Trust-Based Time Series Data Model for Mobile Crowdsensing*

Xiao Ma, Zhenzhe Zheng, Fan Wu†, and Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University
{yusufma555, zhengzhenzhe}@sjtu.edu.cn, {fwu, gchen}@cs.sjtu.edu.cn

***Abstract*—The recent proliferation of mobile devices embedded with capable sensors, provides an opportunity to the popular concept of mobile crowdsensing. By studying the correlation of crowd-sensed data in both spatial and temporal dimensions, we can get a clear understanding of the intrinsic pattern of data in mobile crowdsensing, which is the basic for further data analysis, such as data filtering, smoothing and prediction. However, the crowd-sensed data are normally noise and unreliable due to the diverse mobility patterns and selfish behaviours of mobile users, making the classical data models in wireless sensor networks fail in this new context. In this paper, we propose a robust and reliable time series data model based on Dynamic Bayesian Network to describe the characteristics of the crowd-sensed data. The proposed data model can figure out the spatial and temporal correlation of data in the environment, where the data has high noise levels and mobile users are untrustworthy. We conduct extensive evaluations based on both simulation and a real-world data set. Our evaluation results show that our method successfully modeled the crowd-sensed time series data with effectiveness, efficiency and trustworthiness.**

I. Introduction

In recent years, we have witnessed the rapid and explosive growth of capable human-carried mobile devices, e.g., smartphones, smartbands, smartglasses. The smart devices embedded with powerful sensors, such as GPS, compass, and accelerator, provide a new paradigm for data collection, namely mobile crowdsensing, and revolute the traditional wireless sensor networks. People have deployed numerous mobile crowdsensing applications, including the indoor positioning [1], smart transportation [2], health care [3], social emergency events detection [4] and et al.

The sensed data collected by mobile crowdsensing systems always have complex relations in the spatial and temporal dimensions. Although, in the traditional sensor networks, there are already some works proposing different time series models to capture the spatial and temporal correlation among sensed data, in the mobile crowdsensing, due to the following two challenges, those models would be not applicable anymore. The first challenge comes from the mobility of crowd. In traditional sensor networks, the sensor devices are usually fixed in some pre-determined locations, but the users in mobile crowdsensing systems always have complex and unpredictable mobility patterns, leading to high noise levels of collected sensed data. For such noise and uncertain crowd-sensed data, it is extremely difficult to exploit the spatial and temporal correlation to facilitate data analysis, such as smoothing, filtering, and prediction.

The second critical challenge is the unreliability of mobile users. The mobile users' usage behaviors and activity contexts would have an impact on the status of the smart devices, and further influence the quality of collected data. For example, in noise map construction of the crowdsensing system, putting smartphones in mobile users' pockets or bags would result in reporting different records of the noise levels in the same location. In addition, the selfish mobile users may report low-quality data to get extra payments. Thus, we should take the trustworthiness of mobile users into account when designing crowd-sensed data model. These issues are all not considered in traditional sensor networks, and the corresponding models and methods are intrinsically unable to handle both the high noise levels of crowd-sensed data and the trustworthiness of mobile users.

In order to overcome the above two challenges in mobile crowdsensing, the proposed data model should have the following properties:

1) **Effectiveness**: The data model should be able to capture the complex spatial and temporal correlation of crowd-sensed data, even if the data has a high level of noise and uncertainty.
2) **Efficiency**: The computational complexity of the data model should be controlled within an acceptable threshold to adapt to the requirement of large scale mobile crowd-sensing systems.
3) **Trustworthiness**: The data model should consider the impact of the unreliability of mobile users on the quality of collected data, and give a quantity metric to measure the trustworthiness of mobile users.

In this paper, jointly considering the two challenges, we propose a reliable time series data model for mobile crowdsensing to satisfy the above three properties. We first use random variables to describe the noisy and uncertain crowd-sensed data collected by mobile users. As different mobile users would have different trustworthiness levels, we then

†F. Wu is the corresponding author.

