12-2014

Detecting flow anomalies in distributed systems

Freddy Chong-Tat CHUA

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

Bernardo HUBERMAN

DOI: https://doi.org/10.1109/ICDM.2014.94

Detecting Flow Anomalies in Distributed Systems

Freddy C. Chua
Mechanisms and Design Lab
HPLabs, Palo Alto, CA 94304, USA
freddy.chua@hp.com

Ee-Peng Lim
School of Information Systems
Singapore Management University
eplim@smu.edu.sg

Bernardo A. Huberman
Mechanisms and Design Lab
HPLabs, Palo Alto, CA 94304, USA
bernardo.huberman@hp.com

***Abstract*—Deep within the networks of distributed systems, one often finds anomalies that affect their efficiency and performance. These anomalies are difficult to detect because distributed systems may not have sufficient sensors to monitor the traffic flow within their interconnected nodes. Without early detection and corrections, these anomalies can aggravate over time and possibly cause disastrous outcomes in the system in an unforeseeable future. Using only coarse-grained information from the two end points of network flows, we developed a network transmission model and localization algorithm that detects and ranks the location of anomalies. We evaluate our approach using passengers' records of an urbanized city's public transportation system, and correlate our findings with passengers' postings on social media microblogs. Our experiments show that our localization algorithm gives a better ranking of anomalies than standard deviation measures drawn from statistical models. Our case study also demonstrates that transportation events reported in social media microblogs often match the locations of our detected anomalies detected with our algorithm.**

I. Introduction

In a complex world, networks offer a useful abstract representation for organizing the relationships between entities of interest in distributed systems. Entities are represented as nodes while edges connecting pairs of nodes represent relationships between them. Examples of pervasive distributed systems are social networks [1], protein networks, computer networks [2], [3], transportation networks [4], [5], logistical networks [6], neurological networks, organizational networks [7], wireless sensor networks (Internet of Things), electrical networks, and many more.

A functional network requires reliable and consistent flow of entities through its if it is to achieve its objectives. However, it is inevitable that the building blocks of the system deteriorate non-uniformly over time, leading to occasional anomalous behavior in certain parts of the system. Anomalies in such systems can disrupt normal operations and prevent the network from meeting its objectives in a timely manner.

While critical anomalies leading to catastrophic failures are noticed and addressed by the stakeholders of the distributed system, it is more challenging to recognize the *non-critical* ones that result in a lower than optimal efficiency of the system. Since in the latter case the system can continue to function without corrections, non-critical anomalies are often hard to locate and *ignored*. But if not corrected, non-critical anomalies can aggravate over time and lead to the catastrophic failures of the system in an unforeseeable future.

Before proceeding further, we use Figures 1 and 2 to illustrate the problem. Figure 1 shows a distributed system where entities flow from node to node through directed edges. The edges connecting nodes $\{a, b, c, d, e\}$ form a route through which the entities flow. We do not assume that a and e are always the origin and destination of every entity flow, i.e. entities within the distributed system could originate from or terminate at any of the intermediate nodes $\{b, c, d\}$. The dotted $-\cdot\rightarrow$ line indicates the possibility of an existing anomaly that would disrupt the regular flow of entities along this route.

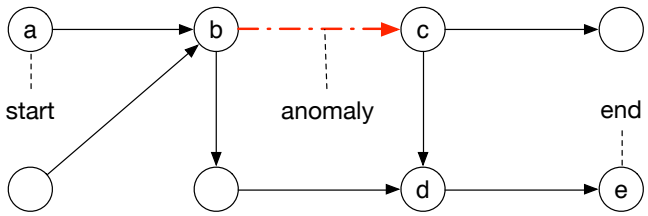


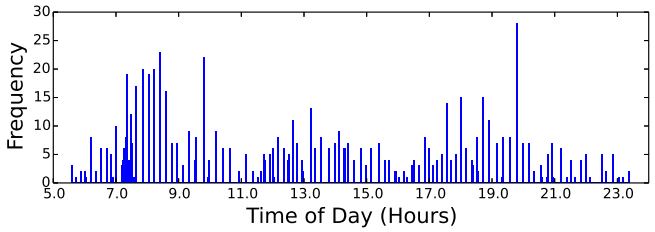
Fig. 1. Entities Flow in Networks

Figures 2a and 2b show the histograms of entities starting its flow at the origin node a, and ending its flow at the destination node e. In the histograms, the x-axis represents the hour of the day, and each bin on the x-axis has a time interval of two minutes. The y-axis shows the number of flows starting or ending at the time corresponding to the bins. The phenomenon that we could immediately observe is that while the start node shows a *regular* transmission of entities, the end node receives the entities at *irregular* intervals. This could suggest the presence of an anomaly within the path such as the segment connecting node b and c, suffering from a severe network congestion as shown in the example of Figure 1.

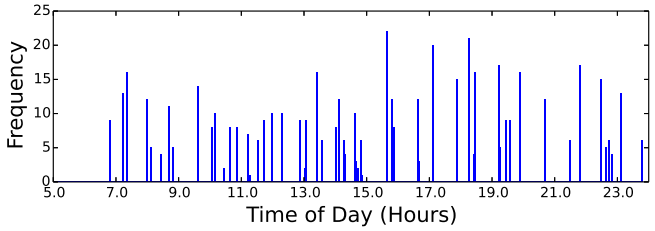
In this paper we propose a *non-intrusive solution* to early detection of such anomalies. Our proposed non- solution relies on *temporal* data related to the *flow* of entities from an *origin* node to a *destination* node. Since the solution should be non-intrusive, we only require temporal information from the two (origin and destination) end points while assuming that detailed knowledge of the flow through the intermediate nodes of its path is missing or difficult to obtain.

We formally define our problem based on the assumption that the following recorded data is available for our analysis. That is, given a set of records R of a distributed system, each record $r \in R$ contains the following,

1) Spatial: The origin node x_r, and destination node y_r of entity flow in r.
2) Temporal: The time $t(x_r)$ when entity flow starts at the origin and the time $t(y_r)$ when entity flow ends



(a) Histogram at the start node of an entity-flow route



(b) Histogram at the end node of an entity-flow route

Fig. 2. Histograms for start and end nodes of a flow route

at the destination.

3) Cost: The distance d_r from x_r to y_r traveled by the entity, or the non-temporal cost incurred due to the entity flow.
4) The path p_r taken by r. The path consists of the sequence of nodes that the entity visits for it to flow from node x_r to node y_r. In situations where complete knowledge of the network or path is not known, it would still be possible to infer the path based on the distance traveled.

A pair of consecutive nodes i, j in path p_r, forms a segment $s_{i,j}$. We determine whether the observed amount of time $\hat{t}_r$, taken for entity flow in p_r deviates significantly from the expected amount of time t_r. For all the records $r \in R$ with observed time that deviates significantly from the expected time, we locate the segments $s_{i,j} \in p_r$ that are likely to be the cause of the deviations.

This task is challenging due to the lack of knowledge on the time it takes for entities to flow through the individual segments of path p_r. We need to infer the expected time for each segment based on the set of available records we have and the limited amount of knowledge each record contains.

