A New Online Anomaly Learning and Detection for Large-Scale Service of Internet of Thing

JunPing Wang *
Laboratory of Precision
Sensing and Control Center,
Institute of Automation,
Chinese Academy,
Beijing, P R China
Email: wangjunping@bupt.edu.cn

Qiuming Kuang†
Laboratory of Precision
Sensing and Control Center,
Institute of Automation,
Chinese Academy,
Beijing, P R China
Email: 641298881@qq.com

ShiHui Duan †
Beijing Key Laboratory of Cloud
Computing Standard and
Verification Research Institute,
China Academy of Telecommunication
Research of MIIT,
Beijing, P R China
Email: duanshihui@ritt.cn

***Abstract*—The online anomaly detection has been propounded as the key idea of monitoring fault of large-scale sensor nodes in Internet of Things. Now the exciting progresses of research have been made in online anomaly detection area. However, the highly dynamic distributing character of Internet of Things makes the anomaly detection scheme difficult to be used in online manner. This paper presents a new online anomaly learning and detection mechanism for large-scale service of Internet of Thing. Firstly, our model uses the reversible-jump MCMC learning to online learn anomaly-free of dynamics network and service data. Next, we perform a structural analysis of IoT-based service topology by Network Utility Maximization (NUM) theory. The results of experiment demonstrate the method accuracy in forecasting dynamics network and service structures from synthetic data.**

I. Introduction

The Internet of Things is a new innovation service ecosystem for the Information and Communication Technologies (ICT). Currently, the complexity of IoT deployments are increasing and evolving quickly, with addition of new hardware components and system software. Such innovative service are deployed on a variety kinds of sensors and actuator for multiple context-aware applications, such as environment monitoring, smart cities, smart homes/building management, and health-care [1] [2]. However, the highly dynamic distributed manner of IoT makes the difficulty for the ability drawing meaningful and precise inferencing from the collected IoT data, which in turn requires having high sensor data quality. Therefore, online anomaly detection becomes major issues due to the extremely large scale of the resulting system, and the high level of dynamism in the network [2].

The online anomaly detection in [3] is required for large-scale context of IoT monitoring, being able to operate in a highly dynamic distributed manner in real time, and nodes work independently without the prior knowledge. The operation in highly dynamic distributed manner makes sense to improve anomaly detection performance in WSNs[3], and prolongs the lifetime of the networks. If sensed data coming from the sensor nodes in close vicinity are largely correlated, the anomaly detector can significantly improve detection accuracy and robustness by taking advantage of such spatial correlations. However, communication capability is often much more constrained than its computational capability in a sensor node [1]. As a result, the centralized data collection is impractical for any sensor node alone by online anomaly detection [13]. Instead, a typical solution is that anomaly detectors are locally built in member nodes within a neighborhood area, where one head node is in charge of obtaining a neighborhood-wide global normal profile through aggregating the local summary information reported by the member nodes. Then, each anomaly detector performs local detection with a global normal profile. In the meantime, the operation in a highly dynamic distributed manner requests that all sensor nodes need to participate in in-network computation, such that the life-cycle of a network can be prolonged due to an even distribution of computational overhead over the entire network. Most of the existing online anomaly detection techniques, which analyze data in an offline manner, fail to handle streaming data of sensor nodes [3]. Usually, offline detection works in an intermittent fashion. During an idle period of time, the network may have been damaged by random faults. Therefore, the anomaly detectors should function in real time, i.e., in an online manner. In addition, the normal profile may change frequently in streaming data, and thus, online update is essential for capturing the newest dynamics of data at any time. Many online anomaly detection techniques may work depending heavily on the prior knowledge, such as the underlying distribution [3] [13].

Though many interesting progresses have been made toward online anomaly detection, the research is still in an infant stage. Furthermore, these anomaly detectors techniques only work well in the applications with specific prior knowledge. This needs provide the new online anomaly detectors with maximized flexibility and adaptability. Taking into consideration existing large-scale nodes and highly dynamic distributed, this paper presents a new online anomaly learning and forecasting model for large-scale service of IoT. The main contributions made in this paper include modeling large-scale service performance variables from IoT node, the algorithm of performance weakening parameter learning, and performance weakening cycle forecasting process. The rest of this paper is outlined as follows: An anomaly parameter learning and

forecasting model is proposed in Section II. Section III details describes process of the anomaly parameter learning and forecasting for large-scale IoT nodes. Finally, effectiveness of the proposed scheme is verified by the experiments in Section IV. Section V gives the conclusions of this paper.

II. Related Work

The sensor device life-cycle forecasting is an area of active research for IoT measurements. The measurement of sensor device life-cycle can deviate from their predictable values due to an unexpected event or without any known causes, especially the S. R West in [4] presents the automated fault detection and diagnosis by statistical machine learning in HVAC systems. In addition to dynamic Bayesian Networks and HMMs, data fusion is also used to combine fault detection results from multiple fault models in an attempt to achieve a more accurate fault detection outcome. The method in [4] develops HMMs to learn probabilistic relationships between groups of points during both normal and faulty operation.

The E. U. Warriach in [5] presents a novel identifying and classifying faults in wireless sensor networks by Hidden Markov Models(HMMs). The proposed approach learns the possible system outcome dynamically without any distinct training period. Furthermore, it can be used to identify and classify data and system faults considering the structural relations between two kind of HMMs dynamically created. Despite HMMs are effectively used to anomaly detection as a method to model usual actions, there does not exist a well-accepted method for detection of node and system data faults, and their classification in wireless sensor networks. A cutting data edge challenge is to develop the capability to carry out fault diagnosis in terms of its identification and classification for data and system faults.