*This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61672348, 61672353, 61422208, 61472252, 61272443 and 61133006, in part by Shanghai Science and Technology fund 15220721300, in part by CCF-Tencent Open Fund, and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

assign each mobile user a confidence parameter to model his trustworthiness level. We also quantify the impact of the confidence parameter on the distribution of random variables. After that, we propose a reliable time series data model using a Dynamic Bayesian Network, satisfying the properties of effectiveness, efficiency, and trustworthiness. We show that the powerful Dynamic Bayesian Network can not only handle the time correlations of the stochastic data but also model the spatial correlations by mapping technique [5], similar to the singular value decomposition(SVD), even when the collected crowd-sensed data has a high level of noise and uncertainty. To the best of our knowledge, this is the first work that handles the untrustworthy time series data modeling in mobile crowdsensing. Based on the proposed reliable time series data model, we can conduct the data forecasting, missing value imputation and other applications on time series for the noise and uncertain crowd-sensed data.

In this paper, our main contribution is to establish a new time series data model to handle the noisy and unreliable data collected from mobile crowdsensing. It shows how to exploit the spatial-temporal correlation even the data source is unreliable and the noise level is very high. We conducted extensive evaluations to demonstrate the effectiveness of our proposed model based on the real-world data.

The rest part of the paper is organized as follows: In Section 2, we review the related work; in Section 3, we will introduce the system model of our approach; in Section 4, we propose the reliable time series data model, and in Section 5, experimental results will be presented. Finally, we conclude the paper in Section 6.

II. Related Works

In recent years, crowdsourcing and crowdsensing have attracted increasing interest [1], [2], [6], [7], [8], [9], [10]. In [11], some of the existing challenges and potential topics have been discussed. In [6], a programming framework for mobile crowdsensing was proposed. In [7], Jin, Haiming, et al. proposed a novel incentive mechanism integrating data aggregation and data perturbation. Specifically, it helps to select workers who are more likely to provide reliable data. In [2], Hu, Shaohan, et al. developed the *Smartroad* traffic event detection system to process the GPS data of in-vehicle smartphones collected through participatory sensing. Its results can be used for many assisted-driving or navigation systems. Another example, in [1], a novel indoor floor reconstruction model was proposed based on crowdsensing, which leverages the crowdsensed data from mobile users, extracting the position, size and orientation information of individual landmarks, as well as obtaining the spatial relation between the adjacent landmarks. In addition to the works mentioned above, there are still many interesting topics being discussed by researchers and we will not list them here.

Similarly, time series always catches the researchers' interests since there are always unexpected aspects for us to explore [3], [12], [13], [14], [15], [16], [17], [18]. In [3], [12], time series method are used to study the health-related problems. Kale, David C., et al. developed a new distance metric for multivariate time series with application to health care[3]; Caballero Barajas et al. It makes use of the locality sensitive hashing, solving the dilemma between quality and speed, enabling distance measuring with a fast search and a high quality. In [14], Jha, Abhay, et al. claimed that in the business scenario, the time series could be sparse if the commodity at the very beginning and the forecasting the time series could be hard with these data. They proposed a clustering model based on PLS regression and OPTMOVE clustering algorithm to forecast the sparse time series based on their similar time series. In a similar scenario with the crowdsensing, the sensor networks, SMiLer([15]) makes use of both kNN and Gaussian Process, solving the problem of the heavy cost introduced by Gaussian Process and output the prediction result with a superior accuracy and an effectively measured uncertainty. [5], [19] focus on the missing value imputation of time series, which is also another important task. [19], focusing on the medical time series, exploited fact that the missing data may appear to be lag-correlated, inputting the missing data using kNN; [5] makes use of the "smoothness" and the "correlation" of the time series data, introducing a Dynamic Bayesian Network based method for missing data imputation, which further supports prediction, smoothing and pattern recognition. There are still many works focusing on solving the realistic problem in time series, and we will not list them one by one.

Nevertheless, all of these previous time series models are not capable for the crowdsensing. Different from the aforementioned prior works, we focus on the modeling of the time series in the crowdsensing. As mentioned in the previous section, the crowdsensed data has complex relations in the spatial and temporal dimensions, but none of the current works has considered this factor. Hence, we aim to solve this problem, the time series modeling in crowdsensing. Inspired by [5], [20], we develop a novel way to model the time series in the crowdsensing network. The general idea is to treat every report as a random variable and "tag" every it with a user correlated trustworthiness parameter. Then the reports will be synthesized into time series data, which will be modeled using a Dynamic Bayesian Network.