Our technical contribution consists of the proposal of a network transmission model that performs the inference on spatial-temporal data where knowledge of the temporal trajectory is missing or not available. Such situations are common when it is not possible to install sensors for monitoring the internal networks of distributed systems. For example, in computer networks, the application layer only have knowledge of the two endpoints, and does not know the behavior of the intermediate nodes. Another example is transportation networks, which only records the passengers boarding and alighting stations for the purpose of calculating their transportation fare.

Building on the network transmission model, we further propose an algorithm that ranks the anomalous data in order of importance by measuring how much impact each anomalous data has on others. Then using the ranking, we can isolate the location of where the anomalies occur in the distributed system, listed in descending order of importance. We test our models and algorithms on the network data of a physical transportation system and verify the accuracy of the detected anomalies through case studies using social media data.

We present an overview of prior work on anomalies detection in Section II. Section III describes the details of our proposed network transmission models as well as two baselines for comparison. Using the results of our proposed model, we utilize the algorithm as described in Section IV for locating the segments of where the anomalies might have occurred. Then we evaluate the performance of our proposed models and the localization algorithm in Section V. We conclude our work in Section VI and highlight certain possible extensions that can be made for this area of research.

II. Related Work

Overview of Anomaly Detection. Chandola et al. [8] gave a comprehensive survey on the topic of anomaly detection for general scenarios. Chandola et al. defined an anomaly as a pattern that does not conform to expected normal behavior. But the notion of expected normal behavior depends on the application domains and types of input data. Chandola et al. surveyed a broad overview of various techniques used in anomaly detection; classification-based [9], [10], nearest neighbor approach [11], clustering-based [12], statistical-based (including parametric [13] and non-parametric [14]), information-theoretic-based [15], and spectral anomaly detection techniques [6]. Although our network transmission model can be classified as one of these methods, the second portion of our paper that measures the impact of anomalous data has not been used before. The kind of input data we used for our anomalies detection is also unique and contains inherent difficulties, which is addressed by our network transmission model.

Job Anomaly Detection. Fu et al. [16] addressed the detection for two broad classes of anomalies; The first class is work flow execution anomalies and the second class is execution low performance anomalies. Our focus on non-critical anomalies is similar to Fu et al.'s definition of execution low performance anomalies. Their detection method is based on text analysis of logs generated by parallel frameworks such as Hadoop. However, we do not require the usage of logs, which is usually difficult to obtain from distributed systems.

Computer Network Anomaly Detection. [2], [3] proposed anomalies detection methods specifically for computer networks. Such anomalies could come from hackers that infiltrate the network and attempt to compromise the security of the computer network. Kind et al. [2] suggested that, the existence of anomalies may not necessarily cause significant changes in the performance and speed of network flow. They proposed a feature-based anomaly detection method that uses the information in headers of computer network packets. Sengar et al. [3] proposed a behavioral distance metric that is adaptable for online detection of streaming network packets. While the kind of anomalies Kind et al. [2] and Sengar et al. [3] studied are also elusive in nature, it only applies to a subset of distributed systems such as computer networks with security vulnerabilities. On the other hand, the models and

algorithms we propose in our paper is well suited for many kinds of distributed systems that do not have the same kind of security flaws found in computer networks.

Anomaly Detection in Transportation. Agovic et al. [6] proposed a manifold embedding based method for detecting anomalies in transportation. Their algorithm takes in high dimensional feature vectors and reduces it to low dimensional representations for better efficiency of detecting anomalies. However, in the network data we look at in this research, our data is low dimensional and coarse-grained information of network flow in distributed systems. We use low dimensional information and reconstruct the high dimensional information in order to obtain a better representation of the network flows.

Fadlil et al. [4] used data that has multiple sensors monitoring different variables of the road conditions. The multiple sensors provided a multi-view of readings that is high dimensional which required manifold embedding [6] for clustering of the data points into two clusters. From the two clusters, Fadlil et al. chose the smaller cluster as the set of anomalous data points. But their approach requires detailed data such as readings from multiple sensors. On the other hand, our algorithm is suitable for distributed systems that do not have access to many sensors.

Thottan and Ji [17] defined network anomalies as network operations that deviate from normal network behavior. They proposed a method of network anomalies detection by monitoring several variables of network operations over time, and model the time series as an autoregressive process. The presence of abrupt changes in the time series is detected as anomalies. In our case, apart from detecting statistical deviations, we go one step further to detect the location of where the anomalies occur in the distributed system.

Pan et al. [18] address the problem of detecting and describing traffic anomalies using crowd sensing with Beijing's taxis' GPS data and China's Weibo[1] social media data. The anomaly here refers to a deviation in traffic volume on segments of road during some special events. The proposed detection algorithm is straightforward because of the availability of GPS data at regular and fine grain time intervals. Their focus is on two special indexing data structures that improve the algorithm efficiency.

Liu et al. [19] proposed the discovery of causal interactions among traffic outliers. The proposed framework first partitions the geographical space into regions represented by nodes in a region graph. Edges between nodes represent the traffic flow between regions. By comparing the features of each edge across different time frames, Liu et al. is able to identify the outlying travel trajectories. From the detected outliers, an outlier tree is constructed with each node representing an outlier trajectory. The parent of an outlier node occurs before in time and the destination of the parent is said to be the cause or origin of the child.

[20]–[22] analyzed taxi GPS data to detect drivers who overcharge their passengers by deliberately taking the longer route to reach the destination. The general idea for finding these anomalous routes is to compare the route taken for each pickup and destination points and obtain a measure of how much it deviates from the usual routes.

Chawla et al. [23] proposed an algorithm to detect anomaly based on Pinciple Component Analysis (PCA). Chawla et al. [23] represent the traffic data into two matrices, 1) the link-path matrix and the 2) link-time matrix. Then using PCA, Chawla et al. [23] is able to factorize the matrices into eigenvectors with its respective eigenvalues. The eigenvectors corresponding to large eigenvalues represent the norm, while those eigenvectors corresponding to lower eigenvalues represent the anomalies. Using these anomalies, Chawla et al. [23] tested their method to see if they could determine the root cause on synthetically generated data sets.

All of these previous works [18]–[23] require fine-grained sampling of GPS and time for anomaly detection and is limited only to spatial-temporal situations such as transportation networks. Our approach does not require GPS data and is made general for most forms of network traffic in distributed systems.

III. Network Transmission Models

To find anomalies in the network, we analyze the set of entities flow records R from a distributed system. But we must first determine which recorded entity flow $r \in R$ is anomalous before we could proceed further with the localization task. A record r is anomalous, if the observed time $\hat{t}_r$ taken to complete the distance d_r deviates significantly from the expected value t_r given to us by a statistical model.