In online anomaly detection approach for sensor device life-cycle measurements, author in [6] solved the problem of outlier detection in IoT and provide a technique-based taxonomy framework to categorize current outlier detection techniques designed for sensor nodes. Author also introduce the key characteristics and brief description of current outlier detection techniques using the proposed taxonomy framework and provide an evaluation for each technique. Furthermore, author present a comparative table to compare these techniques in terms of the nature of sensor data, characteristics of outlier and outlier detection. The shortcomings of existing techniques for sensor nodes clearly calls for developing outlier detection technique, which takes into account multivariate data and the dependencies of attributes of the sensor node, provides reliable neighborhood, proper and flexible decision threshold, and also meets special characteristics of sensor nodes such as node mobility, network topology change and making distinction between errors and events. The novel approach in [7] utilized piecewise linear models of time series, which are succinct, representative, and robust, and therefore enabled it to (a)compute such models in near real-time, (b) create models without prior knowledge about anomaly types that sensor data might contain, and (c) compare and communicate different time series efficiently.

H. Sagha in [8] presents a method for detecting anomalies in classifier ensembles. Author found that the method is comparable with GLR and OCSVM. The advantages of the method compared to them is that it avoids monitoring raw data or features and only takes into account the decisions that are made by their classifiers, therefore it is independent of sensor modality and nature of anomaly. On the other hand, author found that OCSVM is very sensitive to the chosen parameters and furthermore in different types of anomalies it may react differently. A hypergrid KNN-based anomaly detection scheme in [3] is proposed to take advantages of the simplicity and scalability of KNN-based anomaly detection schemes. Based on the intuition of hypergrid, improvements are made over the original KNN-based anomaly detection schemes to meet the specific requirements. More importantly, a method was proposed for estimating the parameters adaptively. To the best of knowledge, the proposed scheme is the first one working well without the need to manually adjust its parameters. Meanwhile,these approaches for anomaly detection have been applied in medical area. A lightweight anomaly detection approach for medical WSNs is proposed in [9]. The proposed approach is based on wavelet decomposition, hampel filter, and boxplot, and it is able to achieve spatial and temporal analysis, without prior knowledge of fault signatures. It is suitable for online detection and isolation for faulty or injected measurements with low computational complexity and storage requirement.

It is known that device fault data of IoT exits the issues of low quality and poor reliability. Some of the more prevalent issues include hardware and software false errors and faults, interference, widely variable environment dependent noise, inconsistencies, and damaged sensors. However, the existing anomaly detection technique for understanding possible relation among a set of variables posits a shared conditional probability distribution for the variables measured on each individuals within a cluster of IoT. IoT is also often referred to as cluster networks, where individuals in IoT are represented by nodes, clusters are called modules, and the focus is on estimating the relevant structure among modules. But, estimation solely from sensor node-specific variables can lead to spurious dependencies, and unverifiable structural assumptions are often used for regularization. Here, this research an extended model that leverages direct observations about the IoT in addition to node-specific variables. By integrating complementary data types, it could avoid the need for structural assumptions.

III. System model

A. Challenges of Online Anomaly Detection for Large-scale IoT

In the virtualization-based next generation Internet, a IoT service delivery system is constructed by a large number of small, low-cost sensor nodes distributed over a large area with one or possibly more powerful sink nodes gathering readings of sensor nodes. The sensor nodes are integrated with sensing,

processing and wireless communication capabilities. Each node is usually equipped with a wireless radio transceiver, a small microcontroller, a power source and multi-type sensors such as temperature, humidity, light, heat, pressure, sound, vibration, etc. The IoT is not only used to provide fine-grained real-time data about the physical world but also to detect time-critical events. A wide variety of applications of IoT includes those relating to personal, industrial, business, and military domains, such as environmental and habitat monitoring, object and inventory tracking, health and medical monitoring, battle-field observation, industrial safety and control, to name but a few. In many of these applications, real-time data mining of sensor data to promptly make intelligent decisions is essential [10] [6].

Devices anomaly data measured and collected by IoT is often unreliable. The quality of anomaly data sets may be affected by noise and error, missing values, duplicated data, or inconsistent data. The limited resource and capability in IoT make the anomaly data generated by sensor nodes unreliable and inaccurate. Especially when battery power is exhausted, the probability of generating erroneous data will grow rapidly [11]. On the other hand, operations of sensor nodes are frequently susceptible to environmental effects. The vision of large scale and high density IoT is to randomly deploy a large number of sensor nodes (up to hundreds or even thousands of nodes) in harsh and unattended environments. It is inevitable that in such environments some sensor nodes malfunction, which may result in noisy, faulty, missing and redundant data. Furthermore, sensor nodes are vulnerable to malicious attacks such as denial of service attacks, black hole attacks and eavesdropping [12] [6].

The section considers that above internal and external factors lead to unreliability of anomaly data, which further influence quality of raw data and aggregated results. Extracting useful knowledge from raw sensor data is extremely important to ensure the reliability and accuracy of anomaly data before the decision-making process. The context of IoT and the nature of anomaly data make design of an appropriate online anomaly learning and forecasting more challenging. Therefore, the main challenging of online anomaly learning and forecasting are:

1) Resource constraints. The low cost and low quality sensor nodes have stringent constraints in resources, such as energy, memory, computational capacity and communication bandwidth. Most of traditional outlier detection techniques have paid limited attention to reasonable availability of computational resources. They are usually computationally expensive and require much memory for data analysis and storage. Thus, a challenge for outlier detection in WSNs is how to minimize the energy consumption while using a reasonable amount of memory for storage and computational tasks.

2) High communication cost. In WSNs, the majority of the energy is consumed for radio communication rather than computation. For a sensor node, the communication cost is often several orders of magnitude higher than the computation cost [6]. Most of traditional outlier detection techniques using centralized approach for data analysis cause too much energy consumption and communication overhead. Thus, a challenge for outlier detection in WSNs is how to minimize the communication overhead in order to relieve the network traffic and prolong the lifetime of the network.