III. System Model

In this model, a crowd of k users $U = \{1, 2, \ldots, n\}$ keeps collecting data in a given area. With the arbitrary movement of the users, data will be generated at different location and at different time. Each user i reports u_i reports, namely $\{r_i^{(1)}, r_i^{(2)}, \ldots, r_i^{(u_i)}\}$, where each observation $r_i^{(j)}$ consists of the following four values: (a) the user measured value $v_i^{(j)} \in \mathcal{R}$, which is a noisy observation; (b) the time $a_i^{(j)}$ indicating when the data was generated; (c) the geographic location $s_i^{(j)}$, represented in longtitude and latitude where the data was generated; (d) an estimate of the precision of the user observation $\theta_i^{(j)} \in R_{>0}$. Thus, each report has form of

$r_i^{(j)} = \langle v_i^{(j)}, a_i^{(j)}, s_i^{(j)}, \theta_i^{(j)} \rangle$, and we will have $u = \sum_{i=1}^{n} u_i$ reports in total.

To model the uncertainty of the noisy data in crowdsensing, we assume in each report, the uncertainty is distributed normally. Given $r_i^{(j)}$, the probabilistic density function of the generic point v could be expressed as

$$\begin{aligned} p(v \mid r_i^{(j)}) &= p(v \mid v_i^{(j)}, \theta_i^{(j)}) \\ &= \sqrt{\frac{\theta_i^{(j)}}{2\pi}} e^{-\frac{\theta_i^{(j)}(v - v_i^{(j)})^2}{2}} \end{aligned} \tag{1}$$

Next, we consider another property of the crowdsensing, the untrustworthiness. In crowdsensing, no user can be trusted. Because people tends to provide more quantity of data to earn more money but ignore the quality of the data or they simply fabricate the data to spoof the system. Thus, the reliability of the crowd-sensed data can pose another challenge to the crowdsensing. Formally, if the report is fully trustworthy, we have the following condition:

$$v_i^{(j)} \sim \mathcal{N}(v \mid v_0, \theta_i^{(j)}), \mathbb{E}(v_i^{(j)}) = v_i^{(j)*}$$

where $v_i^{(j)*}$ is the ground truth value around location $s_i^{(j)}$. More specifically, the fully trustworthy reports can be regarded as the samples from a normal distribution whose expectation is the ground truth value. On the contrary, the untrustworthy reports are not necessarily related to the ground truth value, $v_i^{(j)*}$. Some deviations may be possible. For example, we may have $v_i^{(j)} \sim \mathcal{N}(v_i^{(j)*} + b, \theta_i^{(j)})$ with a bias b from the ground truth $v_i^{(j)*}$.

Given this, we introduce another set of parameters, namely trustworthiness parameters, as $\mathbf{c} = \{c_1, c_2, \ldots, c_n\}$, where the parameter $0 \leq c_i \leq 1$ denotes the trustworthiness of user i. Then we can update our probability density function as:

$$\begin{aligned} p(v \mid r_i^{(j)}, c_i) &= \mathcal{N}(v \mid v_i^{(j)}, c_i \theta_i^{(j)}) \\ &= \sqrt{\frac{c_i \theta_i^{(j)}}{2\pi}} e^{-\frac{c_i \theta_i^{(j)}(v - v_i^{(j)})^2}{2}} \end{aligned} \tag{2}$$

The trustworthiness parameter c_i represents the uncertainty of the confidentiality of the data uploaded by the user; $c_i = 1$ means the user can be fully trusted, and the above function equals to (1); the smaller c_i is, the noisier the variable $v_i^{(j)}$ would be.

In the crowdsensing based network, we have a set of fixed target nodes $L = \{l_1, l_2, \ldots, l_q\}$, where we are interested in the trend of the time series. For each location, it consists of a longitude and a latitude. We assume there is a global clock in the system such that the time at each position is the same. For simplicity, let the counter $t \in \mathbb{N}$. Given these information, we can represent our stream time series at l_k as $X_k = \{x_{k,t}\}_{t=1}^{\infty}$, where each $x_{k,t} \in X_k$ is an aggregation of reports belonging to the cluster of this node at the time slot t. The cluster is just based on the geographical distance of the location of the data to the node, and for simplicity, here, we define the cluster to be the circle centered at the node with a given radius.