A. Baseline 1

For our first baseline, we assume that every recorded entity flow r travels at a constant speed c to cover the total distance d_r required to reach its destination. The distribution of time taken t_r for record r is given by the following Gaussian distribution,

$$t_r \sim \mathcal{N}\left(\frac{d_r}{c}, d_r\sigma^2\right)$$

The unknown value of c can be obtained by minimizing the following Sum-of-Squares error,

$$\sum_{r\in R}(t_r - \hat{t}_r)^2$$

Thus allowing us to obtain,

$$c = \frac{\sum_{r\in R} d_r}{\sum_{r\in R}\hat{t}_r} \qquad \sigma^2 = \frac{\sum_{r\in R}\left(\hat{t}_r - \frac{d_r}{c}\right)^2}{\sum_{r\in R} d_r}$$

B. Baseline 2

The second baseline model which provides a more discriminative estimation than the first baseline is to assume a speed c_p for each distinct path p in the network.

$$t_r \sim \mathcal{N}\left(\frac{d_r}{c_p}, d_r\sigma^2\right)$$

The estimation for c_p and σ^2 can be easily extended from the first baseline to obtain the following,

$$c_p = \frac{\sum_{r\in R_p} d_r}{\sum_{r\in R_p}\hat{t}_r} \qquad \sigma^2 = \frac{\sum_{p\in P}\sum_{r\in R_p}\left(\hat{t}_r - \frac{d_r}{c_p}\right)^2}{\sum_{r\in R} d_r}$$

[1]This service resembles Twitter but is catered for the Chinese population in Chinese language.

where P is the set of possible paths and R_p is the set of records that took the path $p \in P$.

C. Edge-based Model

We propose the Edge-based model that models the speed of individual edges in the network. Given an entity flow record r that originates from x_r and terminates at y_r, the path taken by r is denoted as p_r. Within each path p_r, the entity flows through consecutive sequences of nodes. For every pair of consecutive nodes $i, j \in p_r$, the segment $s_{i,j}$ connecting the pair is associated with the known distance $d_{i,j}$ and an estimated speed $c_{i,j}$. The time taken $t_{i,j}$ for each segment $s_{i,j}$ could then be estimated using the distance $d_{i,j}$ and speed $c_{i,j}$. Summation of the estimated time in each segment gives the expected time t_r needed for r to travel from x_r to y_r.

The distribution of time for each segment $s_{i,j}$ is given by the following Gaussian distribution,

$$t_{i,j} \sim \mathcal{N}\left(\frac{d_{i,j}}{c_{i,j}}, d_{i,j}\sigma^2\right)$$

The distribution of time t_r of r is given by the following linear Gaussian distribution,

$$t_r = \sum_{(i,j)\in p_r} t_{i,j}$$

$$t_r \sim \mathcal{N}\left(\sum_{(i,j)\in p_r} \frac{d_{i,j}}{c_{i,j}}, d_r\sigma^2\right)$$

To estimate the variance σ^2,

$$\sigma^2 = \frac{\sum_{r\in R}\left(\hat{t}_r - \sum_{(i,j)\in p_r}\frac{d_{i,j}}{c_{i,j}}\right)^2}{\sum_{r\in R} d_r}$$

To estimate the speed $c_{i,j}$ of every segment $s_{i,j}$ using the observed time $\hat{t}_r$ of each record r, we **maximize** the log likelihood $\mathcal{L}_r$ from each r. The log likelihood $\mathcal{L}_r$ as contributed by r is given by,

$$\begin{aligned}\mathcal{L}_r &= \log\left(\frac{1}{\sqrt{2\pi d_r\sigma^2}}\right) - \frac{\left(\hat{t}_r - \sum_{(i,j)\in p_r}\frac{d_{i,j}}{c_{i,j}}\right)^2}{2d_r\sigma^2} \\ &= -\frac{1}{2}\log\left(d_r\sigma^2\right) - \frac{\left(\hat{t}_r - \sum_{(i,j)\in p_r}\frac{d_{i,j}}{c_{i,j}}\right)^2}{2d_r\sigma^2}\end{aligned}$$

We add a log barrier penalty to prevent negative speeds. A larger $c_{i,j}$ increases the $\mathcal{L}_r^*$ term as given by the addition, which is encouraged since we are trying to maximize the log likelihood.

$$\mathcal{L}_r^* = \mathcal{L}_r + \tau \sum_{(i,j)\in p_r} \log c_{i,j} \tag{1}$$

where τ is the strength of the penalty. By taking partial derivative with respect to $c_{p,q}$,

$$\frac{\partial \mathcal{L}_r^*}{\partial c_{p,q}} = -\frac{\left(\hat{t}_r - \sum_{(i,j)\in p_r}\frac{d_{i,j}}{c_{i,j}}\right)}{d_r\sigma^2}\cdot\frac{d_{p,q}}{c_{p,q}^2} + \frac{\tau}{c_{p,q}} \tag{2}$$

The partial derivative allows us to perform Stochastic Gradient Descent (SGD) on parameters $c_{i,j}$ as follows,

$$c_{i,j} \leftarrow c_{i,j} + \eta\frac{\partial \mathcal{L}_r^*}{\partial c_{i,j}}$$

There are several interesting properties with the partial derivative in Equation 2. The variance in the denominator shows that the more uncertain we are, the lesser the gradient is, hence less changes to $c_{i,j}$. The smaller $c_{i,j}$ is, the second component in Equation 2 will compensate by adding positive value to prevent $c_{i,j}$ from entering the negative region.

D. Smoothed Edge-based Model

To avoid overfitting the model parameters to the observed data set, we add additional constraints to Equation 1 that minimize the difference between the speeds of consecutive segments in a path. This constraint is based on the assumption that consecutive segments have related speeds. In the equation that follows, a larger difference between the speeds of two consecutive segments lowers the log likelihood as given by the subtraction, which is discouraged.

$$\mathcal{L}_r^{**} = \mathcal{L}_r^* - \frac{\psi}{2}\sum_{(i,j)\in p_r}\left(c_{i,j} - c_{j,k}\right)^2$$

where $c_{j,k}$ is speed of $s_{j,k}$ that comes after $s_{i,j}$.

Estimation for the variance σ^2 remains the same while estimation of $c_{i,j}$ is slightly modified,

$$\frac{\partial \mathcal{L}_r^{**}}{\partial c_{i,j}} = \frac{\partial \mathcal{L}_r^*}{\partial c_{i,j}} - \psi(c_{i,j} - c_{j,k})$$

$$c_{i,j} \leftarrow c_{i,j} + \eta\frac{\partial \mathcal{L}_r^{**}}{\partial c_{i,j}}$$

In Section V, we would evaluate which of these models is a better choice in terms of fitting to records that are not observed during the estimation (training) phase.