3) Distributed streaming data. Distributed sensor data coming from many different streams may dynamically change. Moreover, the underlying distribution of streaming data may not be known a priori. Furthermore, direct computation of probabilities is difficult [6]. Most of traditional outlier detection techniques that analyze data in an offline manner do not meet the requirement of handling distributed stream data. The techniques based on the a priori knowledge of the data distribution also cannot be suitable for sensor data. Thus, a challenge for outlier detection in WSNs is how to process distributed streaming data online.

4) Dynamic network topology, frequent communication failures, mobility and heterogeneity of nodes. A sensor network deployed in unattended environments over extended period of time is susceptible to dynamic network topology and frequent communication failures. Moreover, sensor nodes may move among different locations at any point in time, and may have different sensing and processing capacities. Each sensor node may even be equipped with different number and types of sensors. Such dynamicity and heterogeneity increase the complexity of designing an appropriate outlier detection technique for WSNs.

5) Large-scale deployment. Deployed sensor networks can have massive size (up to hundreds or even thousands of sensor nodes). The key challenge of traditional outlier detection techniques is to maintain a high detection rate while keeping the false alarm rate low. This requires the construction of an accurate normal profile that represents the normal behavior of sensor data [6]. This is a very difficult task for large-scale sensor network applications. Also, traditional outlier detection techniques do not scale well to process large amount of distributed data streams in an online manner.

6) Identifying outlier sources. The sensor network is expected to provide the raw data sensed from the physical world and also detect events occurred in the network. However, it is difficult to identify what has caused an outlier in sensor data due to the resource constraints and dynamic nature of WSNs. Traditional outlier detection technique often do not distinguish between errors and events and regard outlier as errors, which results in loss of important hidden information about events. Thus, a challenge of outlier detection in WSNs is how to identify outlier sources and make distinction between errors, events and malicious attacks.

Thus, the main challenge faced by outlier detection techniques for WSNs is to satisfy the mining accuracy requirements while maintaining the resource consumption of WSNs to a minimum [6]. In other words, the main question is how to process as much data as possible in a decentralized and online fashion while keeping the communication overhead, memory and computational cost low [6].

B. Online Anomaly Learning and Forecasting Model from Large-scale Service of IoT

In service-oriented IoT environments, the utility-based co-operation service environment is composited by service consumers, service-oriented intermediaries, and service provider. Service consumer utilizes physical resources in a service-oriented network infrastructure by requesting infrastructure service from the service provider. Through such an infrastructure service paradigm, service encapsulates shares of networking resource from different service-oriented intermediaries in a set of service components, which are then assembled into an end-to-end service network topology by service-oriented intermediaries. Therefore, the service-oriented intermediaries consist of a series of tandem service components of IoT, each of which is a logical abstraction of the IoT large-scale nodes [14]. Fig.1 shows the online anomaly learning and forecasting model from IoT devices profiles.

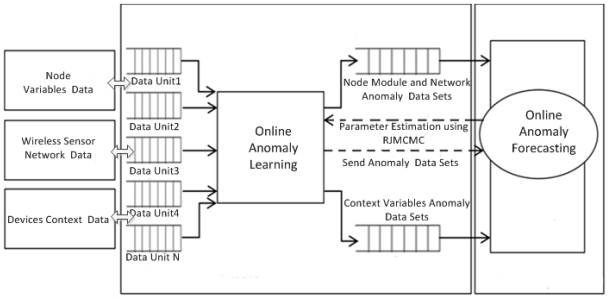

Fig. 1. The online anomaly parameter learning and forecasting model

In the model, we propose an integrated probabilistic model inspired by node status and stochastic service block models, to learn dependency structures from the combination of node status data and service status data. We consider node performance data in terms of directed edges (interactions) and service status data using stochastic service block. Explicitly, by convergence two data types, a node which is likely to have directed edges to members of a module as well as correlation with variables of module will be assigned as parent. A shorter version of this work was presented in [15]. The use of node status data enhances computational tractability and scalability of the method by restricting the space of possible dependency structures. We also show theoretically that the integration of node status data leads to model identifiability without extra structural assumptions.

IV. The Anomaly Parameter Learning and Detection using RJMCMC

A. Modeling Nodes Variables and Services Status Data from IoT Context

In this paper, we will first define a multivariate normal performance variables for nodes status profile1,...,N in current IoT condition, denoted as $X_i \sim \aleph(\mu_i, \Sigma)$,where X_i is a $N \times 1$ vector, with N being the total number of nodes. The covariance and mean capture two different aspects of the model regarding global dependency structures and context-specific effects of parents, respectively, as described below in [16].

the section define the covariance Σ to be independent of conditions and representing the strength of potential performance effects of one variable upon another, if the former is assigned as a parent of the modular containing the latter. In the section of gene expressions, Σ may represent the affinity of a anomaly-factor protein to a target gene promoter. The modular dependencies between performance variables imposes a structure on Σ. To construct anomaly tree structure, we relate sensor node variables to their parents through a regression $X_i = WX_i + \epsilon$,where $\epsilon = \aleph(m_i, I)$. W is a $N \times N$ sparse matrix in which element W_{nr} is nonzero if variable r is assigned as a parent of the node containing variable n. Here we assume W_{nr} has the same value for $\forall n \in M_k, \forall r \in P_{ak}$, which leads to identifiability of model (as explained in section 3. Then, assuming $I-W$ is invertible, $X_i = (1-W)^{-1}\epsilon$ which implies $\Sigma = (I-W)^{-T}(I-W)^{-1}$Therefore, we impose the unit dependency tree structure over Σ through W, which is easier to interpret based on A, S assignments.

