Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 382132, 12 pages
http://dx.doi.org/10.1155/2013/382132

Research Article

A Case Study of Sensor Data Collection and Analysis in Smart City: Provenance in Smart Food Supply Chain

Qiannan Zhang,[1] Tian Huang,[1] Yongxin Zhu,[1] and Meikang Qiu[2]

[1] *School of Microelectronics, Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *Department of Computer Engineering, San Jose State University, San Jose, CA 95152, USA*

Correspondence should be addressed to Yongxin Zhu; zhuyongxin@sjtu.edu.cn

Received 6 July 2013; Accepted 10 September 2013

Academic Editor: Yu Gu

Accelerated growth of urban population in the world put incremental stresses on metropolitan cities. Smart city centric strategies are expected to comprise solutions to sustainable environment and urban life. Acting as an indispensable role in smart city, IoT (Internet of Things) connects the executive ability of the physical world and the intelligence of the computational world, aiming to enlarge the capabilities of things in real city and strengthen the practicality of functions in cyber world. One of the important application areas of IoT in cities is food industry. Municipality governors are withstanding all kinds of food safety issues and enduring the hardest time ever due to the lack of sufficient guidance and supervision. IoT systems help to monitor, analyze, and manage the real food industry in cities. In this paper, a smart sensor data collection strategy for IoT is proposed, which would improve the efficiency and accuracy of provenance with the minimized size of data set at the same time. We then present algorithms of tracing contamination source and back tracking potential infected food in the markets. Our strategy and algorithms are evaluated with a comprehensive evaluation case of this IoT system, which shows that this system performs well even with big data as well.

1. Introduction

As urban population and corresponding needs of living necessities keep expanding in modern cities, much higher requirements are set for municipality governors to manage all aspects in urban living. The performance of cities currently depends on not only the city's endowment of hardware infrastructure, but also on the availability and quality of knowledge communication and social infrastructure [1]. Smart city mainly focuses on applying the next-generation information technology to all fields of life, embedding sensors to all physical objects in every corner of the world [2], and forming the Internet of Things (IoT) via the Internet. Then, we can integrate the internet of things through super computers and cloud computing [3, 4].

IoT refers to uniquely identifiable objects and their virtual representations in an Internet-like structure. It gives the researchers on smart cities a wider platform and much more possibilities [5] and is expected to substantially support sustainable development of future smart cities [6]. The aim of IoT is to create a distributed network of intelligent sensor nodes which can measure many parameters to manage the city more efficiently [7]. The term, Internet of Things, was firstly used by Kevin Ashton in 2011 [8]. With the rapid development of Radio-frequency Identification (RFID), people and objects in the physical world are equipped with all kinds of sensors and radio tags to authenticate their identity and status [9]. The introduction of IoT makes people's daily life easier, safer, and more interesting. Business may no longer run out of stock or generate waste products, as involved parties would know which products are required and consumed. Traffic conditions can be achieved directly via cell-phones or GPS, so that we can safely keep away from traffic jams or even accidents. All kinds of data are collected and analyzed to entertain people on the internet, for example, constellation interpretation or social hotspots.

Provenance, which was originally used in works of arts, refers to the chronology of the ownership or location of a historical object [10]. With the rapid development of IoT and cloud computing, provenance has been studied in plenty of

areas beyond arts, among which tracing the provenance of an object or entity is a major aspect. The main purpose of tracing the provenance is to provide contextual and circumstantial evidence for its original production or discovery, by establishing, as far as practicable, its later history, especially the sequences of its formal ownership, custody, and places of storage.

Industrialization and rapid growth of human demands have made food supply chain in modern cities move beyond regional and include global participation in importing and exporting. According to the U.S. Census, the imported proportion of U.S. food consumption has grown from 7.9% to 9.6% between 1997 and 2005, roughly a 22% gain [11]. The momentum of changes grows even faster nowadays. The scale and heterogeneity of food supply chain make the capacity of existing regulations and approaches limited. At this point, IoT is a must for us as a platform to monitor and manage food supply chain.

In this paper, we discuss a case in tracing provenance of food supply chain, which is a feasible application of IoT in smart cities. Our major contributions are as follows.

We propose Self-adaptive Dynamic Partition Sampling (SDPS) Strategy to collect data from sensors, which would mitigate the workload with minor loss of tracing accuracy. Without loss of performance, our strategy needs only a small portion of end markets' samples from huge volume of raw materials and products along all levels in the food supply chain form. This would be an interesting discovery as smart sampling is not explored intensively to manage data in IoT systems for food supply chains though data collection and modeling have been studied in IoT domain before.

As a case of SDPS applications, we introduce tracing and backtracking algorithms to achieve provenance reasoning in food supply chain. These methods can pinpoint the contamination source in the network and identify the potential problematic products in the markets. We are able to sample a small portion of food only in the end markets and maintain sufficient accuracy of provenance tracing over the whole IoT system at the same time.

We further visualize the data flow and contamination conditions for intuitive analysis. Some work on provenance reasoning has already been realized and well modeled [12], but no one has ever explicitly modeled contamination conditions in cities.

The rest of the paper is organized as follows. In Section 2, we will present existing related works. Section 3 briefly gives a view of the system's hierarchy. Section 4 raises the algorithms and approaches in detail. Results evaluation is presented in Section 5 and conclusion is drawn in Section 6, respectively.

2. Related Work

Provenance issues have been studied by researchers in the areas of computer systems as well as management applications in diversified information systems, which comprise part of the information technology (IT) infrastructure of smart city management. Wikipedia on smart city [1] proposed a prototype of Provenance-Aware Storage System (PASS) which could automatically collect provenance at the operating system level. Hasan et al. [13] focused on a thorough analysis of threats to provenance systems. Both of their methods are metadata models of provenance, but they have not explained how to exploit and process these data model to draw informative conclusions. In the context of food safety management, information systems are important to assist decision making in a short time frame, potentially allowing decisions to be made in real time.

In smart city management domain, food safety issues caused by contamination have not been studied adequately in terms of modeling and visualization. McMeekin et al. [14] introduced the technique of information systems used in the safety management of food supply chain. A stochastic state transition simulation model [15] as described to simulate the spread of Salmonella from multiplying through slaughter, with special emphasis for critical control points to prevent or reduce Salmonella contamination. Wein and Liu [16] developed a mathematical model of a cows-to-consumers supply chain associated with a single milk-processing facility that is the victim of a deliberate release of botulinum toxin. Qin established a quality management model for food supply chain based on game theory [17].

We have modeled and discussed about traceability in food supply chain in [18]. In the current work, algorithms are further optimized for big data and self-correction strategies are applied to make sampling and the whole scheme adaptive. Also, contamination conditions can be visualized to make the IoT system more intuitive.

Sampling strategies in IoT systems have attracted intensive studies [19–21]; however, some issues still remain unsolved; for example, how to exploit a small sampling size from huge volume of food supplies without loss of accuracy.