IV. Localization of Network Anomalies

The models as described in the previous section would allow us to determine whether a record $r \in R$ is anomalous by comparing the difference of the observed time and the expected time $\hat{t}_r - t_r$ with the standard deviation $\sqrt{d_r\sigma^2}$. We use the following ratio α_r to measure the degree of deviation.

$$\alpha_r = \frac{\hat{t}_r - t_r}{\sigma\sqrt{d_r}} \tag{3}$$

Given any record $r \in R$, $\alpha_r > 1$ indicates that the time taken is longer than expected while $\alpha_r < 1$ indicates that the time taken is shorter than expected. In most distributed systems, the records of interest for further investigation would be those with $\alpha_r > \delta$, where δ is a cut-off value to determine whether r has a significantly larger observed time than expected. We would be able to obtain a reduced set of records $R_{\alpha>\delta}$ such that $r \in R_{\alpha>\delta}$ has a ratio $\alpha_r > \delta$. Using the reduced set $R_{\alpha>\delta}$ instead of the full set R, we could save computational costs by focusing on a smaller set of records for finding the location of anomalies in the distributed system.

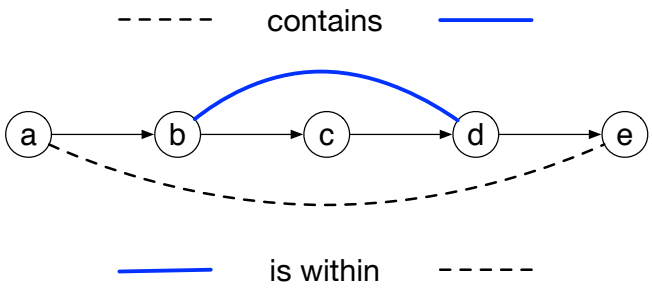


Fig. 3. Example for the *contains* and *within* definitions

But a high ratio α_r for record $r \in R_{\alpha>\delta}$ could be an isolated incident that does not have any significant impact on the distributed system. The ratio α_r also does not reveal the specific segment $s_{i,j}$ in the path p_r of r that causes the longer observed time $\hat{t}_r$. To address these issues, we propose an algorithm that serves two purposes:

1) Measuring how many other records $r' \in R_{\alpha>\delta} \setminus r$ are related to r in order to determine the significance of the network congestion in the path p_r taken by r.
2) Locating the segment $s_{i,j} \in p_r$ that is most likely to contribute to the high α_r ratio of r.

We first define the "relatedness" of two records r and r' in more precise terms using "contains" and "within".

Definition 1: r' *contains* r if all of the following conditions are satisfied:

1) The path $p_{r'}$ connecting the origin $x_{r'}$ and destination $y_{r'}$ of r', passes through all the nodes of path p_r that connects the origin x_r and destination y_r of r. Figure 3 shows an example of this condition. The path connecting node a to node e contains the path which connects node b to the node d.
2) The time $t(x_{r'})$ when r' starts at origin $x_{r'}$, is earlier than the time $t(x_r)$ when r starts at origin x_r. i.e. $t(x_{r'}) < t(x_r)$.
3) The time $t(y_{r'})$ when r' ends at destination $y_{r'}$, is later than the time $t(x_r)$ when r ends at destination y_r. i.e. $t(y_{r'}) > t(y_r)$.

Definition 2: r is *within* r' if and only if r' contains r.

Based on the two definitions, the algorithm for localizing the anomalies in the network proceeds as follows:

1) Obtain the set of records $R_{\alpha>\delta}$ such that $\forall r \in R_{\alpha>\delta}$ has ratio $\alpha_r > \delta$. This gives us the set of records $R_{\alpha>\delta}$ with observed travel time that is significantly larger than the expected travel time.
2) For each $r \in R_{\alpha>\delta}$, obtain the set of records R_r, where

$$R_r := \{r' \in R_{\alpha>\delta} | r' \text{ contains } r \wedge r' \neq r\}$$

That is, R_r is the set of records that contains r. The value of $|R_r|$ has a positive correlation on the importance of path p_r to other records and traffic.
3) Then by sorting the set of records $R_{\alpha>\delta}$ in descending order of $|R_r|, \forall r \in R_{\alpha>\delta}$, and examining the segments $s_{i,j}$ of path p_r, we would be able to locate the segments $s_{i,j} \in p_r$ with severe network congestion between the times of $t(x_r)$ and $t(y_r)$.
4) For any given $r' \in R_{\alpha>\delta}$, we would also be able to locate the congested segments of path $p_{r'}$ by using the path p_r of record r, where r is within r' and $R_r = \emptyset$.

V. Experiments

We first apply the models from Section III on the data set of a distributed system to test the generalization performance. The model that has the best generalization performance would have the lowest prediction error for unobserved data. We then select the best model to use for detection and localization of network anomalies in the distributed system by applying the algorithm as described in Section IV. To evaluate the reliability of the detected and localized anomalies, we examine a set of tweets from Twitter of the same period to check whether users of the distributed system expressed their frustrations by tweeting (complaining) on Twitter.

A. Data Description

We evaluate our models on a data set that contains the passengers' travel records on the Public Transportation System (PTS) of an urbanized city. The PTS consists of the railway system and the public bus system.

Each passenger carries a payment card containing a RFID chip. The RFID chip allows the companies to identify the passenger and charge the transportation fare to their account. The passengers boarding and alighting geo-locations are recorded for each travel trip in order to charge the appropriate amount based on the distance traveled. The time boarded and time alighted are also recorded, which makes it possible to calculate the time taken for the travel.

Being a densely populated city, majority of the city's population commute via the PTS, instead of driving in privately owned vehicles. The data set thus represents an almost complete usage of a real and large-scale distributed system, which would allow us to utilize it for verifying our proposed models and algorithms.

We mentioned in Section I that we wanted to detect non-critical anomalies because it is less noticeable than critical anomalies. The transportation network of bus routes would contain many of these non-critical anomalies because traffic in congested road segments can be diverted to other roads. Each record r contains the following information of a passenger's journey:

1) Bus stop ID of boarding, x_r.
2) Bus stop ID of alighting, y_r.
3) Date and time of boarding, $t(x_r)$.
4) Date and time of alighting, $t(y_r)$.
5) Distance traveled between boarding and alighting, d_r.
6) Bus service: The bus service is a number, which represents the unique route taken by the bus. Many different buses operate using the same service number so that different buses can pick up or alight passengers at various bus stops with regular intervals. Using the bus service number, the bus stop ID of

boarding and alighting, we are able to obtain the path p_r traveled by the passenger of this record r.

The bus routes remain relatively static but may change due to road maintenance and repairs. To ensure that we always have the correct bus routes for each specific bus service, we perform a simple bus route inference step in the next section.

1) Bus Route Inference: Figure 4 shows an example of three records for the *same* bus service number. We have the time and location of the passenger boarding and alighting from the bus for each of their journeys. The nodes in Figure 4 represent the bus stops while the number in the connecting edges represent the distance between two bus stops. By using different records of the same bus service, we are able to construct the route that the bus service takes, as shown in Figure 4. For example, from Figure 4, we can infer that b should be between a and c because b is nearer to c compared to a, and c is nearer to b than d so d should come after c. Then

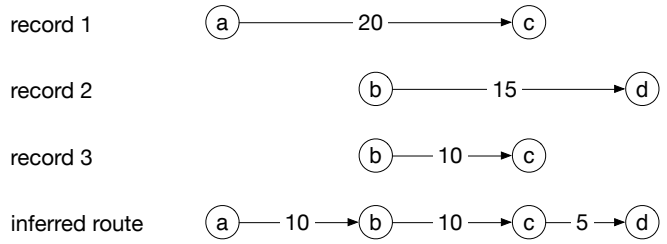


Fig. 4. Examples of the trip records in the data set for a specific bus service

by combining the inferred routes of various bus services, we would be able to obtain a transportation network as shown in Figure 1.