The combination of different activities are represented as a decision tree for each modular of node k (figure 1). We represent a context-specific program as dependencies of variable means on parents activities in each context, such that μ_i^k for modular of node k is a linear mixture of means for parents of that modular of node: $\mu_i^k = \Sigma_{r-1}^{R_k} \gamma_i^r \mu_i^{P_{ak}}$ where $R_k$is the number of parents P_{ak} and γ_i^r are similar for all conditions i occurring in the same context. Thus, in general we can write $\gamma_i = \Gamma_i \gamma_i^r$, where γ_i^r contains the means of parents $1, ..., R$ in condition c. The $N \times R$ matrix Γ_i has identical rows for all variables in one modular of node based on the assignment functions A, S. The graphical model is summarized in figure 2. Thus the model for modular of node variables would be: $X_i \sim \aleph(\Gamma_i \gamma_i^r, (I-W)^{-T}(I-W)^{-1})$.

Given independent conditions, the probability of data $X = [X_1, ..., X_i]$ for i conditions given parameters can be written as multiplication of multivariate normal distributions for each condition: $P(X|A, S, \emptyset, \Sigma, Z^S) = \prod_{i=1}^{I} P(X_i|A, S, \emptyset_c, \Sigma, Z^S)$, where $\emptyset = \theta_1, ..., \theta_c$ denotes the set of condition-specific parameters $\theta_c = \gamma_i^r, \Gamma_i$for $i = 1, ..., I$and Z^S denotes the set of parent split-points for all unit. Then for each condition we have:$P(X_i|A, S, \emptyset_c, \Sigma, Z^S) = \frac{1}{(2\pi)^{(N/2)}|\Sigma^{1/2}|} \exp(-\frac{1}{2}(X_i - \gamma_i)^T \Sigma^{-1}(X_i - \gamma_i))$. Hence, this model provides interpretations for two types of influences of parents. By relating the distribution mean for variables in each modular of node and in each condition to means of their assigned parents (figure 1.B), we model condition specific effects of parents. Based on the states of parents in different contexts (partitions of conditions), this leads to a bias or large signal variations in node variables. Whereas, small signal changes (linear term) are modeled through the covariance matrix Σ which is independent of condition and is only affected by the global wiring imposed by dependency structures.

In order to analyze anomaly of IoT service performance,

we let anomaly performance data as a directed edge between a parent $r \in 1, ..., R$ and node $n \in M_k$, when r is assigned as a parent of the modular $r \in P_{ak}$ is defined as a directed link $L_{r \to n}$ where:

$$P(B_r \in P_{ak} \to n \in M_k | A, S, \pi_k^r) \sim Bernoulli(\pi_k^r) \quad (1)$$

The parameter π_k^r defines the probability of parent r influencing modular M_k (figure 2). In the gene network example, an interaction between a Transcription Factor protein binding to a motif sequence, upstream of target genes, which is common in all genes of a modular can be observed using ChIP data. Therefore, directed interactions from parents to all nodes in a modular would be

$$P(B_{M_k} | A, S, \pi_k^r) = \prod_{r \in P_{ak}} \prod_{n \in M_k} P(B_{r \to n} | A, S, \pi_k^r) \quad (2)$$

where π_k is the vector of π_k^r for all $r \in P_{ak}$ and for all nodes we have:

$$\begin{aligned} P(B|A, S, \pi) &= \prod_{k=1}^{K} \prod_{r \in P_{ak}} \prod_{n \in M_k} P(B_{r \to n} | A, S, \pi_k^r) \\ &= \prod_{k=1}^{K} \prod_{r \in P_{ak}} (\pi_k^r)^{srk} (1 - \pi_k^r)^{|M_k| - srk} \\ &\quad \prod_{r \neq P_{ak}} (\pi_0)^{srk} (1 - \pi_0)^{|M_k| - srk} \end{aligned} \quad (3)$$

with $\pi = \pi_1, ..., \pi_k$ and $s_{rk} = \sum_{n \in M_k} (B_{r \to n})$ is the sufficient statistic for the network data model and $|M_k|$ is the number of nodes in modular k and π_0 is the probability that any non-parent can have interaction with a module. In gene regulatory networks, π_0 can be interpreted as basal level of physical binding that may not necessarily affect gene transcription and thus regulate a gene. In the context of stochastic block models, the group of parents assigned to each modular can be considered as an individual block and thus our model can represented as overlapping blocks of nodes. The likelihood of the model $M = A, S, \theta, \Sigma, Z^S, \pi$ given the integration of node variables and service component data is: $P(X, B|M) = P(X|A, S, \emptyset, \Sigma, Z^S) P(B|A, S, \pi)$. With priors for parameters M the posterior likelihood is:$P(M|X, B) \propto P(M) P(X, B|M)$.

B. Anomaly Performance Parameter Learning Process

To realtime update means of the profile data , We use a Gibbs sampler to obtain the posterior distribution $P(X, B|M)$ and design Metropolis-Hastings samplers for each of the parameters $\emptyset, \Sigma, \pi$ conditioned on the other parameters and data X, B. We use Reversible-Jump MCMC [16] for sampling from conditional distributions of the assignment and structure parameters A, S. we only need to sample one value for means of parents assigned to the same module. This set of means of distinct parents γ_i^r are sampled with a Normal proposal (Algorithm 1). Similarly we sample the parameters $\gamma_i^r, z_k^r, , \pi_k^r, ,$ corresponding to parent $r \in P_{ak}$ of module k, from normal distributions. To update covariance Σ, each distinct element of the regression matrix W corresponding to a module k, denoted as ω_k, is updated. Due to the symmetric proposal distribution, the proposal is accepted with probability $P_{mh} = \min 1, \frac{P(M^{j+1}|X,B)}{P(M^j|X,B)}$,where $M^{(j)} = A, S, \emptyset, \Sigma, Z^S \pi$.

