Algorithm for computing forward variables

Initialization: $\alpha_{L,1} = P(s_{L,1}, d_{L,1})$
For $k_L = 1$ to $2^{-L}N$

$$\alpha_{1,2^{L-1}k_L} = f(L, \alpha_{L,k_L}), \quad \alpha_{L,k_L+1} = g_1(d_{L,k_L+1}) \sum_{s_{L,k_L}} [g_2(s_{L,k_L+1}) \cdot \alpha_{1,2^{L-1}k_L}]$$

end

function $[\alpha_{1,2^{h-1}j}] = f(h, \alpha_{h,j})$
If $h == 2$,

$$\alpha_{1,2j-1} = g_1(d_{1,2j-1}) \sum_{s_{1,2j-2}} [g_2(s_{1,2j-1}) \cdot \alpha_{2,j}], \ \alpha_{1,2j} = g_1(d_{1,2j}|s_{1,2j}) \sum_{s_{1,2j-1}} [g_2(s_{1,2j}) \cdot \alpha_{1,2j-1}]$$

else

$$\alpha_{h-1,2j-1} = g_1(d_{h-1,2j-1}) \sum_{s_{h-1,2j-2}} [g_2(s_{h-1,2j-1}) \cdot \alpha_{h,j}], \quad \alpha_{1,2^{h-2}(2j-1)} = f(h-1, \alpha_{h-1,2j-1})$$

$$\alpha_{h-1,2j} = g_1(d_{h-1,2j}) \sum_{s_{h-1,2j-1}} [g_2(s_{h-1,2j}) \cdot \alpha_{1,2^{h-2}(2j-1)}], \quad \alpha_{1,2^{h-2}(2j)} = f(h-1, \alpha_{h-1,2j})$$

End

Table 2. The scaled backward variable $\bar{\beta}_{i,j}$ represents the probability $\frac{P(\mathcal{D}^c_{i,j}|\mathcal{S}_{i,j})}{P(\mathcal{D}^c_{i,j}|\mathcal{D}_{i,j})}$. The correctness of the algorithm in Table 2 can be verified using the three conditional independence assumptions in our HMM.

3.3.3. Computing posterior probabilities

Since $\bar{\alpha}_{i,j} = P(\mathcal{S}_{i,j}|\mathcal{D}_{i,j})$ and $\bar{\beta}_{i,j} = \frac{P(\mathcal{D}^c_{i,j}|\mathcal{S}_{i,j})}{P(\mathcal{D}^c_{i,j}|\mathcal{D}_{i,j})}$, we have $\bar{\alpha}_{i,j} \cdot \bar{\beta}_{i,j} = P(\mathcal{S}_{i,j}|\mathcal{D})$ according to the Markovian property of our HMM. Then the posterior probability $\gamma(\cdot)$ can be computed as

$$\gamma^{i,j}_k = \sum \bar{\alpha}_{i,j} \cdot \bar{\beta}_{i,j}, \ \ \gamma^{i,1}_{k_1,k_2} = \sum \bar{\alpha}_{i,1} \cdot \bar{\beta}_{i,1}, \ \ \gamma^{L,j}_{k_1,k_2} = \sum \bar{\alpha}_{1,2^{L-1}(j-1)} \cdot \bar{\beta}_{L,j} \cdot \frac{g_1(d_{L,j}) g_2(s_{L,j})}{c_{L,j}},$$

$$\gamma^{i,j}_{k_1,k_2,k_3} = \begin{cases} \sum \bar{\alpha}_{1,2^{i-1}(j-1)} \bar{\beta}_{i,j} \cdot \frac{g_1(d_{i,j}) \cdot g_2(s_{i,j})}{c_{i,j}}, \text{if j is even,} \\ \sum \bar{\alpha}_{i+1,\lceil j/2 \rceil} \bar{\beta}_{i,j} \cdot \frac{g_1(d_{i,j}) \cdot g_2(s_{i,j})}{c_{i,j}}, \text{if j is odd.} \end{cases}$$

Without confusion, we omit the indexing variables under the $\sum$ symbol for the above equations.

4. Anomaly detection by tracking HMM model variations

A first thought on the anomaly detection problem is to treat the anomalies as abrupt changes in the HMM modeled data and then apply change-point detection methods to detect these abrupt changes as anomalies. This is also the general routine used in literature (see Dewaele et al., 2007). However, it is found that directly applying change-point detection methods to the HMM modeled data is computationally expensive. We designe here a lightweight anomaly detection scheme based on detecting the structure changes of the estimated HMM.

Table 2: Algorithm for computing the scaled backward variables

Initialization: $\bar{\beta}_{1,N/2} = 1$

For $k_L = 2^{-L}N$ to 1

$$\bar{\beta}_{L,k_L} = f(L, \bar{\beta}_{1,2^{L-1}k_L}), \quad \bar{\beta}_{1,2^{L-1}(k_L-1)} = \sum_{s_{L,k_L}} \frac{g_1(d_{L,k_L})g_2(s_{L,k_L})\bar{\beta}_{L,k_L}}{c_{L,k_L}}$$

end

function $[\bar{\beta}_{h,j}] = f(h, \bar{\beta}_{1,2^{h-1}j})$

If $h == 2$,

$$\bar{\beta}_{1,2j-1} = \sum_{s_{1,2j}} \frac{g_1(d_{1,2j})g_2(s_{1,2j}) \cdot \bar{\beta}_{1,2j}}{c_{1,2j}}, \quad \bar{\beta}_{2,j} = \sum_{s_{1,2j-1}} \frac{g_1(d_{1,2j-1})g_2(s_{1,2j-1}) \cdot \bar{\beta}_{1,2j-1}}{c_{1,2j-1}}$$