2) Statistics: We perform our initial analysis on three days of data, December 8th 2011, December 15th 2011 and December 22nd 2011. These three days are spaced one week apart and falls on Thursday of the week, which is a typical working day. The reason for choosing these specific days is because of the occurrence of an event (external to the bus system) on December 15th 2011 which affected the transportation system of the city. The choice of December 8th and December 22nd is to show that the anomalies present on December 15th, is absent one week before and one week after. We perform the bus route inference on the records of these three days and obtained the respective transportation network for each day. Statistics of the transportation network is shown in Table I. While most of the transportation network remains fairly static, there are minor fluctuations in the size of the network due to several reasons: 1) The city performs road maintenance frequently, resulting in temporary changes in routes of some bus services. 2) The bus company temporarily provides additional bus services to cater for special events. 3) We only retain the data of bus routes that we are able to infer without any errors.

TABLE I. STATISTICS OF TRANSPORTATION NETWORK

	Dec 8th	Dec 15th	Dec 22nd
No. of records	3,137,469	3,176,830	3,162,985
No. of inferred bus routes	288	285	287
No. of nodes	4497	4492	4514
No. of edges	6176	6151	6195

B. Sum-of-Squares Error Convergence

We would first like to verify that the Stochastic Gradient Descent (SGD) algorithm converges for our proposed model in Section III. The SGD algorithm is supposed to reduce the error between the expected time given by the models and observed time as reflected in the records $r \in R$. The error for all records $r \in R$ is given by the Sum-of-Squares Error (SSE) as shown in Equation 4. A lower SSE value suggests a better fit of the model to the observed data.

$$\text{Sum-of-Squares Error} = \sum_{r \in R} (t_r - \hat{t}_r)^2 \tag{4}$$

Figure 5 shows the convergence of SSE with respect to the number of iterations we ran for the SGD algorithm. As observed in Figure 5, the SGD algorithm decreases the SSE over multiple iterations with the Edge-based model having a lower SSE than Smoothed Edge-based model. This is due to the additional smoothing constraints we have imposed on the Smoothed Edge-based model. The change in SSE decreases as number of iterations increase, which suggests that the SGD algorithm converges for our two proposed models, resulting in marginal improvement in the SSE with increasing number of iterations.

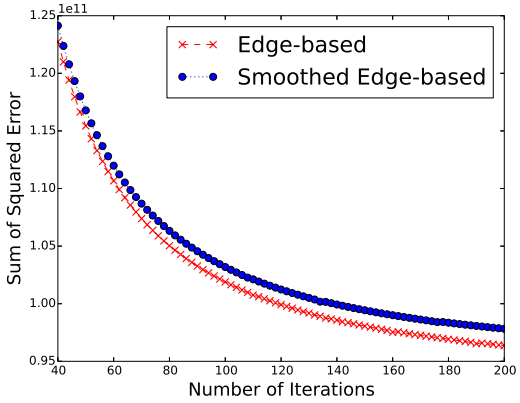


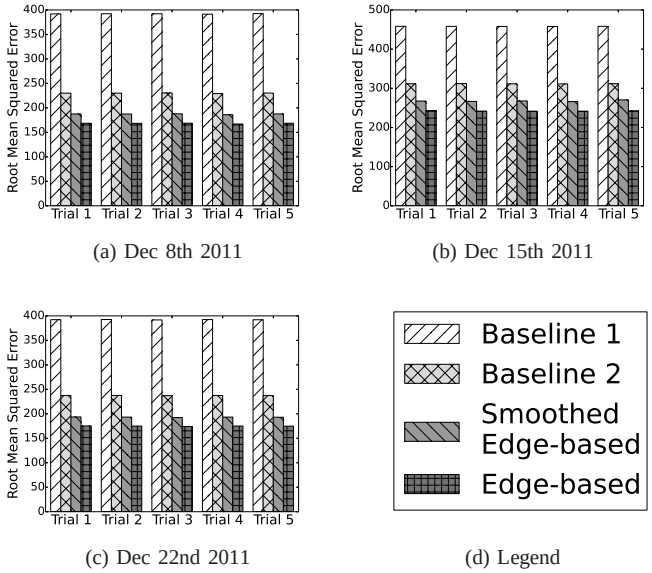
Fig. 5. Stochastic Gradient Descent Convergence rate for Sum-of-Squares Error

C. K-Fold Cross Validation

We evaluate the two models proposed in Section III on how well they generalize to data that is not seen during the training (inference and estimation) phase. We perform a K-fold cross validation evaluation on each day of data. The procedure for K-fold cross validation first uniformly and randomly divide the data set into K number of smaller data sets called folds. We choose $K-1$ of these folds to represent the observed data R_{train} for training, while the remaining fold is used as the unseen data R_{test} for testing the goodness-of-fit of the estimated parameters obtained after training. The goodness-of-fit we used in this part of the experiments is the Root-Mean-Squared-Error (RMSE).

$$\text{RMSE of data} = \sqrt{\frac{\sum_{r \in R_{data}} (t_r - \hat{t}_r)^2}{|R_{data}|}}$$

The experiment is repeated for K number of trials, by cycling the test set through each of the K folds. We performed the K-fold cross validation independently on three days of data and obtain the results as shown in Figures 6 and 7.

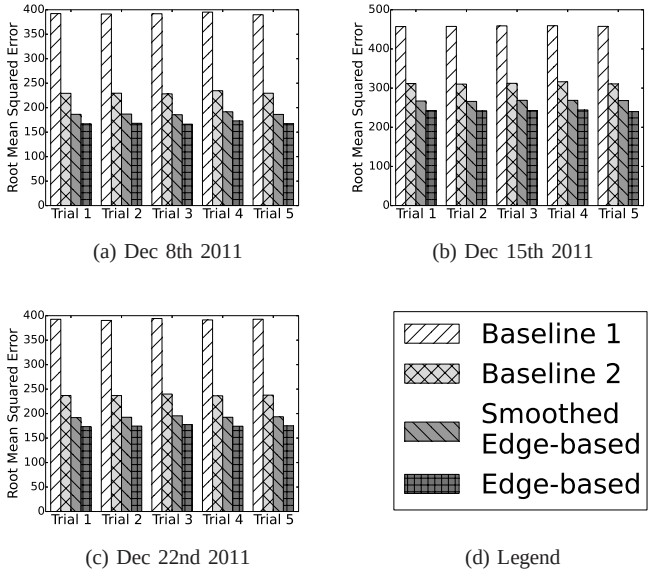


(a) Dec 8th 2011 (b) Dec 15th 2011

(c) Dec 22nd 2011 (d) Legend

Fig. 6. Training Set: 5 Fold Cross Validation

Figures 6a to 6c show the RMSE of the data set R_{train} used for estimating the parameters of the models in Section III. Using the estimated parameters, we proceed to obtain the expected time taken for the records $r \in R_{test}$, and obtain the respective RMSE of the test data, shown in Figures 7a to 7c.