3. Modeling IoT System Structure for Food Supply Chains

With the growing size and demands of modern cities, the structure of food supply chain has become huge and complicated. Moreover, due to huge volume of sensors attached to items travelling along it, it is usually infeasible to collect and process sensing data from all the food in every level. Based on those concerns, to speed up provenance solutions, we only gather a small part of sensor data on the end nodes in the chain. So, how to reckon on this small portion of sensor data to figure out contamination source appears to be a pending issue in our strategy. Additional concerns also arise from this problem regarding loss of accuracy due to small sample volume and performance of tracing scheme. We will propose our heuristic approach and algorithms to tackle this problem later in this paper with additional thoughts on algorithm complexity.

3.1. Physical Structure of IoT Systems for Food Supply Chains. We have sensors at every end node in the supply chain, which provide us comparable information to determine whether the product is safe or not. With the sensor data and their physical

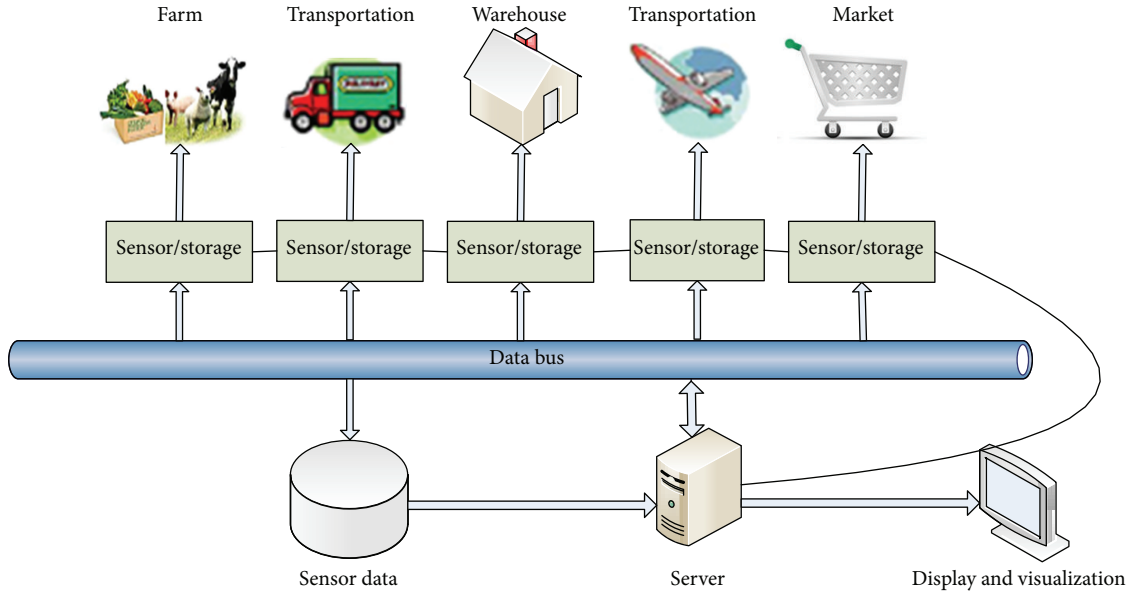

Figure 1: An illustration of IoT system's physical structure modeled for food supply chain.

connection, food supply chain forms an internet of things (IoT) network.

In reality, real-time decision making is critical in food safety issues. If contamination source is unknown for one more hour, more people will be exposed to danger. Besides, in food industry, examination is more or less material-consuming. We cannot take part of every piece of food in every stage of the chain, no matter there are problems or not, to the sensors for physical and chemical check, as it would bring food companies a great economical loss. As a result, we would like to sample food only in the end markets with a small portion.

After that, with this small part of products, we manipulate these data to get a whole picture over contamination conditions in the entire network, such as the contamination source and the other involved foods that need to be recalled.

The Physical structure of the system is shown in Figure 1.

3.2. Modeling of Food Supply Chain. Generally, food supply chain can be divided into seven stages: plantation/cultivation, slaughtering, transportation, inventory, wholesale, retailing, and customers. Although the chain is heterogeneous, we can view it as the flow through the combination and repetition of those stages based on certain rules.

Firstly, it is often impossible for us to know in advance which physical position (e.g., vehicle or warehouse) a piece of food would be in. In other words, the trend of food is almost random.

Secondly, food can access a particular location more than once, and a location can play different roles in the manufacturing of one food product. For instance, pork can be carried by the same vehicle before and after slaughtering, which will generate a circle if we view the chain as a flow.

Thirdly, not all the food in the contamination source will be infected. The percentage of infection is determined by the type of epidemic disease, temperature, density, and other objective aspects.

Finally, other locations which are not the contamination sources may also generate new contaminated food due to cross contamination. The classic Reed Frost Model has become a standard to model cross contamination conditions [22]. Based on explicit contamination discussed in this model, we introduce implicit infection to get the average infection possibility, P, in certain batch and stage location by

$$P = 1 - \left(1 - P_{\text{exp}}\right)^{\#\text{exp}} \left(1 - P_{\text{imp}}\right)^{\#\text{imp}}. \quad (1)$$

We will use # as the mark of number in this paper. In (1), P_{exp} represents the possibility that infection happens if two pieces of food touch each other directly and one of them has been explicitly infected, and #exp is the number of food products that have been explicitly infected. P_{imp} and #imp mean the same, respectively, in implicit infection cases. Food that has been implicitly infected will not infect the others but it will be counted as contaminated ones according to its physical and chemical characteristics. As implicit infection has been considered here, the model is much more realistic. Some scholars have published several extension models based on Reed Frost Model [23]; however, we only take implicit infection into consideration since (1) can describe our case better with sufficient accuracy.

Food supply chain is viewed as a Directed Acyclic Graph (DAG), in which each node stands for one location keeping or processing some batches of food for a period. DAG constructs the relationship within the internet of things based on the order and dependency among all the sensor data. The graph is acyclic since we use batch number working as a time stamp that can distinguish stages in the chain. In this way, although food may be carried by the same vehicle in more than two stages, they have different batch numbers which will be regarded as two nodes in a DAG.

(1) Input: type of foodborne disease: T
(2) Output: sample set
(3) //Training Phase:
(4) Look up contamination probability p according to T;
(5) Configuration: Topology information, #(contamination intervals), p;
(6) //Sampling Phase:
(7) Sample a small portion n;
(8) BEST = ∞;
(9) **while** $n \leq$ BEST **do**
(10) //Compute posterior probability based on Bayesian Estimation;
(11) $P(B \mid A_i) = \binom{k}{n} a_i^k (1 - a_i)^{n-k}$;
(12) $P(A_i \mid B) = \dfrac{P(B \mid A_i)P(A_i)}{\sum_{j=1}^{n} P(B \mid A_j)P(A_j)}$;
(13) **for** i = 1:#(contamination intervals) **do**
(14) **if** P.interval$_i \geq$ BEST **then**
(15) BEST = P.interval$_i$;
(16) **end if**
(17) **end for**
(18) **If** BEST $\leq$ 80% **then**
(19) Find the best sample rate according to its relationship with BEST;
(20) **If** $n \leq$ BEST **then**
(21) Sample (BEST-n) products;
(22) **else**
(23) break;
(24) **end if**
(25) **end if**
(26) **end while**

Algorithm 1: Sampling algorithm.