Input: $NodePerformanceParameterData$, $NetworkData$
Output: Performance Weakening: $P(S|A, \emptyset, \Sigma, Z^S \pi, X, B)$

for $t = 1$ **to** T **do**
  Sample $A^{(j+1)}$ given $A^{(j)}$ using Alg2 in (Azizi et al.,2014);
  Sample $S^{(j+1)}$ given $S^{(j)}$ using Alg3 in (Azizi et al.,2014);
  for *node* $k = 1$ **to** K_j **do**
    Propose $\omega_k^{j+1} \sim \aleph(\omega_k^{j+1}, I)$;
    Accept with probability P_{mh}update $\Sigma^(j+1)$;
    for *parents* $r = 1$ **to** R_j **do**
      Propose $z_k^{r(j+1)} \sim \aleph(z_k^{r(j)}, I)$; accept with P_{mh};
      Propose $\pi_k^{r(j+1)} \sim \aleph(\pi_k^{r(j)}, I)$; accept with P_{mh};
    end
  end
  for *condition* $i = 1$ **to** I **do**
    Propose $\mu_i^{r(j+1)} \sim \aleph(\mu_i^{r(j)}, I)$; accept with P_{mh};
    Propose $\gamma_k^{r(j+1)} \sim \aleph(\gamma_i^{r(j)}, I)$; accept with P_{mh};
  end
end

Algorithm 1: Node Performance Parameter Learning [16]

Learning the assignment of each node to a module, involves learning the number of modules. Changing the number of modules however, changes dimensions of the parameter space and therefore, densities will not be comparable. Thus, to sample from $P(A|S, \emptyset, \Sigma, Z^S \pi, X, B)$, we use the Reversible-Jump MCMC method [16], an extension of the Metropolis-Hastings algorithm that allows moves between modulars with different dimensionality. In each proposal, we consider three close move schemes of increasing or decreasing the number of units by one, or not changing the total number. For increasing the number of units, a random node is moved to a new module of its own and for decreasing the number, two units are merged. In the third case, a node is randomly moved from one modular to another modular, to sample its assignment (Algorithm 2 in [16]. To sample from the dependency structure (assignment of parents) $P(S|A, \emptyset, \Sigma, Z^S \pi, X, B)$, we also implement a Reversible-Jump method, as the number of parents for each unit needs to be determined. Two proposal moves are considered for S which include increasing or decreasing the number of parents for each modular, by one [16].

C. Utility Maximization of Online Anomaly Detection Process

In IoT-based service topology, this paper defines service anomaly forecasting model that consist of service-oriented intermediaries and service providers, denoted by i=1,2,...,r . Let C_i be the capacity profile of node i, $C = [c_1, c_2, ..., c_r]$. Rate of service consumers denoted by $x_j, j = 1, 2, ..., s$. Each node j has k^i available path l, $l = 1, 2, ..., p$ from the node j to the logical destination node i corresponding to the service that is being consumed by a user. There are k^i acyclic paths for

source j represented by a $s \times k^i$0-1 matrix H^j that describes the mapping of nodes i on paths $l, l = 1, 2, ...p$ for particular users j , Defines the $J \times k$ matrix as:

$$H_{il}^j = \begin{cases} 1, & \text{if path } l \text{ of source } j \text{ uses resource } i \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{H} = [\mathbf{h^1}, \mathbf{h^2}, ..., \mathbf{h^s}]$$

H defines the IoT-based service topology, let ω^j be a $k^j \times I$ vector where the jth this require $\omega_j^i \in [0, 1]$ for multipath messages routing. Collects the vectors ω^i , into a $k \times s$ block diagonal matrix M. Let be the set off all such matrices corresponding to multipath routing as

$$\mathbf{M} = \{\mathbf{m}|\mathbf{m} = \mathbf{diag}(\omega^\mathbf{1}, ..., \omega^\mathbf{s}) \in [\mathbf{0}, \mathbf{1}]^{\mathbf{k} \times \mathbf{s}}, \mathbf{I^T}\omega^\mathbf{s} = \mathbf{1}\}$$

As mentioned above, H defines the set of paths available to each source and also represents the IoT-based service topology. M defines how the sources load balance across the multiple paths. This paper defines a $j \times s$ routing matrix $r = HM$ that specifies the fraction of j flow at each resource . The set of all multipath routing matrices is

$$\mathbf{R} = \{\mathbf{r}|\mathbf{r} = \mathbf{h} \times \mathbf{m}, \mathbf{m} \in \mathbf{M}, \mathbf{h} \in \mathbf{H}\}$$

A multipath routing matrix in R is one whose entries are in the range [0, 1]:

$$R_{ij} = \begin{cases} > 1, & \text{if resource } i \text{ is in a path of source } j \\ = 0, & \text{otherwise} \end{cases}$$

The path of source$j$ is denoted by $r^j = [R_{1j}, R_{2j}, ..., R_{rj}]^T$ the jth column of the routing matrix R .

In order to resolve service anomaly forecasting model between service consumers and service provider, this paper defines Service Utility Maximization (SUM) from IoT Network Utility Maximization (NUM) theory [19] [20]. The algorithms of SUM consider a service-oriented network where each source j has a utility function $U_j(x_j)$ as a function of its total transmission rate x_j . The service anomaly forecasting model problem over source rate vector x_j, for a given fixed routing matrix R, is:

$$\max_{r \in R} \sum_{j=1}^{s} \max_{x \geq 0} [U_j(x_j) - \sum_{g \in G_j} g(x_j, d_j, h_j)] \tag{4}$$

$$s.t. \quad rx \leq c \tag{5}$$

The goal is to maximize aggregate user utility by varying x_j (but not R), subject to the linear flow constraint that path loads cannot exceed service-oriented capacity. Congestion-control algorithms implicitly solve (2), with variants maximizing different utility functions. In the formulation, $g(.)$ is base cost function on service quality (delay d_i , hot h_i). It is well-known that the utility functions can be picked based on several different grounds. First, a utility function can capture a user‘s degree of satisfaction with a particular throughout. Second, a utility function can be viewed as a measure of the elasticity of the traffic. Third, the aggregate utility captures the efficiency of the system in allocating bandwidth to the traffic. Fourth, some utility functions can lead to fair resource allocation. A particular family of widely used utility functions is parameterized by $\alpha \geq 0$: if $\alpha = 0$, then $U_j(x_j)$ = log(x), else $U_j(x_j)$=$(1-\alpha)^{-1}x^(1-\alpha)$. Maximizing these fair utilities over linear flow constraints leads to rate-allocation vectors that satisfy the definitions of α-fairness in the economics literature. Equation (1)optimizes "social welfare" by maximizing utility over both source rates and routes. However, (4) is not a convex problem because the feasible set specified by $R(t)x \leq c$ is generally not convex.