We then turn to find the ground-truth of the data in crowdsensing. Here, we use Dynamic Bayesian Network to accomplish this process. Our motivation comes from the correlation between time series and its own smoothness. Smoothness means that $x_t \simeq x_{t+1}$, that the values in the nearby time slot can be tightly related to each other. We can denote that $x_{t+1} = g(x_t) + w_t$. Here, g is a function and w_t is white noise. For every type of time series data, there is an intrinsic property that different time series may appear in the similar pattern, which we call it a correlation. For example, the time series of the acceleration of two elbows when you are running may be very similar with only a time lag. Dynamic Bayesian Network can help us combine these two properties, then we can handle the sequences, discovering the latent relation between time series, and output the groud-truth. We will introduce our Dynamic Bayesian Network based method in detail in the latter section.

IV. Solution

In this section, we will introduce our proposed model and algorithm in detail. First, the fusion of the data will be covered; second, we will talk about the Dynamic Bayesian Network and its learning procedure.

A. Fusing Untrustworthy Reports

In our assumption, reports will be fused into one value of a time series at a point t only when they are geographically within the range of point k and the observed time $a_i^{(j)}$ belongs to the time slot t. Just as mentioned before, we only consider the range as a circle centered at l_k with radius d_k. Note that the circles can overlap, hence one reports could contribute to multiple values of different nodes at the same time tick. Specifically, given a report $r_i^{(j)}$ and a location l_k and time slot t, denote the set of reports contribute to the value $x_{k,t}$ as $\mathbf{R}_{k,t}$, then we have the following definition:

$$r_i^{(j)} \in \mathbf{R}_{k,t} \Leftrightarrow ||l_k - s_i^{(j)}|| \leq d_i \text{ and } t < a_i^{(j)} < t + 1 \tag{3}$$

According to equation (3), we can find the the set of m reports $\mathbf{R}_{k,t} = \{r_{k,t}^{(1)}, r_{k,t}^{(2)}, \ldots, r_{k,t}^{(m)}\}$ contributes to l_k at time slot t, then we use a function f_k specialized for l_k to fuse these reports into one probability density distribution. Normally, we choose function f_k as sum function or average function. Note that all of the reports have the similar trust-based PDF $p(v \mid r_i^{(j)}, c_i)$, then the result will still be an Gaussian distribution, namely $f_k(\mathbf{R}_{k,t})$. In many previous works, covariance intersection(CI) is widely used for data integration. However, in our scenario, traditional CI behaves poorly since the trustworthiness of user is not considered. We refered to the work of Matteo Venanzi et al., consider the set $\mathbf{R}_{k,t}$, and set the fusion function as follows. To be concise, let $\mathbf{R}_{k,t}$ be $\mathbf{R}_{k,t} = \{r_1, r_2, \ldots, r_m\}$ where $r_i = \langle v_i, a_i, l_i, \theta_i \rangle$ and the corresponding trustworthiness parameter of report r_i be c_{r_i}.

$$x_{k,t} = f(\mathbf{R}_{k,t}) \tag{4}$$

$$f(\mathbf{R}_{k,t}) = \mathcal{N}(x_{k,t}|v_f, \theta) \tag{5}$$

$$\theta_f = \sum_{i=1}^{m} c_{r_i}\theta_i \tag{6}$$

$$v_f = \theta_f^{-1} \sum_{i=1}^{m} c_{r_i}\theta_i v_i \tag{7}$$

Specifically, this trust-based fusion model is obtained by fusing the estimates as jointly weighted by the precision and the trustworthiness parameter of the user.