else

$$\bar{\beta}_{h-1,2j} = f(h-1, \bar{\beta}_{1,2^{h-2}2j}), \quad \bar{\beta}_{1,2^{h-2}(2j-1)} = \sum_{s_{h-1,2j}} \frac{g_1(d_{h-1,2j})g_2(s_{h-1,2j}) \cdot \bar{\beta}_{h-1,2j}}{c_{h-1,2j}}$$

$$\bar{\beta}_{h-1,2j-1} = f(h-1, \bar{\beta}_{1,2^{h-2}(2j-1)}), \quad \bar{\beta}_{h,j} = \sum_{s_{h-1,2j-1}} \frac{g_1(d_{h-1,2j-1})g_2(s_{h-1,2j-1}) \cdot \bar{\beta}_{h-1,2j-1}}{c_{h-1,2j-1}}$$

End

4.1. Difficulty of applying change-point detection methods on HMM modeled data

An anomaly detection problem can be formulated as a hypotheses testing problem, i.e., given finite samples $\mathcal{Y}_{1:N} = \{y_1, y_2, \ldots, y_N\}$, testing between two hypotheses,

$$H_0 : \text{for } 1 \le k \le N, p_{\boldsymbol{\theta}}(y_k|\mathcal{Y}_{1:k-1}) = p_{\boldsymbol{\theta}_0}(y_k|\mathcal{Y}_{1:k-1}),$$

$$H_1 : \exists \text{ unknown } 1 \le t_0 \le N, \; s.t. \begin{cases} \text{for } 1 \le k \le t_0 - 1, & p_{\boldsymbol{\theta}}(y_k|\mathcal{Y}_{1:k-1}) = p_{\boldsymbol{\theta}_0}(y_k|\mathcal{Y}_{1:k-1}), \\ \text{for } t_0 \le k \le N, & p_{\boldsymbol{\theta}}(y_k|\mathcal{Y}_{1:k-1}) = p_{\boldsymbol{\theta}_1}(y_k|\mathcal{Y}_{1:k-1}), \end{cases}$$

where $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ represent the model parameters for the normal network traffic and the abnormal network traffic respectively. Since $\boldsymbol{\theta}_1$ usually can not be known in advance, the hypothesis H_1 is composite (i.e., $\boldsymbol{\theta}_1 \in \{\boldsymbol{\theta} : \boldsymbol{\theta} \neq \boldsymbol{\theta}_0\}$). The generalized likelihood ratio test (GLR) (see Chen and Gupta, 2000) is one of the most popular change-point detection methods for solving this type of hypothesis testing problem. The GLR test can be written as

$$g_k = \max_{1 \le j \le k} \sup_{\boldsymbol{\theta}_1} S_j^k, \quad S_j^k = \sum_{i=j}^{k} \ln \frac{p_{\boldsymbol{\theta}_1}(y_i|\mathcal{Y}_{1:i-1})}{p_{\boldsymbol{\theta}_0}(y_i|\mathcal{Y}_{1:i-1})}, \quad t_0 = \min\{k : g_k \ge h\}.$$

It is known that the likelihood of an HMM belongs to the so called *locally asymptotic normal* family (see Cappe and Moulines, 2005), and the GLR statistic $\sup_{\boldsymbol{\theta}_1} S_j^k$ can be approximated by the second-order expansion of S_j^k at $\boldsymbol{\theta}_0$ without the computation of $\sup_{\boldsymbol{\theta}_1}$ over all possible $\boldsymbol{\theta}_1$'s. However, the computation of this second-order expansion involves computation of the Fisher information matrix of $\ln p_{\boldsymbol{\theta}_0}(y_i|\mathcal{Y}_{1:i-1})$, which, in our case, would require an update of an $L|\mathcal{K}|^3 \times L|\mathcal{K}|^3$ matrix each time when a new data sample arrives. This is not computationally realistic for our application, especially in the resource constrained wireless sensor networks.

In the next subsections, we design a lightweight algorithm for anomaly detection by detecting structure changes of the estimated HMM. An important requirement for anomaly detection is to make the decision making process online. Therefore, we first develop an online EM algorithm for HMM model estimation.

4.2. An online discounting EM algorithm

The online discounting EM algorithm is derived based on the so called *limiting EM algorithm* (see Cappe, 2009). We first briefly present the limiting EM algorithm. Let $\mathbf{x}$ denote the hidden states and $\mathbf{y}$ denote the observations. If the joint probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})$ belongs to an exponential family such that $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) = h(\mathbf{x},\mathbf{y})\exp(\langle\phi(\boldsymbol{\theta}), ss(\mathbf{x},\mathbf{y})\rangle - A(\boldsymbol{\theta}))$ where $\langle\cdot\rangle$ denotes the scalar product, $ss(\mathbf{x},\mathbf{y})$ is the sufficient statistics for $\boldsymbol{\theta}$ and $A(\boldsymbol{\theta})$ is some log-partition function. If the equation $\langle\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}), ss\rangle - \nabla_{\boldsymbol{\theta}}A(\boldsymbol{\theta}) = 0$ has a unique solution, denoted by $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}(ss)$, then the limiting EM algorithm obeys the simple recursion $ss^{t+1} = E_{\boldsymbol{\theta}^*}[E_{\bar{\boldsymbol{\theta}}(ss^t)}[ss(\mathbf{x},\mathbf{y})|\mathbf{y}]]$, where $\boldsymbol{\theta}^*$ represents the true model parameter . Since $E_{\boldsymbol{\theta}^*}[E_{\boldsymbol{\theta}}[ss(\mathbf{x},\mathbf{y})|\mathbf{y}]]$ can be estimated consistently from the observations by $\frac{1}{N}\sum_{t=1}^{N} E_{\boldsymbol{\theta}}[ss(x_t,y_t)|y_t]$, an online EM algorithm can be obtained by using the conventional stochastic approximation procedure (see Cappe, 2009),