Now replace the problem by defining the $K^j \times I$ vectors y^j in terms of the scalar x^j , and the $K^j \times I$ vectors ω^j as the new variables:

$$y^j = x_j \omega^j \tag{6}$$

The mapping from (x_j, ω^j) to y^j is one-to-one; the inverse of (5) is $x_j = I^T y^s$ and $\omega^j = y^j \div x_j$. Change the variables in (1) and (2) from $(x_j, omega^j)$ to y^j , by substituting $x_j = I^T y^s$ and $rc = Hy$, obtaining the equivalent problem:

$$\max_{y \geq 0} \sum_{j=1}^{s} [U_j(I^T y^j) - \sum_{g \in G_j} g(I^T y^j, d_j, h_j)] \tag{7}$$

$$s.t. \quad Hy \leq c \tag{8}$$

Provided that the functions $U_j(.)$ and $g(.)$ are strictly concave, this is a strictly concave problem with a linear constraint, and therefore, has no duality gap. To find a distributed algorithm that solves (4) and (5), this verify the problem through its Lagrangian dual. Is:

$$L(y, p) = \sum_{j=1}^{s} [U_j(I^T y^j) - \sum_{g \in G_j} g(I^T y^j, d_j, h_j)] \tag{9}$$

$$- \sum_{j=1}^{s} p_j (Hy - c) \tag{10}$$

Under a more critical assumption of strict concavity on utility functions, there always exists a unique optimal solution x to the maximization problem. Where $p = [p^1, p^2, ..., p^s]^T$ is a $J \times I$ vector of Lagrange multiplies associated with the capacity constraint on resource j. Letting $p_{ij} = \Sigma_{j=1}^{s} H_{il}^j p^j$ and $p = [p_{j1}, p_{j2}, ..., p_{jk^s}]^T$. They continue by formulating the objective function of the dual problem as:

$$D(p) = \sum_{j=1}^{s} [U_j(I^T y^j) - \sum_{g \in G_j} g(I^T y^j, d_j, h_j)] \tag{11}$$

$$-\sum_{j=1}^{s} p_j y^j + \sum_{j=1}^{s} p_j c \tag{12}$$

Let $B_j(y^j, p_j)$ be defined as

$$B_j(y^j, p_j) = \max_{y \geq 0} \sum_{j=1}^{s} [U_j(I^T y^j) - \sum_{g \in G_j} g(I^T y^j, d_j, h_j)] \tag{13}$$

$$-\sum_{j=1}^{s} p_j y^j \tag{14}$$

Since $D(p)$ is separable in s , we can swap the order of the maximization and the summation, forming the following equivalent equation:

$$D(p) = \sum_{j=1}^{s} B_j(y^j, p_j) + \sum_{j=1}^{s} p_j c \tag{15}$$

The dual problem of (4) and (5) corresponds to minimizing over the dual variables, i.e.

$$minD(p) \tag{16}$$

Since the objective function of the primal problems (4) and (5) is strictly concave, the dual problem is always differentiable. The gradient of D is

$$\frac{\delta D}{\delta p^j} = c^j - \sum_{j=1}^{s} \sum_{i=1}^{k^i} H_{il}^j y_i^{*s} \tag{17}$$

Where y_i^{*s} comes from the solution of $B_i(y^i, p_i)$. Using gradient descent iterations on the dual variables yields the following equation:

$$p^j(t+1) = [p^j(t) + \beta^j(c^j - \Sigma_{j=1}^{s}) \Sigma_{i=1}^{k^j} H_{il}^s y_i^s(t))] \tag{18}$$

Where y_i^{*s} is the solution of the following optimization problem at time t:

$$y_j^i(t+1) = \max_{y_j^i} U_j(I^T y^j) - \sum_{g \in G_j} g(I^T y^j, d_j, h_j) \tag{19}$$

$$-y_j^i \sum_{i=1}^{r} p^i(t) H_{il}^j \tag{20}$$

The combine solution of (16),(17) completes the SUM algorithm that solves (1). The resources alter the rates of each source $y_j^i(t+1)$ by this feedback from downlink resource via congestion prices p^i. Each resource maximizes the utility for source j while balancing the price of the placing load on a path l . The path price is the result of the source rate with the price per load for path l. The result of the rates $y_j^i(t+1)$ at the resources determines the total traffic that traverses one resource. The resulting load through each resource serves as this feedback that is used to compute the congestion price p^i. As it is classified as a separable, strictly concave nonlinear optimization problem with linear constraints; the combine of a gradient projection algorithm applied to such a problem is well known for sufficiently small step sizes $\alpha > 0$.