Records of each location and product are documented, respectively. Location records include batch numbers, the number of sampled products labeled as GOOD (uninfected) or BAD (infected) in this batch, and the IDs of polluted samples in this batch. Product records contain the information of examination result for a piece of food, the orders of batches and locations the product passed, and the pointers to these location records. The pointers serve as the connection between those two data structures.

3.3. Logic Structure of IoT System for Food Supply Chains. As shown in Figure 2, the hierarchy of this IoT system contains four layers: data collection and management layer, intelligent processing layer, graphic representation layer, and self-correction layer. Specific approaching methods and algorithms will be discussed in the following sections.

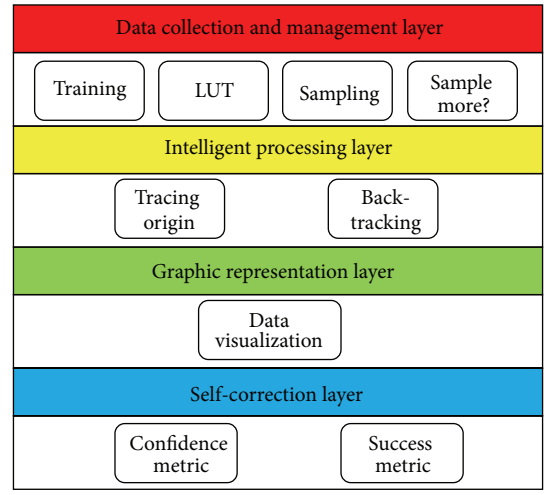

Figure 2: Logic structure of IoT system model for food supply chain.

4. Heuristic Provenance Approach and Tracing and BackTracking Algorithms

In this section, detailed approaches and algorithms to solve provenance issues in food supply chain are introduced. Firstly, we present a Self-adaptive Dynamic Partition Sampling (SDPS) Strategy to improve the efficiency and intelligence of sensor data collection and management. Then, tracing and backtracking algorithms are discussed, respectively, to catch the contamination source and dig out potential infected food products still circulating in the markets. Finally, we introduce Self-Correction Method to maintain and update the system, which would make the system adaptive and flexible to certain applications.

4.1. Self-Adaptive Dynamic Partition Sampling Strategy. As we mentioned in Section 3.1, to improve the efficiency of manipulating sensor data, Self-adaptive Dynamic Partition Sampling Strategy (SDPS) is introduced which cuts down the number of samples in a great deal. The pseudocode for sampling algorithm is shown in Algorithm 1.

4.1.1. Partition Strategy. Partition strategy makes samples more general and representative. In this case, the system divides the whole group of products into several parts according to the batches they belong to in the end markets. The sampled volume for batch n of market m is determined by

$$\#\text{sample}_{(m,n)} = \#\text{sample}_{\text{total}} \times \frac{\#\text{products}_{(m,n)}}{\sum_{m=0}^{M}\sum_{n=0}^{N}\#\text{products}_{(m,n)}}. \tag{2}$$

Here, M and N are the total number of end markets and batches in network. Subscripts (m, n) and total mean the number of samples or products in batch n, market m, and in all batches for all end markets, respectively.

4.1.2. Dynamic Strategy. Dynamic strategy based on Bayesian estimation is adopted to achieve minimal sample volume. According to the infection probability of a particular pathogen determined by medical experiments, the model can be trained to gain the distribution of total infection probability within the whole food supply network, which is the prior probability. The probability density function of the distribution is presented as a function of infection probability intervals.

On the other hand, after sampling a small part, the infectious rate of the samples, the posterior probabilities, can be obtained. If k infected products are found within n samples, under a certain contamination percentage interval with prior probability of a_i%, conditional probability is obtained by binomial distribution in the following:

$$P(B \mid A_i) = \binom{n}{k} a_i^k (1-a_i)^{n-k} = \frac{n!}{k!\,(n-k)!} a_i^k (1-a_i)^{n-k}. \tag{3}$$

Here, A_i means the event that the contamination percentage of the whole products falls into the ith interval with a prior probability of a_i% and B means the event that we find k contaminated products in n samples.

After that, Bayesian Formula, (4), is applied to combine prior probabilities with posterior probabilities and get revised probabilities, which describe the specific environment better as follows:

$$P(A_i \mid B) = \frac{P(B \mid A_i)\,P(A_i)}{\sum_{j=1}^{n} P(B \mid A_i)\,P(A_i)}. \tag{4}$$

4.1.3. Self-Adaptive Strategy. The tracing algorithm which will be discussed in the next section has some requirements for its input sampling data. If the ratio of infected products to uninfected ones is too high, the tracing algorithm performs poorly as there are not enough healthy samples to exclude the suspicions. On the contrary, if the ratio is too low, the noise introduced by sampling process may dominate the result. In these two extreme cases, more samples than other cases should be tested to improve the accuracy. Hence, under each sampling rate, there is a relationship between the best tracing algorithm accuracy and pollution proportion interval for a particular topology. Given all the relationships in a specific interval, for economical reasons, this strategy picks up the smallest sampling rate that achieves certain requirements (e.g., 90% accuracy). Then, we sample the food in end market again under that rate and update Bayesian Estimation to find if the sampling rate has met the requirements.

4.2. Tracing Algorithm. Pseudocodes of tracing algorithm are shown in Algorithm 2. After sampling and sensing, we add up the number of infected and uninfected food products passing every location and batch. They are stored in two variables: *GOOD* and *BAD* for each place.

Supposing that the samples properly reveal the condition of the whole products set, the criterion to find the suspect sources is set as $GOOD < \varepsilon$ and $BAD > 0$. It is feasible because the contamination source would be the primary spot generating polluted food and the number of uninfected samples is limited there. ε, regarded as the error factor, is a small integer which enables the algorithm to remain valid when not all the food passing the source is infected or there is some disturbance caused by nonideal problems (e.g., imperfect sampling). The specific ε value is decided by the samples' number and infection probability of pollutant source, which can be roughly represented as follows:

$$\varepsilon = \frac{\#\text{samples} \times \text{pollution\_probability}}{\#\text{batches}}. \tag{5}$$

This criterion is not strict enough to pinpoint only one contamination source as the result. So, extra work should be applied to eliminate these confusion suspects. First of all, to improve the speed of the algorithm, suspects with small *BAD* value will be excluded. Then, the system will generate a *Suspect Tree* composed of the suspected locations and batches according to their order in the food supply chain. After that, traverse the *Suspect Tree* layer by layer and the first node that meets the same criterion will be picked up as the root source since the original contaminant is always on the top over cross contaminant in the tree.