B. Dynamic Bayesian Network

Dynamic Bayesian Network is a kind of Bayesian Network specialized for handling time evolving events. Here is an illustration of Dynamic Bayesian Network in Fig. 1. In this network, we assume the linear projection $\mathbf{G}$ maps the latent variable $\mathbf{Z}_t$ to the fused data $\mathbf{X}_t$, where $\mathbf{X}_t$ is a vector of the fusion of the raw data. This projection automatically catches the spatial correlation between the data, just like SVD. To model the temporal correlation, we assume the latent variable are time dependent on the previous latent variable through a linear projection matrix $\mathbf{F}$, which is according to the smoothness of time series. The transition functions can be quantified as follows:

$$\mathbf{Z}_1 = \mathbf{Z}_0 + \omega_0 \tag{8}$$

$$\mathbf{Z}_{t+1} = \mathbf{F}\mathbf{Z}_t + \omega_t \tag{9}$$

$$\mathbf{X}_t = \mathbf{G}\mathbf{Z}_t + \epsilon_t \tag{10}$$

where $\mathbf{Z}_0$ is the initial value of the latent variable, and $\omega_0 \sim \mathcal{N}(0, \Gamma_0)$, $\omega_t \sim \mathcal{N}(0, \Gamma_1)$, $\epsilon_t \sim \mathcal{N}(0, \Gamma_2)$ are Gaussian white noises. Besides, the fused-data is generated from the raw data by function (4)-(7). The joint distribution of $\mathbf{Z}$ and $\mathbf{X}$ is given by

$$p(\mathbf{Z}, \mathbf{X}) = p(\mathbf{Z}_0)\prod_{i=1}^{T} p(\mathbf{Z}_i \mid \mathbf{Z}_{i-1})\prod_{i=1}^{T} p(\mathbf{X}_i \mid \mathbf{Z}_i) \tag{11}$$

where the $\mathbf{V}_i$ denotes the collection of measures reported at time t.

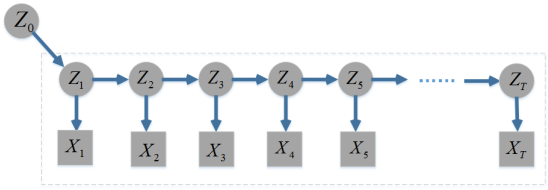


Fig. 1: Dynamic Bayesian Network Constructed

C. Model Learning

Given the above model, we propose our learning method, trust-based EM algorithm. We have the following parameters to estimate:

1) The transition parameter $\mathbf{F}$ and $\mathbf{G}$.
2) $\mathbf{Z}_0$ and the covariance matrices of the Gaussian white noises Γ_0, Γ_1 and Γ_2.
3) The hidden variables $\{\mathbf{Z}_i\}$.
4) The trustworthiness parameters $\mathbf{c}$ of the users.

To be concise, we denote the collections of the parameters as $\alpha := \{\mathbf{F}, \mathbf{G}, \Gamma_0, \Gamma_1, \Gamma_2, \mathbf{c}\}$. Normally, the goal of the parameter optimization is to maximize the log-likelihood of the network, $\mathcal{L}(\alpha) = P(\mathbf{Z}, \mathbf{X})$. However, this is not an easy task, since we have a set of hidden variables $\mathbf{Z}$ in this network and simply using the maximum-likelihood estimation can be quite expensive. An alternative way is using the EM algorithm. Traditional EM algorithm has two steps:

1) *E step*: estimate the distribution of the latent variables, i.e. the distribution of Z_i for every i
2) *M step*: choose the parameters to maximize the log-likelihood of the network

However, in our scenario, trustworthiness parameters are introduced and $\mathbf{X}$ are not directly observed variables as the normal case. Actually, they are the fusion of the measurements, which will change as we update the trustworthiness parameters. Hence, it could be hard for the current EM algorithms to learn the parameters.

We have to consider two factors. First, in order to model the temporal trend of the time series, we have to maximize the log-likelihood of the Dynamic Bayesian Network. Second, due to the unreliability of the users, their reports are not fully trustworthy and we want to know the trustworthiness level of each user. However, maximizing one of them does not mean that we can get the optimal solution of the other one. Hence, to handle this dilemma, we propose our trust-based EM algorithm to find the best trade-off between them. Our basic idea is to maximize the log-likelihood of the joint distribution of the network and the reports. The likelihood of the network is given by equation (11). As for the reports, let $\mathbf{R}(r_i^{(j)})$ be the collection of reports that report $r_i^{(j)}$ belongs to, and we define the likelihood of corresponding trustworthiness parameter as $L(c_i \mid r_i^{(j)}, f(\mathbf{R}(r)_i^{(j)})) = \int_R p(v \mid r_i^{(j)}, c_i) f(\mathbf{R}(r_i^{(j)}))dv$. Our algorithm can be stated as follows:

1) *E1 step*: For every k and t, update the distribution of each $x_{k,t}$ with

$$Q_t(x_{k,t}) = f(\mathbf{R}_{k,t}; \alpha) \tag{12}$$

2) *E2 step*: Estimate the distriution of the latent variables $\mathbf{Z}$, i.e., the distributions of Z_t for every t. Here, we use Belief Propagation for the latent variable inference.

3) *M step*: update α with α^*

$$\begin{aligned}\alpha^* &:= argmax_\alpha\{\log(p(\mathbf{Z}_0;\alpha)\prod_{t=1}^{T}p(\mathbf{Z}_t \mid \mathbf{Z}_{t-1};\alpha)\\ &\prod_{t=1}^{T}p(\mathbf{X}_t \mid \mathbf{Z}_t;\alpha)\prod_{i=1}^{n}\prod_{j=1}^{u_i}\int_R p(v \mid r_i^{(j)};\alpha)f(\mathbf{R}(r_i^{(j)}))dv)\}\\ &:= argmax_\alpha\{-D(\mathbf{Z}_1,\mathbf{Z}_0,\Gamma_0)-\sum_{t=2}^{T}D(\mathbf{Z}_t,F\mathbf{Z}_{t-1},\Gamma_1)\\ &-\sum_{t=1}^{T}D(\mathbf{X}_t,G\mathbf{Z}_t,\Gamma_2)-\frac{\log|\Gamma_0|}{2}-\frac{(T-1)\log|\Gamma_1|}{2}\\ &-\frac{T\log|\Gamma_2|}{2}+\sum_{i=1}^{n}\sum_{j=1}^{u_i}(\frac{1}{2}\log\frac{c_i\theta_i^{(j)}\theta_f}{c_i\theta_i^{(j)}+\theta_f}\\ &-\frac{c_i\theta_i^{(j)}\theta_f(v_i^{(j)}-v_f)^2}{2(c_i\theta_i^{(j)}+\theta_f)})\}\end{aligned} \tag{13}$$

Here, $D(\mathbf{v}_1,\mathbf{v}_2,\Lambda)$ corresonds to the square of the Mahalanobis distance between two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, i.e., $D(\mathbf{v}_1,\mathbf{v}_2,\Lambda)=(\mathbf{v}_1-\mathbf{v})\Lambda^{-1}(\mathbf{v}_1-\mathbf{v}_2)^T$.

We want to note that this model can be easily extent to handle the reports consists of multiple dimensions. Currently, many mobile devices have various embedded sensors and can collect multiple types of data simultaneously. In order to handle this kind of reports, we only need separately handle each dimension using our model.

V. Experiment Results

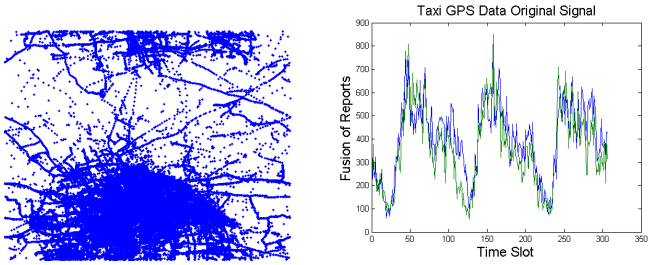
A. Data

We tested the performance of our model on two datasets: one simulation dataset, another real-world dataset.

1) Simulation: We generated the test data according to the following principles:

- We assume a fixed number of users exist in this scenario, and for every user, there is a randomly generated trustworthiness parameter.
- We assume the time series at each fixed node satisfies a sinusoidal curve, and the corresponding number of reports are generated with randomly assigned user tag.
- The value of j^{th} report of user i satisfies: at time t, $v_i^{(j)}=\frac{sin(t)}{N}+\delta$. Here, N is a constant which represents the total number of reports contributes to this location and $\delta\sim\mathcal{N}(0,\frac{1}{c_0c_i})$ where c_0 is a constant.