TABLE I
NODE THROUGHPUT AND OVERALL UTILITY

Nodes	g=0.01	g=0.03	g=0.05
Node A	(243.77,256.23)	(233.56,266.44)	(221.24,268.55)
Node B	(243.77,256.23)	(233.56,266.44)	(221.24,268.55)
Node C	(243.77,256.23)	(233.56,266.44)	(221.24,268.55)
Node D	(243.77,256.23)	(233.56,266.44)	(221.24,268.55)
Service provider E	(243.77,0)	(233.56,0)	(221.24,0)
Service provider F	(0,256.23)	(0,266.44)	(0,268.55)
Service provider G	(243.77,0)	(233.56,0)	(221.24,0)
Service provider H	(0,256.23)	(0,266.44)	(0,268.55)
Service provider J	(243.77,0)	(233.56,0)	(221.24,0)
Service provider K	(0,256.23)	(0,266.44)	(0,268.55)
Overall Utility	78.55	77.89	77.01

V. RESULT

To validate our proposal, we have used data collected from sensors deployed in an actual living lab realized in the context of smart beijing city, in particular, pressure, PIR, acoustic, temperature, humidity, and light intensity sensors. We examined the measurements collected every 5 seconds in 30 consecutive days at a base station. For the dataset, the ground truth is also available. The dataset is of medium size, consisting of slightly more than 50000 samples.

In the section, we describes the occurrence of data and system faults from 50 sensor nodes samples by applying the RJMCMC method to the given dataset. The dataset exhibited a mixture of offset, gain and stuck-at data anomalys because of system faults such as the service messages faults. We forecast utility maximization of anomaly by MATLAB CVX tool [21]. The capacities of the intermediary nodes are less than the aggregate capacity of the services; this allows us to easily study how IoT service adapts the allocation of flows through changing incoming rates and external parameters such as measured average delay. The capacity of each intermediary and service provider is 600 requests per second. The topology is represented in the $J \times K$ 0-1 matrix shows in table.1. Service messages faults forecasting scene is constructs by IoT-based service topology and assembling resources from four service-oriented intermediaries (Nodes A, B, C, D), which are configured to forward requests for either type of service from the consumers to the service logical destination node. Considering the scenarios in which the service-oriented provider environment (Service provider E, Service provider F,

Service provider G, Service provider H, Service provider J, Service provider K) offers the semantically equivalent service to two consumers (Service1, and Service2). Service 1 starts a service session g_1 for transmitting a stream of video packets, while Service 2 generates a service session g_2 to deliver a flow of audio traffic. Both Service 1 and Service 2 require a small cost price. The traffic parameters for g_1 are peak rate $s = 2Mb/s$, sustained rate $\delta = 1Mb/s$, and the maximum burst size $\sigma = 100kbits$. the traffic parameters for g_2 are peak rate $s = 3Mb/s$, sustained rate $\delta = 0.7Mb/s$, and the maximum burst size $\sigma = 200kbits$. Each intermediary is assumed to provide service with a maximum transmission unit $M = 1000bytes$.

Each Node of service traffic holds two congestion property of delay-sensitive, and hop-count-sensitive in IoT-based service topology. The delay-sensitive congestion property d_s is weighted by the r_s parameter; if a service is not delay-sensitive, then $r_s = 0$ for the service, otherwise, it should be selected to be proportional to the overall utility gained from the service. The property compares the total service delay d_s for each path against a delay threshold t_s; if the measured delay exceeds the threshold, the exponential term of the function grows quickly to divert traffic away from paths containing the offending nodes. d_s is computed by multiplying the relevant portion of the topology matrix H with a vector z_s of measured service delays at each node:

$$d_s = \{(H^s)^T z^s\}$$

Hop-count-sensitive congestion property $h_s$is weighted by the parameter; if a service is not sensitive to the hop count, then for the service, otherwise it should be selected to be proportional to the overall utility gained from the service. It show the ability of the IoT-based service to maximize the overall utility of the system while favoring path that have smaller hop counts.

$$h_s = \{(H^s)^T r^s\}$$

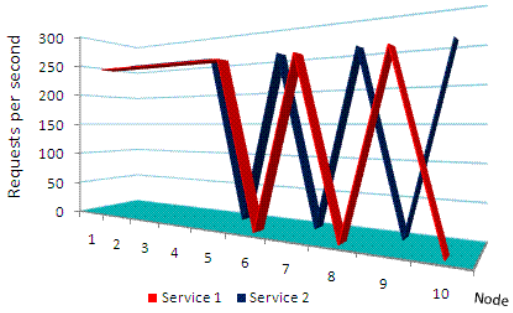

Fig. 2. Node Overall Utillity at g=0.01

In this experiment, this varies the input rates at $g(delay - sensitive, hop - count - sensitive)$. g the measured at each Node for Service 1 requests. As the g approaches and subsequently passes the g threshold (0.01), it show that the allocations for Service 1 requests should tend to avoid paths that contain each Node. Service 2 requests should be insensitive to the g measurements.

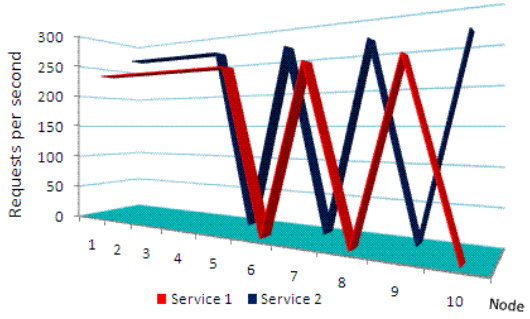

Fig. 3. Node Overall Utillity at g=0.03

This experiment begins with the vector $H = [1100111110]$, this means that all nodes are currently processing service 1 requests in an average of 1 delay unit (milliseconds). When all offered loads are all 500 requests per second, the system should allocate $1/2$ of the resources to Service 1, and the other $1/2$ to Service 2.