4.3. BackTracking Algorithm. In order to judge the performance of backtracking algorithm, *Hit Rate* and *False Alarm Rate* are put forwards to denote the algorithm's ability of capturing infected products and the probability of reckoning good products as infected ones by mistake. Supposing the total number of products and infected products are N and I, respectively, and the algorithm selected n potentially infected products, including i infected ones, we define *Hit Rate* as i/I, and *False Alarm Rate* as $(n-i)/(N-I)$. Although, theoretically, both high *Hit Rate* and low *False Alarm Rate* are expected, there is a tradeoff between them.

The backtracking algorithm is described in Algorithm 3.

4.4. Self Correction Method. Two metrics are defined to judge the performance of the system and provide reference for latter parameters' settings.

```
(1) Input: samples' spatial information and examination results
(2) Output: contamination origin
(3) k = 0;
(4) for i = 1:#samples do
(5)     for j = 1:#(locations/batches on sample_i's path) do
(6)         if sample_i is infected then
(7)             sample_i.location_j.batch_j.BAD++;
(8)         else
(9)             sample_i.location_j.batch_j.GOOD++;
(10)        end if
(11)    end for
(12) end for
(13) for m = 1:#(locations/batches in the entire chain) do
(14)    if location_m.batch_m.GOOD ≤ ε && location_m.batch_m.BAD ≥ 0 then
(15)        Record location.batch into suspect[k];
(16)        k++;
(17)    end if
(18) end for
(19) Exclude suspects w/small BAD;
(20) if #suspect ≥ 1 then
(21)    Get food IDs passed all suspects;
(22)    if #ID ≥ 0 then
(23)        Get food IDs passed at least one suspect;
(24)    end if
(25) end if
(26) Construct "Suspect Tree" of batches according to the paths of these IDs.
(27) for n = 1:#(tree nodes) do
(28)    if (suspect[n].location_n.batch_n.GOOD ≤ ε &&
(29)            suspect[n]location_n.batch_n.BAD ≥ 0) then
(30)        origin = suspect[n];
(31)    end if
(32) end for
```

Algorithm 2: Tracing algorithm.

```
(1) Input: contaminated samples set: {Re-check}
(2) Output: infected food products set: {Bad}
(3) while (1) do
(4)     Construct a tree of location/batch according to
(5)     the paths of contaminated products in {Re-check};
(6)     Traverse the tree DFS;
(7)     Record all nodes in Bad;
(8)     Empty {Re-check};
(9)     if node.location.batch is new then
(10)        Find the food IDs passed these nodes;
(11)        Sensor them.
(12)        if food is contaminated then
(13)            Put its ID in {Re-check};
(14)        end if
(15)    else
(16)        break;
(17)    end if
(18) end while
```

Algorithm 3: Back tracking algorithm.

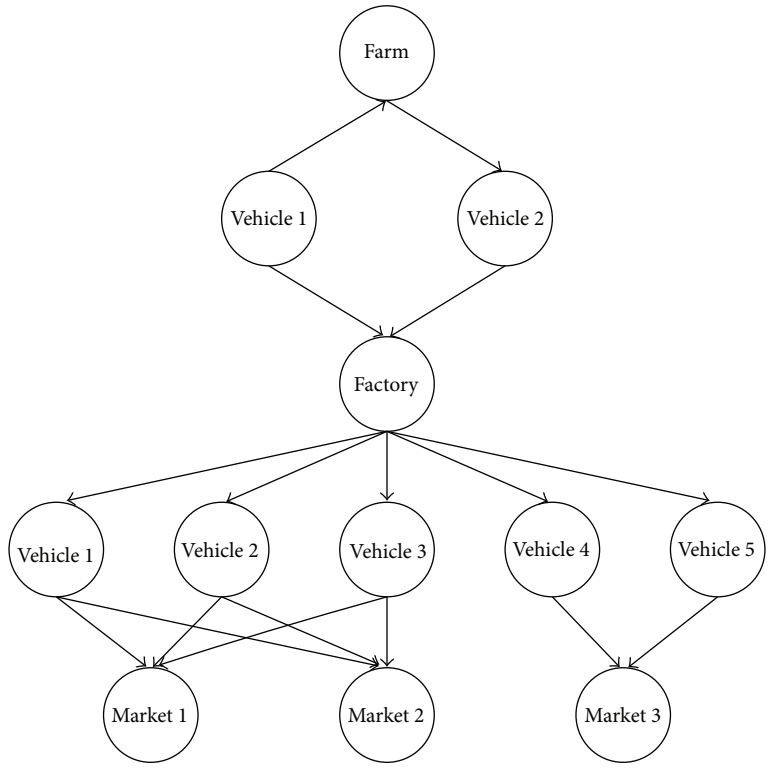

Figure 3: Topology (DAG) of a food supply chain case for evaluation.

Confidence Metric (CM) is defined as the difference between prior probability and posterior probability as follows:

$$\mathrm{CM} = \frac{\left|P_{\mathrm{post}} - P_{\mathrm{pri}}\right|}{P_{\mathrm{pri}}}. \tag{6}$$

P_{post} and P_{pri} are the posterior and prior possibilities, respectively. If CM is small, we are more confident that the dataset suits the model trained previously and vice versa. Thus, we can correct (4) and combine prior and posterior probabilities to get more reasonable infection probabilities, P_{comb}, of the entire network as follows:

$$P_{\mathrm{comb}} = \frac{P\left(A_i \mid B\right) + \mathrm{CM} \times P\left(A_i\right)}{1 + \mathrm{CM}}. \tag{7}$$

Success Metric (SM) measures the accuracy of the system as follows:

$$\mathrm{SM} = \frac{\#\mathrm{success}}{\#\mathrm{total}}. \tag{8}$$

It is defined as the ratio of successfully detected times to total tested times. With lower SM, the criterion of sampling would be set stricter and vice versa.

These two variables help to adjust sampling algorithm slightly to fit it into certain environment and applications.

4.5. Timing and Space Complexities. Suppose there are m samples, n stages, and l batches for all locations in a food supply chain, timing complexities of tracing, and backtracking algorithms are $O(mn) + O(l)$ and $O(m) + O(l)$, respectively. There exists a tradeoff between tracing accuracy and time consumption in SDPS. Obviously, more samples mean longer time and better knowledge of the network. Compared with the time spent on chemical testing and sensing, time consumption in SDPS is negligible.

A piece of record is required for every location and every food products, so space complexity of the whole IoT system is $O(a + b)$, where a is the number of food products and b is the number of locations in the network.

5. Evaluation Results and Analysis

We set up two specific cases (Figures 3 and 4) to evaluate the proposed system. The first case gives a general evaluation and shows that our SDPS scheme outperforms other sampling methods, while the second one focuses on the performance on large system and big data.