2) Real-World Data: : We also tested our model on the Shanghai Taxi GPS dataset. This dataset consists of the GPS data of Shanghai taxis collected during July 2007. Each report consists of the taxi id, date, time, and the location information. We selected the taxi data on July 1st, 2nd, and 3rd, targeted the data around the downtown area which includes 121 locations in total, and tested the performance on two near locations. When fusing the reports, we select a fixed cluster radius and fused the reports by summation function f_s. Specifically, let the collection of reports be $\mathbf{R}=\{r_1,r_2,\ldots,r_n\}$, and user function u be $u(r_i^{(j)})=i$, then the summation function $f_s(R)=\sum_{i=1}^{n}c_{u(r_i)}$. The geological distribution of all the reports is shown in Fig. 2(a), and the original curve without processing by our model of the two nodes are shown in Fig. 2(b).



(a) Distribution (b) Original Signal

Fig. 2: Taxi Data

B. Results

We tested our model on the datasets described above on two aspects: a) prediction, by setting the tail of the time series to be empty, b) missing value imputation, by setting the middle part of the time series empty. Meanwhile, we will display that by learning the trustworthiness parameter, our model will reduce the noise and return the ground truth.

In the simulation, we generated data on 200 time ticks on three time series. The original curve can be shown in Fig. 3.

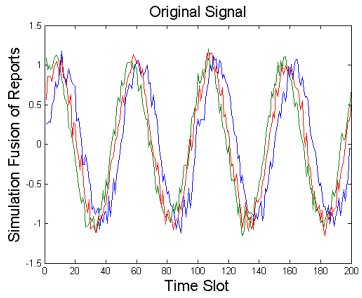
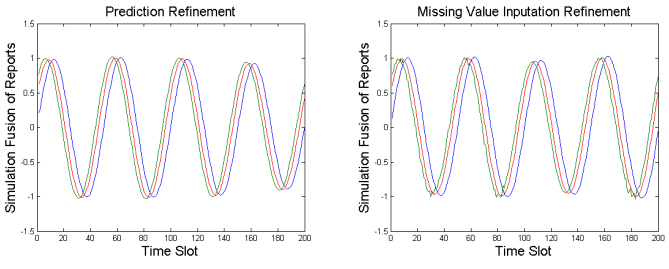


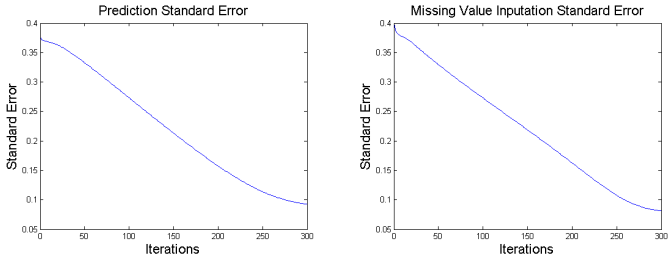
Fig. 3: Simulation Original Signal



(a) Prediction (b) Missing Value Inputation

Fig. 4: Simulation Results

For prediction, we use the first 150 data as the training set and the rest 50 data for testing. As the iteration times goes from 1 to 300, the standard error is being calculated. We find that our method successfully eliminated the noise and output the curve as expected. For missing value imputation, we also use 150 data as the training set, the rest for testing. The results can be shown in Fig. 4 to Fig. 5.



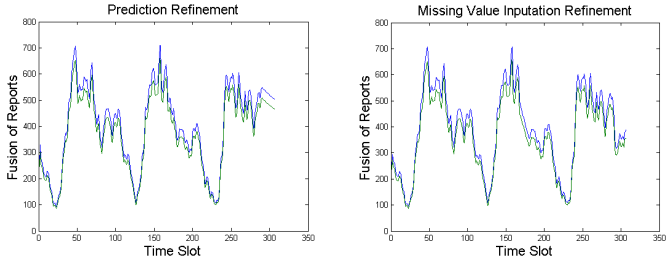
(a) Prediction (b) Missing Value Inputation

Fig. 5: Simulation Standard Error

We can find that our model successfully made the prediction and missing value imputation, as well as got the ground truth of the sinusoidal curve. The result displayed in Fig 4. tends to have some shrink at the prediction part(the rightmost part of the Fig 4.(a)) and the imputation part(the third peak of the Fig 4.(b)). With the more iterations, this defect will be eliminated.