$$\hat{ss}^{t+1} = \gamma_{t+1}E_{\bar{\boldsymbol{\theta}}(\hat{ss}^t)}[ss(x_{t+1},y_{t+1})|y_{t+1}] + (1-\gamma_{t+1})\hat{ss}^t,$$

where γ_{t+1} is a time discounting factor. The estimation of model parameters can then be derived from the sufficient statistics $\hat{ss}$. It is proved by Cappe (2009) that under suitable assumptions, this online EM algorithm is an asymptotically efficient estimator of the model parameter $\boldsymbol{\theta}^*$.

It is not difficult to see that the joint probability distribution of our HMM model, i.e., $P(\mathcal{S},\mathcal{D})$, satisfies the above mentioned conditions. For each wavelet coefficient $d_{i,j}$, we have the following sufficient statistics for computing the HMM model parameters,

$\tau_{l,k}^{i,j} = \sum_{m=1}^{n_l^{i,j}} P(s_{l,m}=k,\mathcal{S}_{i,j}|\mathcal{D}_{i,j})$, $\hat{\tau}_{l,k}^{i,j} = \sum_{m=1}^{n_l^{i,j}} P(s_{l,m}=k,\mathcal{S}_{i,j}|\mathcal{D}_{i,j})\cdot d_{l,m}$,

$\bar{\tau}_{l,k}^{i,j} = \sum_{m=1}^{n_l^{i,j}} P(s_{l,m}=k,\mathcal{S}_{i,j}|\mathcal{D}_{i,j})\cdot d_{l,m}^2$,

$\tau_{l,k_1,k_2,k_3}^{i,j} = \sum_{m=2}^{n_l^{i,j}} P(s_{l,m}=k_1, s_{l,m-1}=k_2, s_{l+1,\lceil m/2\rceil}=k_3,\mathcal{S}_{i,j}|\mathcal{D}_{i,j})$,

where $l\in\{1,\ldots,L\}$ is the scale index, $k\in\mathcal{K}$ is the hidden state index, and $n_l^{i,j}$ represents the number of observed wavelet coefficients in scale l after $d_{i,j}$ arrives, i.e.,

$$n_l^{i,j} = \begin{cases} \lceil 2^{i-l}j\rceil & \text{if } l\geq i,\\ 2^{l-i}j & \text{if } l<i.\end{cases}$$

It is straightforward to prove that the HMM model parameters $\{\pi_{k_1|k_2,k_3}^l, \mu_k^l, \sigma_k^l\}$ can be updated using $\{\tau_{l,k}^{i,j}, \hat{\tau}_{l,k}^{i,j}, \bar{\tau}_{l,k}^{i,j}\tau_{l,k_1,k_2,k_3}^{i,j}\}$ as follows,

$$\pi_{k_1|k_2,k_3}^l = \frac{\sum_{\mathcal{S}_{i,j}}\tau_{l,k_1,k_2,k_3}^{i,j}}{\sum_{k_1}\sum_{\mathcal{S}_{i,j}}\tau_{l,k_1,k_2,k_3}^{i,j}},\ \mu_k^l = \frac{\sum_k\sum_{\mathcal{S}_{i,j}}\hat{\tau}_{l,k}^{i,j}}{\sum_k\sum_{\mathcal{S}_{i,j}}\tau_{l,k}^{i,j}},\ (\sigma_k^l)^2 = \frac{\sum_k\sum_{\mathcal{S}_{i,j}}\bar{\tau}_{l,k}^{i,j}}{\sum_k\sum_{\mathcal{S}_{i,j}}\tau_{l,k}^{i,j}} - (\frac{\sum_k\sum_{\mathcal{S}_{i,j}}\hat{\tau}_{l,k}^{i,j}}{\sum_k\sum_{\mathcal{S}_{i,j}}\tau_{l,k}^{i,j}})^2. \quad (4.1)$$

The other HMM parameters π^k and $\pi_{k_1|k_2}^{i,1}$ can be updated using sufficient statistics defined as $\tilde{\tau}_{L,k}^{i,j} = P(s_{L,1}=k,\mathcal{S}_{i,j}|\mathcal{D}_{i,j})$ and $\tilde{\tau}_{l,1}^{i,j} = P(s_{l,1}=k_1, s_{l+1,1}=k_2,\mathcal{S}_{i,j}|\mathcal{D}_{i,j})$. We omit the related computations here, as they are similar.