(a) Dec 8th 2011 (b) Dec 15th 2011

(c) Dec 22nd 2011 (d) Legend

Fig. 7. Testing Set: 5 Fold Cross Validation

In all of these Figures, with reference to the legend shown in Figures 6d and 7d, we illustrate the RMSE given by various models for each trial. The results consistently show that the Edge-based model has the best performance, follow by Smoothed Edge-based, then Baseline 2 and finally Baseline 1. This suggests that two of our proposed models outperform the baselines and the speeds we have inferred for the segments are accurate in estimating the expected times of the journeys. Given that Edge-based model outperforms Smoothed Edge-based model, we will use the results of the Edge-based model for the rest of our analysis and ignore the smoothing constraints of the Smoothed Edge-based model.

We also observed that the RMSE of all four models in Figures 6b and 7b for Dec 15th 2011 is higher than the RMSE shown the Figures representing Dec 8th 2011 (Figures 6a, 7a) and Dec 22nd 2011 (Figures 6c and 7c). This is an indication that a significant anomaly is present on the day of Dec 15th 2011, which causes the RMSE goodness-of-fit for our proposed model to differ from other normal days. In the next section, we will give a case study of identifying the anomalies using the algorithm we proposed in Section IV.

D. Evaluation of Anomalies using Twitter

The main objective which motivates our research is to detect and localize the network anomalies in distributed systems. Given that in comparison to other models, the Edge-based model gives a more accurate estimation of how long a journey should take, we use the estimation of Edge-based model as inputs to the algorithm described in Section IV to detect and localize the network anomalies.

We mentioned in Section I that we focus on non-critical anomalies which are elusive and difficult to detect in real usage of distributed systems. As a result of this elusive property of non-critical anomalies, our detection and localization algorithm does not have sufficient labeled data to evaluate its accuracy in a quantitative manner. In fact, there would be no research problem to address if such anomalies are easily observed for us to perform our experiments.

Fortunately, since we use the Public Transportation System (PTS) of an urbanized city, we are able to compare the detected anomalies with the tweets of the residents who commute in the city. We use the Twitter data set that was formerly used to analyze political sentiments in Hoang and Lim [24], to perform a qualitative comparison of our detected anomalies with the tweets that comment on the traffic conditions. We use the Twitter data set $\mathcal{T}$, and the PTS records R between the periods of November 1st 2011 and January 31st 2012 for comparison.

We described in Section IV that the detection algorithm first obtains the set of records $R_{\alpha>\delta}$, where each record $r \in R_{\alpha>\delta}$ has a larger observed travel time than the expected travel time, i.e. $\alpha_r > \delta$. We set δ, the cut-off value of determining whether a record deviates significantly as the top 1% of the ratio values α_r, for all $r \in R$. The records $R_{\alpha>\delta}$ are then sorted in descending order of $|R_r|$, so that records $r \in R_{\alpha>\delta}$ with the most important path p_r are ranked first.

1) Evaluation of $|R_r|$ vs α_r: We evaluate the use of $|R_r|$ vs α_r as a ranking metric by comparing with the number of tweets in $\mathcal{T}$ that mention the keywords (ignoring case) $\mathcal{K}$,

$$\mathcal{K} := \{ \text{“traffic”} \wedge (\text{“jam”} \vee \text{“jams”}) \}.$$

Table II provides the statistics of the Twitter data $\mathcal{T}$ and the tweets $\mathcal{T}_\mathcal{K}$ with the keywords K for each month. Table II also show the number of tweets $\mathcal{T}_\mathcal{M}$ containing keywords $\mathcal{M}$, which we will elaborate in later part of this section.

TABLE II. STATISTICS OF TWITTER DATA

Period	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{T}_\mathcal{K}\rvert$	$\lvert\mathcal{T}_\mathcal{M}\rvert$
Nov 2011	17,421,755	1,330	36
Dec 2011	19,216,767	1,829	8,247
Jan 2012	19,565,979	1,891	847

Figure 8a shows the histogram of tweets $\mathcal{T}_\mathcal{K}$ containing keywords $\mathcal{K}$. The width of the bins in Figure 8a is chosen to represent the duration of one day, so that the frequency (y-axis) shown in the histogram represents the number of tweets that contain keywords $\mathcal{K}$ for the specific day (x-axis).

We used the Edge-based model of Section III and the anomalies localization algorithm of Section IV on the records of each day between the periods of November 1st 2011 to January 1st 2012. For each day, we derive α_r of every record r in that day using Equation 3, then we obtain the set of records $R_{\alpha>\delta}$, and derive the set of $R_r, \forall r \in R_{\alpha>\delta}$. Next, we obtain the mean and median values of $|R_r|, \alpha_r, \forall r \in R_{\alpha>\delta}$ of each day and derive the plots shown in Figures 8b and 8c.

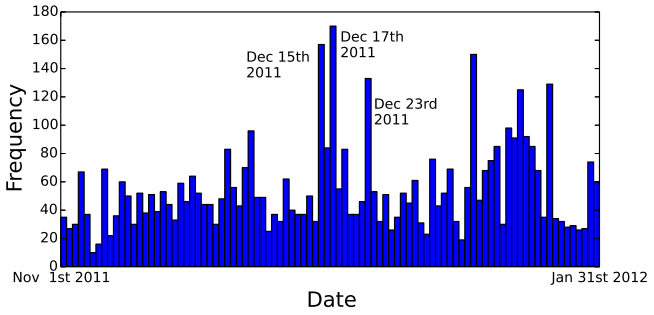
The histogram in Figure 8a appear to correlate better with the plots in Figure 8b compared to the plots in Figure 8c, especially for the two dates marked in Figure 8b. We calculated the Pearson correlation of the frequencies in Figure 8a with the mean values in Figure 8b and obtain the value of 0.64. On the other hand, the Pearson correlation between the frequencies of Figure 8a and mean values of Figure 8c is only 0.20. This confirms that our proposed metric of $|R_r|$ is better than α_r for ranking the anomalies in the set of records $R_{\alpha>\delta}$.

2) Case studies of the location for the detected anomalies: In this final section of our evaluation on the detected anomalies, we describe the details of evaluating the location and time of our detected anomalies through qualitative comparison with the contents of the tweets. From Figure 8a, we have marked December 15th and 17th of 2011 which shows significant spike in the number of tweets mentioning "traffic jam(s)". Based on prior knowledge, we are aware of the existence of an external event that causes the congestion on these two days. The external event is the breakdown of the railway system that caused temporary disruption to the railway services.