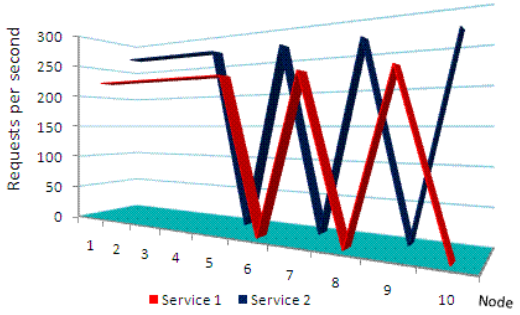

Fig. 4. Node Overall Utillity at g=0.05

As while this continues to increase the g at each Node for Service 1 traffic, we see that the system slowly reduces the amount allocated to Service 1 paths that include each Node until the threshold is met. Then,the system explicitly avoids allocating any traffic to Service 1 paths that include each Node. The system is aware of Service 2 to g, so as Service 1 traffic is diverted away from each Node, Service 2 traffic is diverted to each Node in order to make better use of the available resources. This can be clearly seen in Fig.2,3,4, where traffic is routed on to alternate paths in order to maintain the overall utility of IoT-based service topology.

VI. Conclusion

The paper proposes a new online anomaly learning and forecasting model for large-scale service of Internet of Thing. The proposed model learns the dependency structures of dynamics network and service data from highly dynamic distributed IoT system. Furthermore, our model uses the reversible-jump MCMC learning to online learn anomaly-free of dynamics

network and service data. It can be used to identify and classify anomaly data of dynamics network and node variables, which considers the structural relations between nodes dynamically created. We then perform to estimate the parameters of IoT-based service topology by network utility maximization theory. The results of experiment shows high performance on synthetic data and interpretable structures from an actual living lab realized in smart Beijing city. The future work will focus on the scalable of the framework to a larges set of anomaly types and a broader probability evaluation with actual datasets coming from real-world.

ACKNOWLEDGMENT

The author also would like to thank anonymous editor and reviewers who gave valuable suggestion that has helped to improve the quality of the manuscript. This research has been supported by the Project for 2015 National Key Technologies RD Program No. $2015BAH04F01$.

REFERENCES

[1] L. Atzori, A. Lera, G. Morabito, "The Internet of Things: A survey", In *Computer Networks*, 54(15):2787-2805, October 2010.

[2] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, "The Internet of Things:Vision, applications and research challenges", In *Ad Hoc Networks*, 10(7):1497-1516, April 2012.

[3] M. Xie, J. K. Hu, S. Han, H.H. Chen, "Scalable Hypergrid k-NN-Based Online Anomaly Detection in Wireless Sensor Networks", In *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, 24(8):1661-1670, AUGUST 2013.

[4] S. R West, Y. Guo and X R. Wang, "automated fault detection and diagnosis of HVAC subsystems using statistical machine learning", In *12th International Conference of the International Building Performance Simulation Association*, 2011.

[5] E. U. Warriach, M. Aiello, and K. Tei, "A machine learning approach for identifying and classifying faults in wireless sensor networks", In *IEEE 15th International Conference on Computational Science and Engineering*, 2012.

[6] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Outlier Detection Techniques for Wireless Sensor Networks: A Survey", In *IEEE Communications Surveys and Tutorials*, 12(2):159-170, 2010.

[7] Y. Yao, A. Sharma, L. Golubchik, R. Govindan, "Online anomaly detection for sensor systems: A simple and efficient approach", In *Performance Evaluation*, 67(11):1059-1075, November 2010.

[8] H. Sagha, H. Bayati, J. R. Milln, R. Chavarriaga, "Online anomaly detection and resilience in classifier ensembles", In *Pattern Recognition Letters*, 34(15):1916-1927, November 2013.

[9] O. Salem, Y. N. Liu, A. Mehaoua, "A Lightweight Anomaly Detection Framework for Medical Wireless Sensor Networks", In *IEEE Wireless Communications and Conference (WCNC):service and applications*, 2013.

[10] X. Ma, D. Yang, S. Tang, Q. Luo, D. Zhang, and S. Li, "Online Mining in Sensor Networks", In *IFIP international conference on network and parallel computing*, Vol. 3222, pp. 544-550, 2004.

[11] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogerakiand, and D. Gunopulos, "Online Outlier Detection in Sensor Data using Nonparametric Models", In *J. Very Large Data Bases*, VLDB 2006.

[12] F. Martincic and L. Schwiebert, "Distributed Event Detection in Sensor Networks", In *Proc. International Conference on Systems and Networks Communication*, pp. 43-48, 2006.

[13] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly Detection in Wireless Sensor Networks: A Survey", In *J. Network and Computer Applications*, 34(4):1302-1325, Jul. 2011.

[14] S. Alam, J. Noll, "Virtualizing Sensor for the Enablement of Semantic-aware Internet of Things Ecosystem", In *INTERNATIONAL JOURNAL OF DESIGN, ANALYSIS AND TOOLS FOR CIRCUITS AND SYSTEMS*, 3(1):41-51, August 2011.

[15] E. Azizi, "Joint Learning of Modular Structures from Multiple Data Types", In *NIPS Workshop of Frontiers of Network Analysis: Methods, Models, and Applications*, 2013.

[16] E. Azizi, E. Airoldi, J. Galagan, "Learning Modular Structures from Network Data and Node Variables", In *International conference on machine learning*, 2014.

[17] Joshi, A., De Smet, R., Marchal, K., Van de Peer, Y., and Michoel, T, "Module networks revisited: computational assessment and prioritization of model predictions", In *Bioinformatics*, 25(4):490-496, 2009.

[18] Azari Soufiani, H. and Airoldi, E.M, "Graphlet decomposition of a weighted network", In *Journal of Machine Learning Research*, JMLR WCP 22:5463, 2012.

[19] M.Chiang, S.H.Low, A.R.Calderbank, and J.C.Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures", In *Proccessing of IEEE*, 95(1):, Jan, 2007.

[20] J.Wang, L.Li, S.H.Low, and J.C.Doyle, "Cross-Layer Optimization in TCP/IP Networks", In *IEEE/ACM Trans. Networking*, 13(3):, June, 2005.

[21] M.Grant and S.Boyd, "CVX: Matlab Software for Disciplined Convex Programming", In *PrenticeHall*, pages 261–271, Feb, 2008.