The next step on the design of our online EM algorithm is to obtain recursive (online) updates of the sufficient statistics $\tau_{l,k}^{i,j}$, $\hat{\tau}_{l,k}^{i,j}$, $\bar{\tau}_{l,k}^{i,j}$ and $\tau_{l,k_1,k_2,k_3}^{i,j}$. According to the Markovian property of the HMM, the online updates of the sufficient statistics can be achieved by following a similar dynamic programming procedure as the one for computing the scaled forward variables $\bar{\alpha}_{i,j}$. Recall that $\bar{\alpha}_{i,j}$ is computed by adding a normalization step after $\alpha_{i,j}$ is computed in the algorithm in Table 1. The sufficient statistics can be updated once $\bar{\alpha}_{i,j}$ is updated. For illustration purposes, we show here how to update the sufficient statistics when $\alpha_{h,2j-1}$ in Table 1 is computed, as updates for the other cases are similar. For $l \in \{1, \ldots, L\}$, let $\gamma_{h-1,2j-1}$ be a time discounting factor, and δ_{h-1}^{l} be the Dirac Delta function such that $\delta_{h-1}^{l} = 1$, if $l = h-1$ and $\delta_{h-1}^{l} = 0$ if $l \neq h-1$, and let

$$r_{h-1,2j-1}^{l} = \delta_{h-1}^{l} \cdot \gamma_{h-1,2j-1} \cdot \bar{\alpha}_{h-1,2j-1}, \quad t_{h-1,2j-1}^{l} = (1 - \delta_{h-1}^{l}\gamma_{h-1,2j-1})\frac{g_1(d_{h-1,2j-1})}{c_{h,j}},$$
$$q_{h-1,2j-1}^{l} = \delta_{h-1}^{l}\gamma_{h-1,2j-1}\frac{g_1(d_{h-1,2j-1})g_2(s_{h-1,2j-1})\bar{\alpha}_{h,j}}{c_{h-1,2j-1}},$$

where $g_1(\cdot)$ and $g_2(\cdot)$ are defined as in Table 1. The sufficient statistics can be updated as follows,

$$\tau_{l,k}^{h-1,2j-1} = r_{h-1,2j-1}^{l} + t_{h-1,2j-1}^{l} \sum_{s_{h-1,2j-2}} g_2(s_{h-1,2j-1})\tau_{l,k}^{h,j}, \tag{4.2}$$

$$\hat{\tau}_{l,k}^{h-1,2j-1} = r_{h-1,2j-1}^{l} d_{h-1,2j-1} + t_{h-1,2j-1}^{l} \sum_{s_{h-1,2j-2}} g_2(s_{h-1,2j-1})\hat{\tau}_{l,k}^{h,j}, \tag{4.3}$$

$$\bar{\tau}_{l,k}^{h-1,2j-1} = r_{h-1,2j-1}^{l} d_{h-1,2j-1}^{2} + t_{h-1,2j-1}^{l} \sum_{s_{h-1,2j-2}} g_2(s_{h-1,2j-1})\bar{\tau}_{l,k}^{h,j}, \tag{4.4}$$

$$\tau_{l,k_1,k_2,k_3}^{h-1,2j-1} = q_{h-1,2j-1}^{l} + t_{h-1,2j-1}^{l} \sum_{s_{h-1,2j-2}} g_2(s_{h-1,2j-1}) \cdot \tau_{l,k_1,k_2,k_3}^{h,j}. \tag{4.5}$$

In summary, the online discounting EM algorithm works as follows. Each time a new wavelet coefficient arrives, the sufficient statistics are updated accordingly, e.g., when $d_{h,2j-1}$ arrives, updating the sufficient statistics using equation (4.2, 4.3, 4.4, 4.5). After a minimum number n_{min} of wavelet coefficients are observed, where n_{min} is small, i.e., $n_{min} = 20$ might be enough, the HMM model parameters are updated according to equation (4.1).

4.3. Change-point detection on model variations

To measure the structure changes of the estimated HMM models over time, we use the concept of the symmetric relative entropy to define a model variation score (see Hirose and Yamanishi, 2008). Denote the model at time $t-1$ and t by P_{t-1} and P_t respectively, then the model variation score is defined as

$$v_t = \lim_{n\to\infty} \frac{1}{n} D(P_t||P_{t-1}) + \lim_{n\to\infty} \frac{1}{n} D(P_{t-1}||P_t),$$

where $D(p||q)$ represents the relative entropy of distribution p to q, and n represents the length of the input data. It is natural to let $n \to \infty$ as we can then compare the two models under the stationary

states in the limit of $n \to \infty$. It can be proved that

$$\lim_{n\to\infty} \frac{1}{n} D(P_t||P_{t-1}) = \sum_{i=1}^{L} \frac{1}{2^i} D((\pi^i_{k_1|k_2,k_3})^t||(\pi^i_{k_1|k_2,k_3})^{t-1})$$
$$+ \sum_{i=1}^{L} \frac{1}{2^i} \sum_k (\pi^i_k)^t D(\mathcal{N}((\mu^i_k)^t, (\sigma^i_k)^t)||\mathcal{N}((\mu^i_k)^{t-1}, (\sigma^i_k)^{t-1})),$$

where $\pi^i_k = P(s_{i,j} = k)$. Therefore, besides the probability distributions $\pi^i_{k_1|k_2,k_3}$ and $\mathcal{N}(\mu^i_k, \sigma^i_k)$ provided by the online EM algorithm, the computation of $\lim_{n\to\infty} \frac{1}{n} D(P_t||P_{t-1})$ also involves the probability distributions $\pi^i_{k_1,k_2,k_3} = P(s_{i,j} = k_1, s_{i,j-1} = k_2, s_{i+1,\lceil j/2 \rceil} = k_3)$ and π^i_k. The estimation of π^i_k and $\pi^i_{k_1,k_2,k_3}$ can be obtained from the sufficient statistics $\tau^{l,m}_{i,k}$ and $\tau^{l,m}_{i,k_1,k_2,k_3}$ as $\pi^i_k = \sum_{\mathcal{S}_{l,m}} \tau^{l,m}_{i,k}$, $\pi^i_{k_1,k_2,k_3} = \sum_{\mathcal{S}_{l,m}} \tau^{l,m}_{i,k_1,k_2,k_3}$. The other relative entropy $\lim_{n\to\infty} \frac{1}{n} D(P_{t-1}||P_t)$ can be computed similarly. We can see that the symmetric model variation score actually captures two types of changes. The first is the changes in the transition probabilities of the hidden states while the second is the changes in the generation pattern of the observed data from a fixed state. By using the symmetric relative entropy as a distance measure between two HMM models, it is expected that not only the changes of the data generation pattern will be detected, but also the changes in the hidden states can also be detected.