5.1. Experimental Setup. In Figure 3, note that vehicles 1 and 2 serve as the transportation node both from farm to factory and factory to market. This makes the model closer to reality as some locations in the chain can act as different characters in food procession.

The configuration of the two cases is listed in Table 1. Every location in the chain holds 25 and 500 batches in case of 1 and 2, respectively. Total of 60 and 800 thousand of food

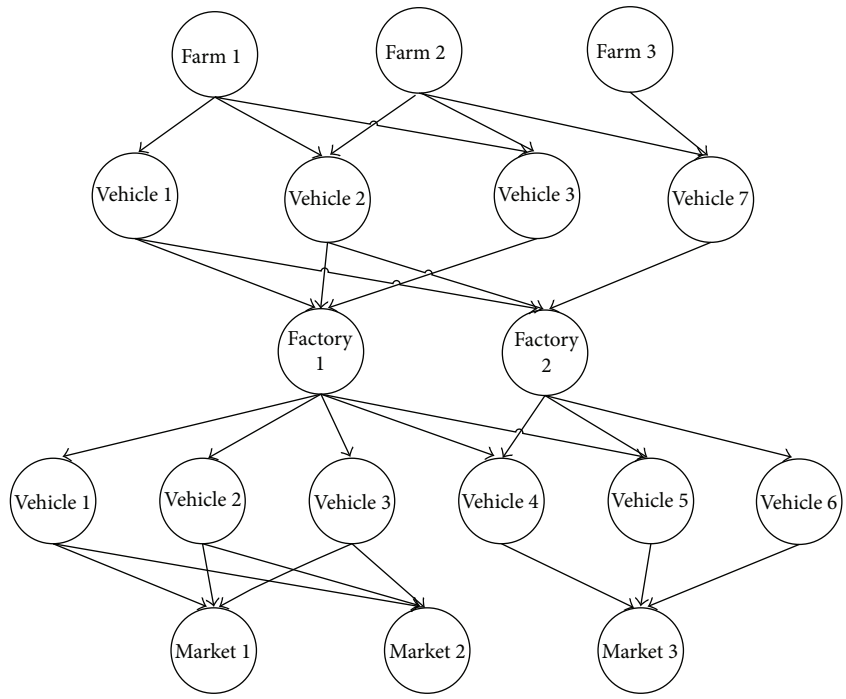

Figure 4: Topology (DAG) of a food supply chain case for evaluation (with big data).

products are circulating in the network. Which location a piece of food will pass is absolutely random.

We build the simulator in C++, which reads in the configuration files that describe supply chain topologies, generates simulated contamination behavior, and senses data along food supply chains. In simulation, the contamination source and sampling volume can be set by the users explicitly. Food contamination rules are set as that food can be infected by passing (1) contamination source directly or (2) cross infection spots indirectly according to our revised Reed Frost Model.

To make this paper compact, we only show the training process of the first model here. In the training process, we get prior probabilities which highly depend on the topology and configuration of the chain. Figure 5 shows the distribution of infected proportion under different contamination probabilities after 300,000 tests. Under each contamination probabilities (depend on contamination type), the actual portion of pollution is almost Gaussian distributed, which is the same as what we discussed in Section 4. Note that the value of x axis (x) should be transferred to the pollution proportion interval by the following function: $[(x-1)*4\%, x*4\%)$ since the total space is divided into 25 sections. The relation between the products' contamination percentage and the tracing algorithm accuracy under different sample rates is shown in Figure 6. For clarity, only 3 sampling rates are tested: 3%, 5%, and 10%. To make sample strategy more efficient, more rates can be evaluated in real situations. Figure 6 confirms the hypothesis we proposed: the source is difficult to be detected if only a small or too large part of food is contaminated. In both ways, the flow path of contamination is hidden easily.

Table 1: Configuration of the two cases.

	Case 1	Case 2
#batches/location	25	500
#total products	60,000	800,000
Flow rule	Random	Random

Table 2: Simulation results of back tracking algorithm.

Hit rate	False alarm rate
96%	3%

5.2. Evaluation Results. Figure 7 shows the accuracy of the tracing algorithm. In different probability of infection in the whole chain, the accuracy can achieve no less than 80%. In Figures 8 and 9, partition and dynamic strategy in SDPS are tested, respectively. In all probabilities of infection cases, partition strategy has higher tracing accuracies than that of global sampling strategy (11.8%, 22.7%, 10.9%, and 4.1% higher with the infection probabilities of 30%, 67%, 80%, and 90%). And compared with sampling in fixed rates (3%, 5%, and 10%), dynamic method achieves higher tracing accuracy even with a lower average sampling rate of 7.8%.

For backtracking part, the result of simulation is shown in Table 2, Both *Hit Rate* and *False Alarm Rate* are satisfactory.

Case 2 has a large data scale. We also fetch a few each times and let the system tell us the amount of samples to get next time based on (4). System's actual sampling rate turns to be 7.8%. As in Figure 10, the accuracy of tracing algorithm is higher than 80% as well, which shows that our proposed approach works well with big data.

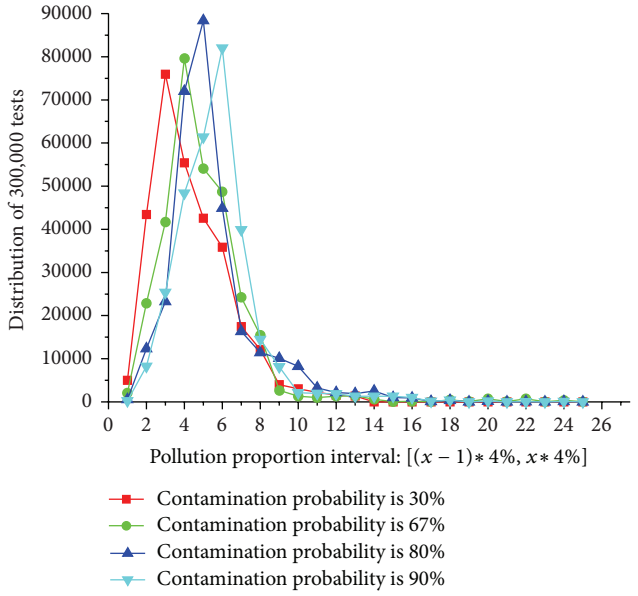

Figure 5: Prior probabilities distribution under different contamination probabilities: 30%, 67%, 80%, and 90%.