For the taxi dataset, we randomly selected two close nodes to test its performance. We divide one day into 102 time slots, and similarly, by setting part of the time series to be the test set, we got the following results, shown in Fig. 6.

According to the experiment, our method successfully outputs the reliable time series model. It efficiently and effectively learned the trustworthiness parameter of the users then catches the spatial correlation between two time series, as well as the temporal correlation within the time series, outputting satisfying prediction and missing value imputation result. The



(a) Prediction (b) Missing Value Inputation

Fig. 6: Taxi Data Experiment

result in Fig 6. showed that the noise has been reduced when processed by our model, and our model can make the prediction and missing value imputation in this scenario.

VI. Conclusion and Future Work

In this paper, we proposed a Trust-Based Time Series Data Model based on Dynamic Bayesian Network for crowdsensing. For each user, we used a trustworthiness parameter to model his trustworthiness level. Next, we proposed a corresponding EM algorithm to learn the parameters efficiently and effectively. In the experiment, we tested our algorithm both on simulation and real world dataset and proved that our algorithm is effective at finding the ground truth, reducing noise, prediction, and missing value imputation.

However, we noticed that this model can only be applied to a limited number of time series. When handling dozens of time series, an overfitting problem may occur. We aim to solve the problem and continue to optimize our model in the future.

References

[1] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *MobiCom, 2014*.

[2] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Trans. Sen. Netw.*, vol. 11, no. 4, pp. 55:1–55:27, Jul. 2015.

[3] D. C. Kale, D. Gong, Z. Che, Y. Liu, G. Medioni, R. Wetzel, and P. Ross, "An examination of multivariate time series hashing with applications to health care," in *ICDM, 2014*.

[4] Z. Xu, H. Zhang, Y. Liu, and L. Mei, "Crowd sensing of urban emergency events based on social media big data," in *TrustCom, 2014*.

[5] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, "Dynammo: Mining and summarization of coevolving sequences with missing values," in *KDD, 2009*.

[6] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *MobiSys, 2012*.

[7] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *MobiHoc, 2016*.

[8] Z. Huo, L. Shu, Z. Zhou, Y. Chen, K. Li, and J. Zeng, "Data collection middleware for crowdsourcing-based industrial sensing intelligence," in *MobiMWareHN, 2015*.

[9] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *INFOCOM, 2012*.

[10] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, "Distributed time-sensitive task selection in mobile crowdsensing," in *MobiHoc, 2015*.

[11] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[12] K. L. Caballero Barajas and R. Akella, "Dynamically modeling patient's health state from electronic medical records: A time series approach," in *KDD, 2015*.

[13] B. Hu, Y. Chen, J. Zakaria, L. Ulanova, and E. Keogh, "Classification of multi-dimensional streaming time series by weighting each classifier's track record," in *ICDM, 2013*.

[14] A. Jha, S. Ray, B. Seaman, and I. S. Dhillon, "Clustering to forecast sparse time-series data," in *ICDE, 2015*.

[15] J. Zhou and A. K. Tung, "Smiler: A semi-lazy time series prediction system for sensors," in *SIGMOD, 2015*.

[16] C. Luo, J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang, "Correlating events with time series for incident diagnosis," in *KDD, 2014*.

[17] L. Ulanova, T. Yan, H. Chen, G. Jiang, E. Keogh, and K. Zhang, "Efficient long-term degradation profiling in time series for complex physical systems," in *KDD, 2015*.

[18] Y. Cai, H. Tong, W. Fan, P. Ji, and Q. He, "Facets: Fast comprehensive mining of coevolving high-order time series," in *KDD, 2015*.

[19] S. A. Rahman, Y. Huang, J. Claassen, and S. Kleinberg, "Imputation of missing values in time series with lagged correlations," in *ICDMW, 2014*.

[20] M. Venanzi, A. Rogers, and N. R. Jennings, "Trust-based fusion of untrustworthy information in crowdsourcing applications," in *AAMAS, 2013*.