5. Numerical Evaluations

In this section, we numerically evaluate the statistical properties of the proposed anomaly detection scheme. Two types of LRD time series, including the Fractional Gaussian Noise (FGN) and the Autoregressive Fractionally Integrated Moving Average (ARFIMA) model, are used for generating data. We then inject two types of model variations as anomalies. First is the mean level shift, i.e., a step function with a constant amplitude is imposed on the original signal. Second is to vary model parameters for the data generation process, including the standard deviation and the Hurst parameter for FGN and ARFIMA. The duration for the normal state and the anomaly state is generated from exponential distributions with different mean values. The performance of the detection scheme is evaluated by the detection latency and the Receiver Operating Characteristic (ROC) curve, which is a plot of the detection rate versus the false alarm rate at different thresholds.

The selection of the wavelet basis used in our scheme is based on a balance between its *time localization* and *frequency localization* characteristics (see Barford et al., 2002). Long filters usually have poor time localization, which can lead to excessive blurring in the time domain, thus may miss strong but short-duration changes in the time series. In contrast, short filters have good time localization but poor frequency localization, which can lead to the appearance of large wavelet coefficients when no significant event is occurring and can cause high false alarms rates if detection is based on a simple threshold. In our scheme, we build a hidden Markov model for the wavelet coefficients and the detection is based on HMM structure changes rather than a simple threshold, thus the sensitivity of the filter's frequency localization capability on detection performance is significantly reduced. In our experiments, we found that the D2 (Haar wavelets) and D4 wavelets from the Daubechies family wavelets can give us relatively good performance. Hence we use the Haar wavelets for all the experiments in this paper. For performance comparison, we implement a baseline method adapted from the method

proposed by Barford et al. (2002), in which only the mean and variance of the wavelet coefficients is used for anomaly detection. More specifically, for the wavelet coefficients at each scale, it only computes the mean and variance over a time window with fixed length. Any abrupt changes in the mean and variance values are treated as anomalies.

Fig. 2 shows one representative example of the detection performance on mean level shifts in the synthetic LRD time series. The top figure illustrates the time series, which is generated from an ARFIMA model with Hurst parameter 0.9 and length 2^{15}. The standard deviation for the generated data sequence is set to 1. The mean level shift occurs at the first quarter of the time and ends at the middle with intensity 0.75, which is less than the standard deviation. The bottom two figures show the corresponding model variation scores computed by our online EM algorithm with 5-level and 4-level wavelet decomposition respectively. The x axis represents the aggregated time due to the wavelet decomposition and the y axis represents the model variation score. From Fig. 2, we can see that the