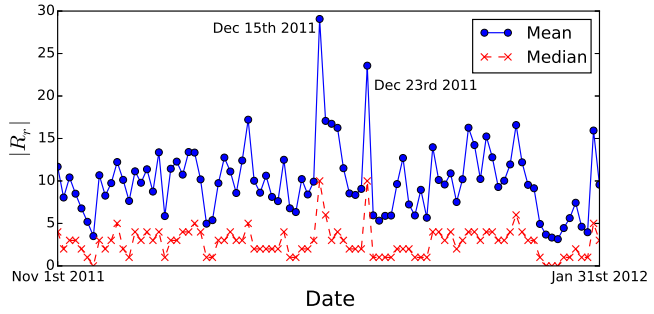
The city is supported by two main transportation systems, the railway system (MRT[2]) and the public bus system. When the railway system breaks down, the remaining transportation options in the city has to bear the load of transporting the passengers. These other transportation options which include taxis and privately owned motorized vehicles share the well-connected road network with the public bus system. The sharing of physical road space causes delay in the speeds of bus during the breakdown of the railway system.

In order to determine when the railway system breaks down, we search the set of tweets $\mathcal{T}$ between the periods of

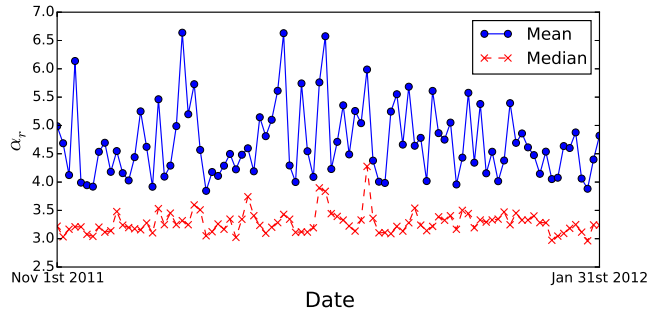
[2]MRT is the acronym of the railway system and it stands for Mass Rapid Transit.



(a) Histograms of $\mathcal{T}_\mathcal{K}$ that mention "traffic jam(s)"



(b) $|R_r|$: Mean values has 0.64 Pearson correlation with frequencies



(c) α_r: Mean values has 0.20 Pearson correlation with frequencies

Fig. 8. Comparision of Anomaly Ranking Metric between $|R_r|$ and α_r for the period between Nov 1st 2011 and Jan 1st 2012

November 1st 2011 to January 31st 2012 for the keywords (ignoring case) $\mathcal{M}$,

$$\mathcal{M} := \{\ \text{mrt} \wedge (\text{break down} \vee \text{breaks down} \vee \text{breakdown})\ \}$$

and obtain the subset of tweets $\mathcal{T}_\mathcal{M}$. Table II shows statistics of $\mathcal{T}_\mathcal{M}$ for each month. We use Figures 9a and 9b to show the number of tweets generated for each day. Figure 9a shows that there are two days, May 15th 2011 and May 17th 2011 with many tweets containing the keywords $\mathcal{K}$. These two days also correlate with the frequencies as seen in Figure 8a.

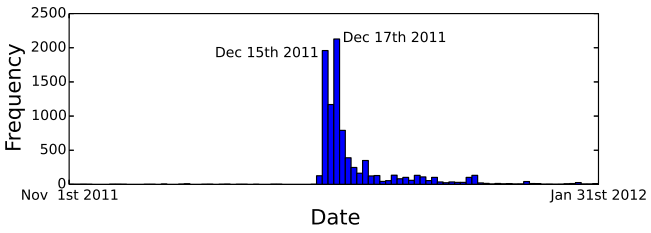
Table III shows examples of tweets that comment on the traffic conditions whenever the railway system breaks down, with Tweets #3 and #5 of Table III showing passengers' feedback that the breakdown of railway system also affects the bus travel times. At the time of this writing, all the URLs as shown, link to the publicly available tweets on Twitter website.

TABLE III. EXAMPLES OF TWEETS FROM $\mathcal{T}_{\mathcal{M}}$, COMMENTING ON THE TRAFFIC AFTER RAILWAY SYSTEM (MRT) BREAKS DOWN

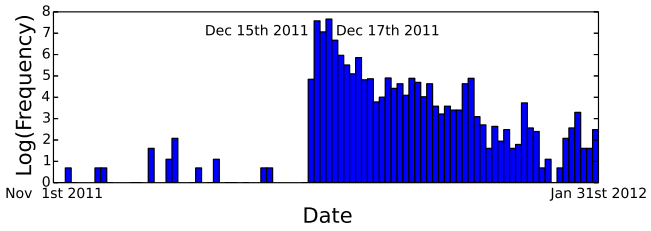
#	URL	Content	Time GMT+8
1	http://twitter.com/tickingbombs/status/131164[illegible]	[illegible]n at **DS**. Whats the fare hike for again?	2011-11-01 7:25:08
2	http://twitter.com/mrbrown/status/14675664329[illegible]	[illegible]down pic #smrt RT @hai_ren: The crowd at **PL**. #SMRTruinslives http://t.co/RGQwSp3s	2011-12-14 8:01:25
3	http://twitter.com/mac_beno_l/status/1473044[illegible]	[illegible]reak down, causing MAJOR traffic jam along **OR**. Feeling very sick :(	2011-12-15 20:18:15
4	http://twitter.com/true_joygiver/status/1473199[illegible]	[illegible] the serious MRT breakdown! Stuck in **SS** 4 hours & forced to miss a dinner gathering! =(	2011-12-15 21:19:48
5	http://twitter.com/ONGLSD/status/1473231[illegible]	The train breakdown isn't just affecting SMRT's traffic, it's causing a major killer jam from **SR** all the way to **OR**. AVOID!!	2011-12-15 21:32:31
6	http://twitter.com/afiqahmauwan/status/1473[illegible]	[illegible]down took me 2.5 hours to reach home from **DG** to **YI**.	2011-12-16 00:27:45
7	http://twitter.com/aidah/status/1478242905[illegible]	[illegible]breakdown this morning between **MB** and **NS** #fb #north-south line. Frustrating.	2011-12-17 6:43:51

TABLE IV. SUMMARY OF ANOMALIES

#	$\lvert R_r \rvert$	α_r	Origin	Destination	Distance (meters)	Time Boarded	Time Alighted	Observed (mins)	Expected (mins)
1	46	3.89	N.A.	DS	2,900	Nov 1 2011 08:04:37	Nov 1 2011 08:23:34	18.95	9.11
2	69	3.16	PL	N.A	600	Dec 14 2011 08:43:35	Dec 14 2011 08:49:57	6.37	3.11
3	483	5.79	SS	OR	1,000	Dec 15 2011 20:54:01	Dec 15 2011 21:15:04	21.05	10.22
4	181	25.27	SR	SS	1,700	Dec 15 2011 20:14:37	Dec 15 2011 21:32:02	77.42	15.79
5	161	14.12	SR	OR	2,700	Dec 15 2011 07:57:14	Dec 15 2011 21:06:36	69.37	26.01
6	0	4.16	DG	YI	31,900	Dec 15 2011 20:48:23	Dec 15 2011 23:17:43	149.33	105.38
7	149	12.98	NS	N.A.	200	Dec 17 2011 11:54:52	Dec 17 2011 12:07:45	12.88	4.51



(a) Frequency for each Day



(b) Log(Frequency) for each Day

Fig. 9. Histograms of tweets that mention "mrt breakdown" or "mrt break(s) down"

In each of these tweets, the railway stations affected by the breakdown that are mentioned by the users are highlighted in **bolded font**[3].