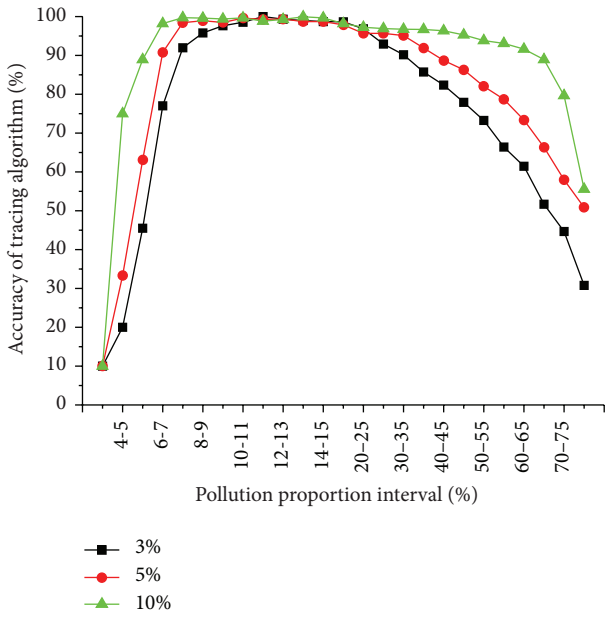

Figure 6: The relationship of tracing algorithm accuracy and pollution proportion intervals under different sample rates: 3%, 5%, and 10%.

5.3. Contamination Visualization. With the tool introduced by [24] and the information we choose to record, SDPS provides sampling data that can be represented visually after being tested by sensors. The Figures 11 and 12 show the data flow of infected and uninfected food products, respectively. In this case, we use the configuration in Case 1 and set the contamination source to be the 4th batch in factory in advance.

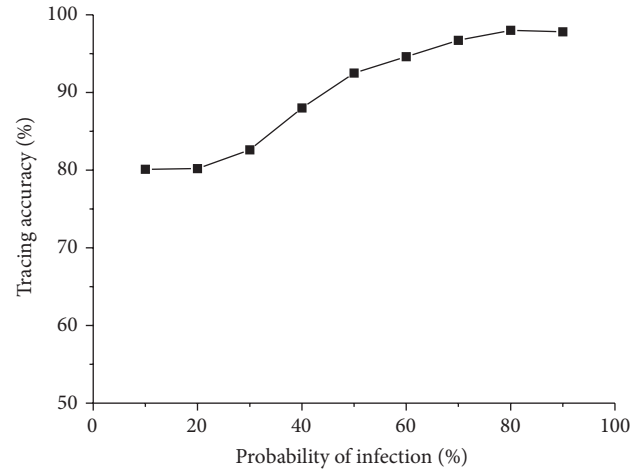

Figure 7: Accuracy of tracing algorithm with different probabilities of infection.

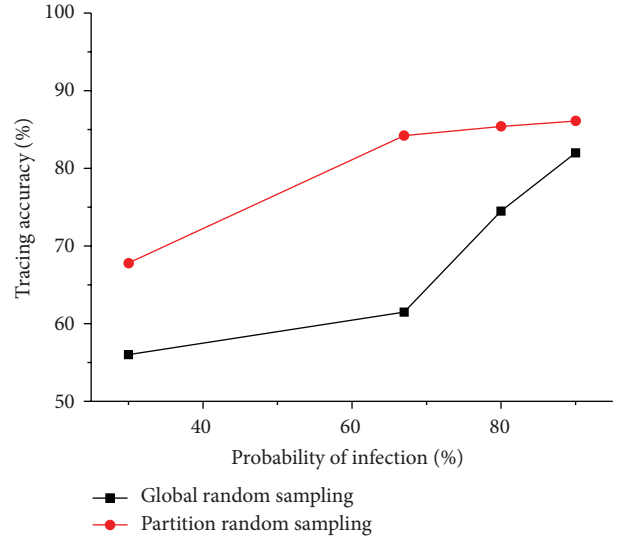

Figure 8: Simulation results of partition sampling strategy: tracing accuracy under infection probabilities of 30%, 67%, 80%, and 90%.

The vertical lines marked with locations and batches numbers represent the nodes in data flow. Lines going through these nodes are the traces of food products. As shown in those figures, most of the infected food while none of the uninfected food passed the 4th batch of factory (the node with a circle around it). So, it has a great chance to be the source of contamination, which is also proven by our tracing system.

SDPS makes data concise but still comprehensive, which facilitates visualization tool displaying the useful information.

Apart from aiding detecting contamination source, visualization can also help to know contamination conditions (e.g., contamination severity/distribution) of the whole IoT system better. For example, a well-managed warehouse or a city with lower temperature may lead to less contamination.

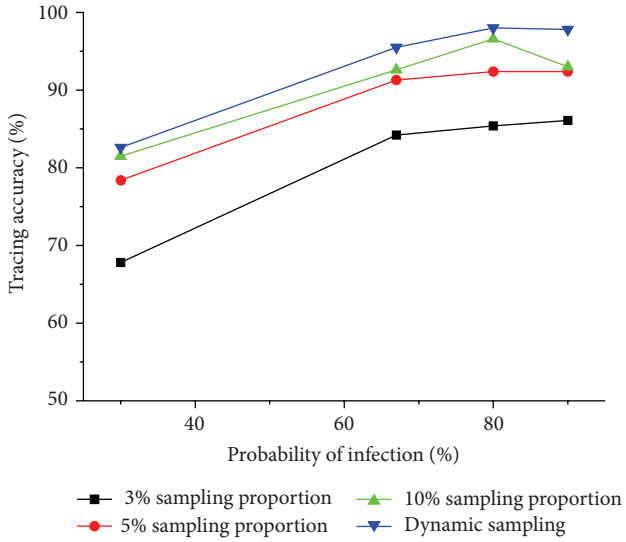

Figure 9: Simulation results of dynamic sampling strategy: tracing accuracy under infection probabilities of 30%, 67%, 80%, and 90%.

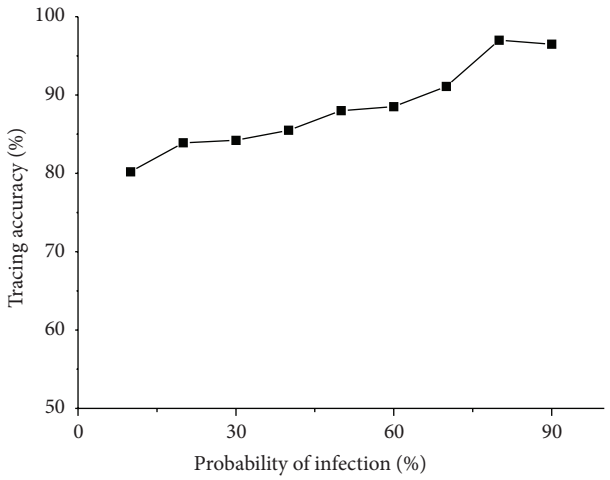

Figure 10: Simulation results of tracing accuracy with big data under different infection probabilities.

Information of these kinds can be directly read from visualization images and help manufacturers to design their food supply chain more scientifically.

Visualization of the contamination condition in IoT system makes the provenance reasoning in food supply chain intuitive and informative.

5.4. Performance Estimation in Real Situations. In reality, food supply chain is more complicated. A lot of factors, such as specific food type, environment, temperature, manufacture process, and other parameters, could make the chain difficult to predict. To implement our strategy into real situations, those factors should be concerned and some parameters should be adjusted accordingly.