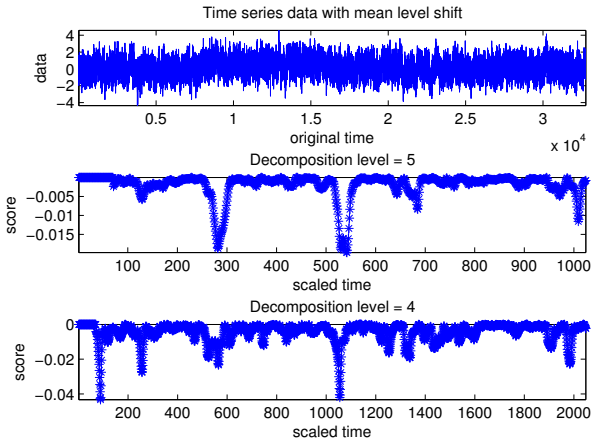

Figure 2: Effect of Wavelet decomposition levels

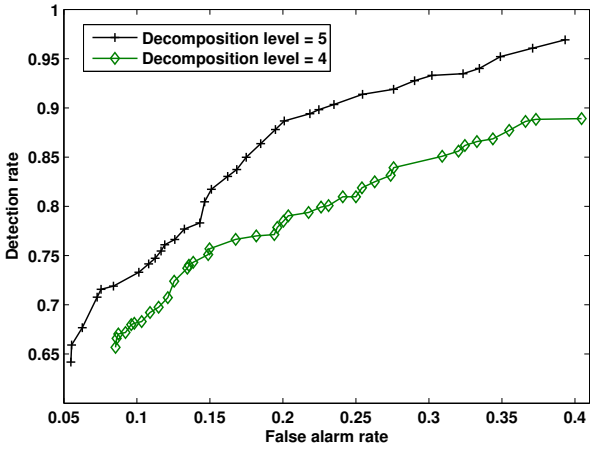

Figure 3: ROC curves: different wavelet levels

visual inspection of the injected mean level shift from the time series directly can be difficult. However, in the tracked model variation scores, there are abrupt changes of the model variation scores at the two time locations where the mean level shift starts and ends. These two abrupt changes suggest where the injection starts and ends. Especially when the wavelet decomposition level is 5, these are the only 2 abrupt changes that exist. When the wavelet decomposition level is 4, there are some false alarms. We further compare the effects of the wavelet decomposition level on detection performance. Fig 3 shows the Receiver Operating Characteristic (ROC) curves using different decomposition levels. The ROC curves are obtained over 1000 randomly generated time series with standard deviation 1. The mean level drift starts at different random time points with length 2^{13}. For a fair comparison, all of them have the same injected intensity of 0.75. We can see that a higher decomposition level can reduce the false alarms while achieving the same detection rate. However, an L-level decomposition has a 2^L time aggregation scale, i.e., it transforms the data samples within a 2^L time window to the wavelet domain, so the wavelet coefficients within this window are time-indistinguishable. Therefore, a higher level decomposition would often give longer detection latency than that of a smaller level decomposition. In our experiments, we found that a 5-level wavelet decomposition can give a reasonable good balance between detection accuracy and latency.

The intensity of the injected mean level shift also affects the detection performance. Fig. 4 shows the ROC curve and detection latency for the injected mean shift with different intensities. Each curve is obtained over 1000 simulation traces with the 5-level wavelet decomposition. As is expected, for higher injected mean level shifts, the detection becomes much easier, in terms of lower false alarms, higher detection rates, and smaller detection latency.

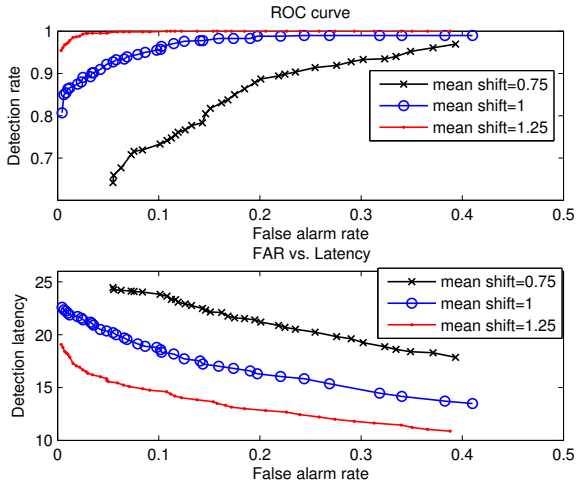

Figure 4: Effects of injected intensities

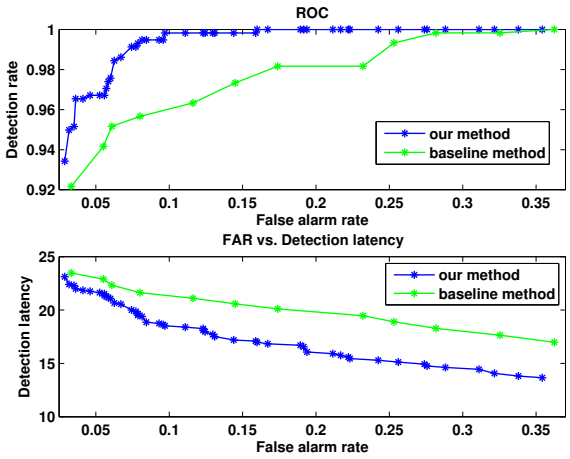

Figure 5: Comparison with the baseline

As for the detection on changes in the data generation model, we have similar observations, i.e., when the Hurst parameter or the data variance changes more, the detection becomes easier. Due to space constraints, we are not showing the corresponding results here.

We next compare the performance of our algorithm to the baseline method. It is observed that our method can always beat the baseline method. For example, Fig. 5 shows the ROC curves and detection latency for our method and the baseline methods on the detection of Hurst parameter changing from 0.9 to 0.7. While achieving the same false alarm rate, our method has a higher detection rate and smaller detection latency. Similar results are observed for the detection of other types of injected anomalies. It verifies our expectation that the hidden Markov model can capture more characteristics of the wavelet domain data than using only the first and second order statistics.

6. NS-2 Simulation Studies

We next create anomaly scenarios in wireless sensor networks using NS-2 simulator. We simulate the in-band wormhole attack in the routing layer and evaluate the proposed detection scheme. Note that our detection scheme is a general method, i.e., it is not only designed for the wormhole attacks, but can adapt to detect any other attacks that will cause deviation of the network traffic from its normal state.

The in-band wormhole attack is a collaborative attack in the routing layer. During a wormhole attack, the malicious nodes perform a tunneling procedure to form a wormhole where one node receives packets and covertly tunnels them to another colluding node, and then the colluding node replays these packets as if it receives them from its physical neighbors. The in-band wormhole connects the purported neighbors via multi-hop tunnels over the existing wireless medium. It can affect shortest path routing calculations and allow the attacking nodes to attract and route traffic from other parts of the network

to go through them, thus create artificial traffic choke points that can be utilized at an opportune future time to analyze network traffic and degrade network performance. Fig. 6 shows an example of a 3-hop in-band wormhole. By 'n-hop wormhole', we mean that the actual path length between the two wormhole endpoints is n-hop but the routing protocol is fooled to consider it as only 1-hop.