Although we have the complete record of passengers for the three months listed, we do not have the complete set of tweets during this period. We also do not assume that Twitter is able to capture all the passengers' feedback about the usage of the public bus system during this period of time. It is therefore not possible to obtain a quantitative correlation between the ranking of anomalies detected by our algorithm and the number of tweets mentioning the location of the congested areas. But from the content of tweets shown in Table III, we can find specific records with ratio α_r that belong to the top 1% and match the description of the tweets.

We use Table IV to show examples of records that correspond in terms of location, date and time to the traffic congestions as mentioned by Twitter users in Table III. Each record # in Table IV corresponds to the tweet # in Table III. One may notice that either the origin or destination for each record in Table IV is related to the bolded location of the tweets in Table III, while N.A. indicates that it is not related. The times of the record are also very close to the tweets and we could assume that passengers[4] tweet about their frustration during or after the journey. The only exception is #7 that has a huge difference in times because Dec 17 is a Saturday (non-working day) and passengers only commute later in the day. The records in Table IV show that the observed time taken to reach the destination is much longer than the expect time. While these few examples do not necessarily cover all cases, it certainly shows that our detected anomalies could match with the complaints in Twitter or any other social media.

VI. CONCLUSION

We began our research with the purpose of finding non-critical anomalies in the networks of distributed systems. In most distributed systems, the data that can be obtained without the use of sophisticated instruments or internal sensors is often non-informative about the internal workings of the networks. That causes difficulties in detecting anomalies and further difficulties in finding the location of anomalies within the distributed system. To overcome these difficulties, we proposed

[3]Due to requests from our data providers, we have to anonymize the location names with their initials.

[4]Disclaimer: The passengers in the records of Table IV are not the same people as the Twitter users of Table III.

the Edge-based network transmission model to infer the flow speeds of the edges within the networks of distributed systems. With the model, we are able to derive the expected time necessary for entity to complete its flow. Using the records of entities flow with observed time that is significantly longer than expected, we apply our proposed localization algorithm to measure the relationship of each record to all other records with large deviations. The number of related records allows us to determine how important each record is to the traffic conditions of the network. By finding records that are highly related to other records and with the shortest travel path in the network, we are able to determine the location of the anomalies within the distributed system.

One major assumption that we had made in this work is that the knowledge of the exact path taken by the entity flow is known or can be easily inferred. While this is true for transportation systems, it may not be true for other kinds of distributed systems. One way to overcome this is to have an intermediate step to infer the path using a Markovian model.

We make some final remarks about other possible improvements. The current models have not considered the notion of peak and off-peak usage of the traffic patterns in networks of distributed systems. During peak usage, the load is generally higher and could result in longer observed travel times. This issue can be easily addressed by using a mixture of Gaussian distributions to model the edge speeds. Another possible improvement is the algorithm for counting the "contains" and "within" relationships between records. Using concepts of transitivity, e.g. if record a contains record b, and record b contains record c, then record a contains record c, one could save on the computation costs significantly.

ACKNOWLEDGMENT

We would like to thank the Land Transport Authority (LTA) of Singapore for sharing with us the EZLink dataset.

REFERENCES

[1] F. Wu, B. Huberman, L. Adamic, and J. Tyler, "Information flow in social groups," *Physica A: Statistical Mechanics and its Applications*, vol. 337, no. 1-2, pp. 327–335, 2004.

[2] A. Kind, M. P. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Trans. on Netw. and Serv. Manag.*, vol. 6, no. 2, pp. 110–121, Jun. 2009.

[3] H. Sengar, X. Wang, H. Wang, D. Wijesekera, and S. Jajodia, "Online detection of network traffic anomalies using behavioral distance," in *Quality of Service, 2009. IWQoS. 17th International Workshop on*, July 2009, pp. 1–9.

[4] J. Fadlil, H.-K. Pao, and Y.-J. Lee, "Anomaly detection on its data via view association," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ser. ODD '13. New York, NY, USA: ACM, 2013, pp. 22–30.

[5] N. J. Yuan, Y. Wang, F. Zhang, X. Xie, and G. Sun, "Reconstructing individual mobility from smart card transactions: A space alignment approach," *2013 IEEE 13th International Conference on Data Mining*, vol. 0, pp. 877–886, 2013.

[6] A. Agovic, A. Banerjee, A. Ganguly, and V. Protopopescu, "Anomaly detection using manifold embedding and its applications in transportation corridors," *Intell. Data Anal.*, vol. 13, no. 3, pp. 435–455, Aug. 2009.

[7] J. Mihm, C. H. Loch, D. Wilkinson, and B. A. Huberman, "Hierarchical structure and search in complex organizations," *Manage. Sci.*, vol. 56, no. 5, pp. 831–848, May 2010.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[9] C. De Stefano, C. Sansone, and M. Vento, "To reject or not to reject: that is the question-an answer in case of neural classifiers," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 30, no. 1, pp. 84–94, Feb 2000.

[10] D. Barbara, N. Wu, and S. Jajodia, "Detecting novel network intrusions using bayes estimators," in *Proc. SIAM Intl. Conf. Data Mining*, 2001.

[11] M. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *Data Mining and Knowledge Discovery*, vol. 12, no. 2-3, pp. 203–228, 2006.

[12] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recogn. Lett.*, vol. 24, no. 9-10, pp. 1641–1650, Jun. 2003.

[13] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 255–262.

[14] C. Chow, "Parzen-window network intrusion detectors," in *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4 - Volume 4*, ser. ICPR '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 40 385–.

[15] S. Ando, "Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection," ser. ICDM '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 13–22.

[16] Q. Fu, J.-G. Lou, Y. Wang, and J. Li, "Execution anomaly detection in distributed systems through unstructured log analysis," ser. ICDM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 149–158.

[17] M. Thottan and C. Ji, "Anomaly detection in ip networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2191–2204, Aug 2003.

[18] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," ser. SIGSPATIAL'13. New York, NY, USA: ACM, 2013, pp. 334–343.

[19] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 1010–1018.

[20] Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou, "A taxi driving fraud detection system," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, Dec 2011, pp. 181–190.

[21] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "ibat: Detecting anomalous taxi trajectories from gps traces," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, ser. UbiComp '11. New York, NY, USA: ACM, 2011, pp. 99–108.

[22] J. Zhang, "Smarter outlier detection and deeper understanding of large-scale taxi trip records: A case study of nyc," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, ser. UrbComp '12. New York, NY, USA: ACM, 2012, pp. 157–162.

[23] S. Chawla, Y. Zheng, and J. Hu, "Inferring the root cause in road traffic anomalies," ser. ICDM '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 141–150.

[24] T.-A. Hoang and E.-P. Lim, "Virality and susceptibility in information diffusions," 2012.