The factor that influences the behavior of food supply chain the most is the type of food. Different food has its own characteristics, which may dominate the model, the provenance procedure, and expecting results. Firstly, food type can decide the possible contamination source. In (1), P_{exp} and P_{imp} are related with the virus that spreads among food. For example, avian influenza virus, which is a common infectious disease among poultry, began to be contiguous among human beings. After the mutation, infectious ability of this virus grew significantly. As a result, P_{exp} and P_{imp} of avian influenza virus in chickens would also increase. Secondly, food type is a deciding factor for its storage pattern and quality guarantee period. Some canned drinks are stacked layer by layer separately, so they would not got cross contaminated. However, raw meat is generally kept together, which provides an easy environment for virus to spread. Thirdly, the state of food is also dominant in provenance. One piece of food in solid state can be seen as a unit, while liquid food, like yogurt, could be ruined by only one deteriorated drop. In this way, for yogurt, the sampling process could be very different as spatial position should be taken into consideration and virus' behavior of liquid should also be studied.

Besides food type, there are other factors playing important roles in real situation. Food in summer is more likely to turn rotten than winter; some manufacturing factories are more hygienic than others; with time passing, food may get easier to be infectious; and the types or dosages of food addictives may make the contamination process slowed down.

Although different food supply chain can behave variously, our proposed strategy can cover most of the cases because it obeys the general model of food supply network and epidemiological principles.

6. Conclusion

In this paper, we present a heuristic approach to tracing contamination sources in large IoT systems for complicated food supply chains, which is a critical issue in metropolitan life. In our approach, Self-adaptive Dynamic Partition Sampling (SDPS) Strategy was proposed to collect data for sensors, whose input is only a small portion of end market samples from huge volume of samples along food supply chains. The approach was illustrated with a case study of IoT system about provenance in food supply chain, which can efficiently stop the outbreaks of foodborne disease. With the intelligent SDPS Strategy, objects tested by sensors are the most reasonable portion of the entire products set. The efficiency is highly improved and the accuracy stays almost the same as sensing all the objects at the same time. SDPS keeps the integrity of information and approaches a nearly real-time examination. Also, we present a tracing algorithm to find the contamination sources of food supply chains, and a backtracing algorithm to provide strategy for recalling problematical food undiscovered in the chain. It is indicated in simulation results that our SDPS scheme can achieve up to the tracing accuracy of 97.8% with a smaller average sampling percentage compared with traditional global random sampling. We managed to sample a small portion of food only

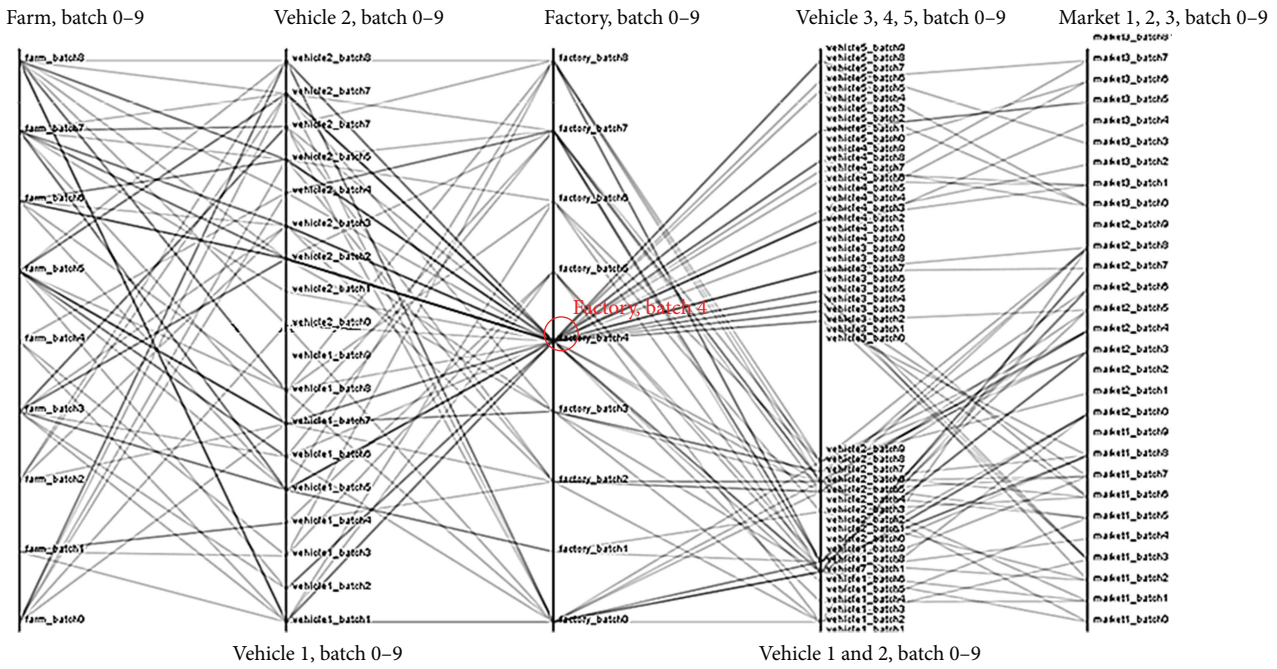

Figure 11: Visualized data flow of infected food products in food supply chain.