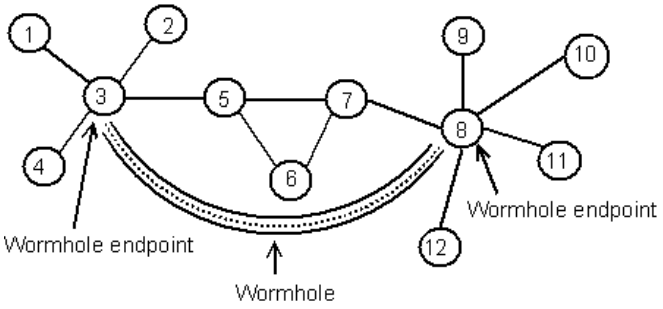

Figure 6: A 3-hop in-band wormhole

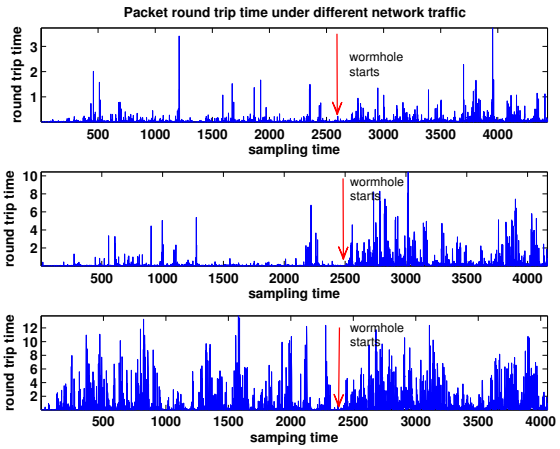

Figure 7: Three scenarios of wormhole

In the in-band wormhole attack, the multi-hop tunneling process will cause the transmission delay along a path to deviate from its normal state. For example, in Fig. 6 the attacker fools the routing protocol to consider the 3-hop path 3-5-7-8 as a 1-hop path 3-8. The transmission delay of a 3-hop path, however, would be different from a real 1-hop path due to different path lengths, not to mention that the wormhole attack can introduce additional congestion in the path due to the attraction of traffic from other parts of the network. The difficulty of detection lies in the fact that traffic variability, such as network congestion, may lead to high false alarm rates.

We create the in-band wormholes in networks containing 50 nodes in a 1000x1000 square field using NS-2. Different simulation scenarios are considered, including networks that have different levels of background traffic and wormholes that have different lengths. Fig. 7 shows three typical scenarios of the collected packet round trip times between a source-destination pair that was attracted by a 4-hop wormhole, where the wormhole starts at time around 2500. The top figure (scenario 1) corresponds to the case when the background traffic is relatively light, so there is no congestion in the wormhole or other places of the network. The packet round trip time becomes slightly higher after the wormhole attack starts. The middle one (scenario 2) shows the case when the background traffic becomes heavier. Since more traffic was attracted by the wormhole, leading to some level of congestion inside the wormhole, paths go through the wormhole have much longer round trip time than its previous normal state. This case is the easiest case for wormhole detection. In the bottom figure (scenario 3) the background traffic becomes much heavier, in which case the network congestion causes large traffic variation even when there is no wormhole attack. It is the most difficult case for wormhole detection. Fig. 8 presents the ROC curve for detection of the 4-hop wormhole under different levels of background traffic corresponding to the three scenarios. The results are obtained over 100 simulation runs. In the worst case, i.e., scenario 3, our algorithm performs much better than a random guess, e.g., when the false alarm rate reaches 0.4, the detection rate is around 0.8. The results are satisfactory considering that the end-to-end packet round trip time is the only traffic profile we used for detection. We expect that if our

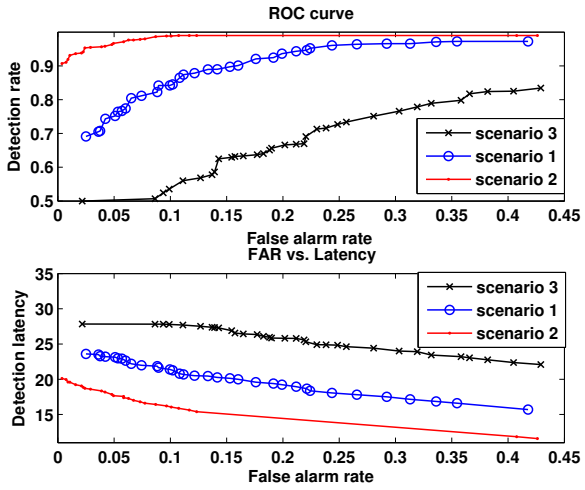

Figure 8: Detection of different wormholes

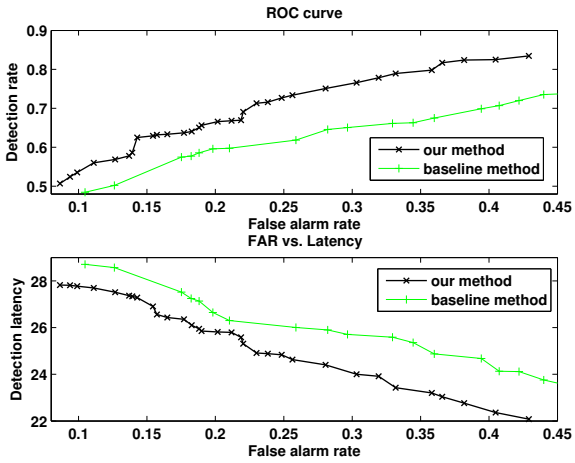

Figure 9: Comparison with the baseline

detector is combined with methods based on other characteristics of the network traffic, better detection performance can be achieved. Fig. 9 shows performance comparison of our method with the baseline method for scenario 3, in which our method achieves better performance.

7. Conclusions

In this paper, we studied the anomaly detection problem in wireless sensor networks. As discovered by recent works, the traffic in wireless sensor networks can have the similar long range dependency (LRD) property as for the wireline and wireless 802.11 networks, which could significantly increase the difficulty of network anomaly detection. To reduce the effect of LRD on anomaly detection performance, we proposed a wavelet-domain hidden Markov model for capturing the normal network traffic. The wavelet transform is able to turn the long range dependency that exists among the sample data into a short memory structure among its wavelet coefficients. The HMM in the wavelet-domain is used to further capture the remaining dependency among the wavelet coefficients, thus modeling the traffic variability more accurately. Network anomalies are then detected as abrupt changes in the tracked HMM model structures. The performance of our algorithm is evaluated by extensive simulations, including numerical experiments in Matlab and experiments using Network Simulator 2. In our future work, we plan to study the optimization of model parameters for the wavelet domain HMM model to improve performance.

ACKNOWLEDGEMENTS

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under award number 013641-001 for the Multi-Scale Systems Center (MuSyC), through the FRCP of SRC and DARPA, US Air Force Office of Scientific Research MURI award FA9550-09-1-0538, and Army Research Office MURI award W911-NF-0710287.

REFERENCES

Abry, D. and Veitch, D. (1998). Wavelet analysis of long range dependent traffic. *IEEE Transactions of Information Theory*. 44: 2–15.

Abry, P., Helgason, H. and Pipiras, V. (2010). Wavelet-based analysis of non-gaussian long-range dependent processes and estimation of the Hust parameter. *preprint submitted to Elsevier*.

Barford, P., Kline, J., Plonka, D.and Ron, A. (2002). A Signal Analysis of Network Traffic Anomalies. *ACM SIGCOMM Internet Measurement Workshop*. pp. 71–82, Marseille, France.

Cappe, O. and Moulines, E. (2005). Inference in the hidden Markov models. *Springer*. Springer.

Cappe, O. (2009). Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*.

Chen, J. and Gupta, A. (2000). Parametric statistical change point analysis. *Birkhauser Verlag*. BirkhŁuser Boston.

Cheng, C. , Kung, H. and Tan, K. (2002). Use of spectral analysis in defense against DoS attacks. *Proceedings of IEEE Globecom*. 2002 Nov. 17-21, pp. 2143-2148.

Cherrer, A. , Larrieu, N., Owezarski, P. , Borgnat, P., Abry, P. , Phys, L. and Lyon, E. (2007). Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies. *IEEE Transactions on Dependable and Secure Computing* 4:56-70.

Crouse, M., Nowak, R. and Baraniuk, R. (1998). Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on signal processing*. 46: 886–902.

Dewaele, G. ,Fukuda, K. , Borgnat, P. , Abry, P. and Cho, P. (2007). Extracting hidden anomalies using sketch and non-Gaussian multiresolution statistical detection procedures. *Proceedings of ACM SIGCOMM Workshop on Large-Scale Attack Defense*. pp. 145–152, Kyoto, Japan.

Gu, Y., McCallum, A. and Towsley, D. (2005). Detecting anomalies in network traffic using maximum entropy estimation. *Proceedings of the ACM SIGCOMM conference on Internet Measurement*. pp. 32–32, Berkeley, CA, USA.

Hirose, S. and Yamanishi,K. (2008). Latent Variable Mining with Its Applications to Anomalous Behavior Detection. *Journal of Statistical Analysis and Data Mining*. 2: 70–86.

Hussain, A., Heidemann, J. and Papadopoulos, C. (2003). A framework for classifying denial of service attacks. *Proceedings of ACM SIGCOMM*. pp. 99–110, Karlsruhe, Germany.

Jin, S. and Yeung, D. (2004). A covariance analysis model for DDoS attack detection. *Proceedings of IEEE ICC*. 2004 June 20-24, pp. 1882–1886.

Kim, M., Kim, T., Shin, Y., Lam, S. and Powers, E. J. (2004). A Wavelet-based approach to detect shared congestion. *Proceedings of ACM SIGCOMM*. pp. 293–306, Portland, Oregon, USA.

Lakhina, A., Crovella, M. and Diot, C. (2005). Mining Anomalies Using Trafc Feature Distributions. *Proceedings of ACM SIGCOMM Conference*. pp. 217–228, Philadelphia, Pennsylvania, USA.

Mallat, S. and Zhong, S.(1992). Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14: 710 -732.

Nychis, G., Sekar, V. , Andersen, D., Kim, H. and Zhang, H. (2008). An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. *Proceedings of ACM SIGCOMM conference on Internet Measurement* Oct. 20-22, Vouliagmeni, Greece.

Wang, P. and Akyildiz, F. (2009). Spatial Correlation and Mobility Aware Traffic Modeling for Wireless Sensor Networks. *Proceedings of IEEE Globecom*. pp. 3128–3133, Honolulu, Hawaii, USA.

Zhang, Y., Roughan, M. , Willinger, W. and Qiu, L. (2009). Spatio-Temporal compressive sensing and Internet traffic matrices. *Proceedings of SIGCOMM*. pp. 267–278, Barcelona, Spain.

Zhang, L., Zhu, Z. ,Jeffay, K., Marron, J. and Smith, F. (2008). Multi-resolution anomaly detection for the Internet. *Proceedings of INFOCOM Workshops*. pp. 1–6.

Zuraniewski, P. and Rincon, D. (2006). Wavelet transforms and chenge-point detection algorithms for tracking network traffic fractality. *Proceedings of 2nd Conference on Next Generation Internet Design and Engineering*. pp. 216-223.