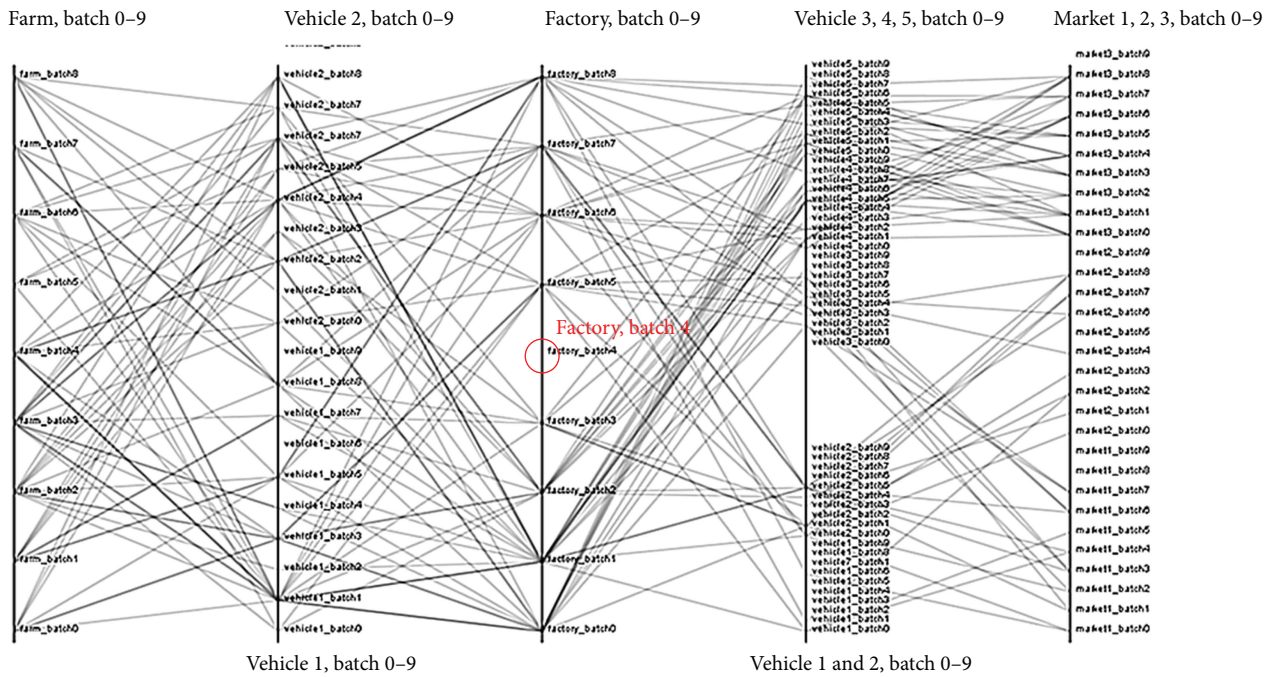

Figure 12: Visualized data flow of uninfected food products in food supply chain.

in the end market without loss in accuracy of provenance tracing over the whole IoT system. In addition, our analytic data and visualized images can clearly model contamination conditions in food supply chain within the context of IoT system. This will give the clients an intuitive impression on food supply networks in a city.

In this paper, we assume that all provenance information of food products is hosted by a centralized repository and these provenance metadata are organized in a uniform manner. Our future work is to further make practical implementation of the provenance of food supply chain in a community as our testing bed for megacity management.

Acknowledgments

This paper is sponsored in part by the Shanghai International Science and Technology Collaboration Program under Grant 13430710400, and Campus for Research Excellence and Technological Enterprise (CREATE) program of Singapore National Research Foundation under the joint project on Energy and Environmental Sustainability Solutions for Megacities (R-706-000-101-281) by Shanghai Jiao Tong University (SJTU) and National University of Singapore (NUS). Professor Qiu is partially supported by NSF CNS-1249223 and NSFC 61071061.

References

[1] "Wikipedia on smart city," http://en.wikipedia.org/wiki/Smart_city.

[2] M. Qiu and E. H. M. Sha, "Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems," *ACM Transactions on Design Automation of Electronic Systems*, vol. 14, no. 2, article 25, 2009.

[3] J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, and Z. Gu, "Online optimization for scheduling preemptable tasks on IaaS cloud systems," *Journal of Parallel and Distributed Computing*, vol. 72, no. 5, pp. 666–677, 2012.

[4] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Proceedings of the International Conference on Electronics, Communications and Control (ICECC '2011)*, pp. 1028–1033, Zhejiang, China, September 2011.

[5] X. Tang, J. Pu, K. Cao, Y. Zhang, and Z. Xiong, "Integrated extensible simulation platform for vehicular sensor networks in smart cities," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 860415, 10 pages, 2012.

[6] P. Vlacheas, R. Giaffreda, V. Stavroulaki et al., "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 102–111, 2013.

[7] A. Asin, "Smart cities from libelium allows systems integrators to monitor noise, pollution, structural health and waste management," *Smart Cities Articles*, 2011.

[8] Kevin Ashton, "That "internet of things" thing," *RFID Journal*, 2011.

[9] P. Magrassi and T. Berg, "A world of smart objects," Gartner Research Report TR-17-2243, 2002.

[10] Oxford English Dictionary (OED), "The fact of coming from some particular source or quarter, source, derivation," http://en.wikipedia.org/wiki/Provenance.

[11] A. V. Roth, A. A. Tsay, M. E. Pullman, and J. V. Gray, "Unraveling the food supply chain: strategic insights from China and the 2007 recalls," *Journal of Supply Chain Management*, vol. 44, no. 1, pp. 22–39, 2008.

[12] S. Miles, P. Groth, S. Munroe, and L. Moreau, "Prime: a methodology for developing provenance-aware applications," *ACM Transactions on Software Engineering and Methodology*, vol. 20, no. 3, article 8, 2011.

[13] R. Hasan, R. Sion, and M. Winsltt, "The case of the fake Picasso: preventing history forgery with secure provenance," in *Proceedings of the 7th Conference on File the Storage Technologies (FAST '09)*, pp. 1–14, New York, NY, USA, December 2009.

[14] T. A. McMeekin, J. Baranyi, J. Bowman et al., "Information systems in food safety management," *International Journal of Food Microbiology*, vol. 112, no. 3, pp. 181–194, 2006.

[15] M. A. van der Gaag, F. Vos, H. W. Saatkamp, M. van Boven, P. van Beek, and R. B. M. Huirne, "A state-transition simulation model for the spread of Salmonella in the pork supply chain," *European Journal of Operational Research*, vol. 156, no. 3, pp. 782–798, 2004.

[16] L. M. Wein and Y. Liu, "Analyzing a bioterror attack on the food supply: the case of botulinum toxin in milk," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 28, pp. 9984–9989, 2005.

[17] L. Qin and Q. S. Wang, "Food supply chain quality management model and simulation based on game," in *Proceedings of the International Conference on Computer Modeling and Simulation (ICCMS '09)*, pp. 291–293, Macau, China, February 2009.

[18] Q. Zhang, D. Wang, T. Huang et al., "Modelling provenance in food supply chain to track and trace foodborne disease," in *Proceedings of the International Conference on Computer Modeling and Simulation*, pp. 69–75, Hong Kong, China, February 2012.

[19] S. Li, L. Xu, and X. Wang, "Compressed sensing signal and data acquisitio in wireless sensor," *IEEE Transactions on Industrial Informatics*, 2012.

[20] Z. Ding and X. Gao, "A database cluster system framework for managing massive sensor sampling data in the internet of things," *Chinese Journal of Computers*, vol. 35, no. 6, pp. 1175–1191, 2012.

[21] L. Zhang, J. Liu, and H. Jiang, "Energy-efficient location tracking with smartphones for IoT," in *Proceedings of the IEEE Sensors*, pp. 1–4, Taipei, China, October 2012.

[22] H. Abbey, "An examination of the Reed-Frost theory of epidemics," *Human Biology*, vol. 24, no. 3, pp. 201–233, 1952.

[23] L. Elveback, J. P. Fox, and A. Varma, "An extension of tee reed-frost epidemic model for the study of competition between viral agents in the presence of interference," *The American Journal of Epidemiology*, vol. 80, no. 3, pp. 356–364, 1964.

[24] X. Yuan, H. Guo, H. Xiao, Z. Wang, and X. Zhang, "High-dimensional data virtualization," in *Proceedings of the Communications of the CCF*, pp. 13–16, April 2011.