Context-Aware Data Cleaning for Mobile Wireless Sensor Networks: A Diversified Trust Approach

Concepción Sánchez Alemán[1], Niki Pissinou[2], Sheila Alemany[2], Kianoosh Boroojeni[2], Jerry Miller[2], Ziqian Ding[2]
[1]Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA
[2]School of Computing and Information Sciences, Florida International University, Miami, FL, USA
{csanc066, pissinou, salem010, kghol002, millej, ziding}@fiu.edu

***Abstract*— In mobile wireless sensor networks (MWSN), data imprecision is a common problem. Decision making in real time applications may be greatly affected by a minor error. Even though there are many existing techniques that take advantage of the spatio-temporal characteristics exhibited in mobile environments, few measure the trustworthiness of sensor data accuracy. We propose a unique online context-aware data cleaning method that measures trustworthiness by employing an initial candidate reduction through the analysis of trust parameters used in financial markets theory. Sensors with similar trajectory behaviors are assigned trust scores estimated through the calculation of "betas" for finding the most accurate data to trust. Instead of devoting all the trust into a single candidate sensor's data to perform the cleaning, a Diversified Trust Portfolio (DTP) is generated based on the selected set of spatially autocorrelated candidate sensors. Our results show that samples cleaned by the proposed method exhibit lower percent error when compared to two well-known and effective data cleaning algorithms in tested outdoor and indoor scenarios.**

***Keywords*—Context-aware, beta-trust, trust diversification, online data cleaning, mobile wireless sensor networks**

I. Introduction

The study of infrastructure-less mobile sensor networks has taken a promising direction due to varied applications in military systems, homeland security, health care systems, environmental monitoring, and vehicular systems; as well as exponential growth of sensor interaction among millions of devices and systems merging to form the Internet of Things (IoT) [1]. Mobility adds challenges in data preservation accuracy due to data collision [2], sensor isolation, and short-term sensor connectivity [3]. The result is a significant collection of unreliable data. With so much unreliable data provided by mobile sensors, data cleaning becomes a critical factor in system functionality and reliable, real-time decision making in applications [4].

The novel data cleaning technique Diversified Trust Portfolio (DTP) proposed in this paper employs a combination of relative and absolute measurements. The calculation of "beta" measures the behavioral similarity between two nodes for each contextual parameter. Unlike other economic theories, beta analysis allows for a comparison of trajectory behavior of each candidate node with respect to a baseline node to identify dirty data with a simple, lightweight formula. In addition to the introduction of betas, this technique combines the use of diversified portfolios to find an effective trust distribution. Diversified portfolio computation seeks to minimize the error between the value of reference and the predicted value (product of our approach). Both concepts, betas and portfolio diversification, are introduced in the Capital Asset Pricing Model (CAPM) in financial-field applications [5], [6]. To the best of our knowledge, this is the first study that utilizes the beta calculation in combination with portfolio diversification from CAPM in modeling trust. The major contributions we have made with this work can be summarized as follows:

- We introduce a unique data cleaning method tailored for dynamic environments where network topology is temporary and sensor nodes must determine quickly sensor data accuracy and which data to trust.
- We present a context-aware economic-based approach to identify the trustworthy set of sensors to help during the data cleaning. Through assignments of trustworthiness weights to the selected candidate sensors, we minimized the error in estimating the missing values.

The related work can be reviewed in Section II. Our proposed method is explained in Section III. The evaluation of results is described in Section IV. Conclusions and future work are discussed in Section V.

II. Related Work

Multiple data cleaning techniques have been proposed to improve data quality. Lei et al. [7] estimated the missing values in incomplete sensor data by repeating two processes: selecting spatially correlated sensors and updating the training sensor dataset with the data selected from those sensors. This procedure assists in obtaining a more suitable neighbor sensor and refines the regression model by querying a record within the training sensor dataset. This method was effective when tested with various real databases, however it has not been applied in mobile environments.

Gill et al. [8] proposed a context-aware model-based technique for cleaning environmental data from sensors. They used geographical and meteorological datasets to create statistical models to train the system. Outliers are identified and discarded, then the partially cleaned data is analyzed by comparing the observed value with the predicted value for each attribute. Every time the observed value surpasses the error threshold in relation to the predicted data, the predicted data is used to replace the observed value. Although this method exhibits a high level of effectiveness, the authors did not consider the importance of in-network data cleaning capable of handling large volumes of trajectory data. Light-weight techniques are crucial as sensors deployed in mobile environments tend to work unattended with limited power and computational capacity [9].

Multivariate linear regression was employed by Kurasawa et al. [10] to predict the missing data by exploiting multiple attribute correlation. The method exploits spatio-temporal relationships and uses machine learning to build training datasets through the back end, sending data back to the sensor periodically. Zhang et al. [11] presented a method to select a single reliable sensor from their statistical model based on observed data and latent variables. These selected sensors' data was used to improve data quality in environmental monitoring. Although [10] and [11] were proposed for wireless environments, it still requires the presence of a sink and a back end for processing, making it unsuitable for infrastructure-less scenarios such as the IoT.

To address the above challenges in data cleaning, the development of a light-weight in-network method capable of handling large volumes of data is necessary.

III. Proposed Method

Our method utilizes the computation of two beta scores, the Speed Beta (β_s) and the Angle of Travel Beta (β_θ), combined with the spatial autocorrelation, local Moran's I, to select the set of candidate sensors for the process of data cleaning. The local Moran's I measures the spatial autocorrelation between a group of sensor nodes, while the beta computations compare the behavior of a group of sensors in relation to the node under analysis [15]. These computations quantify the risk involved in trusting a set of spatially correlated candidate nodes, as behavior similarities are directly proportional to their level of trustworthiness.

We consider a decentralized, in-network computational method. This means the detection of dirty data, the selection of the nodes to clean data, and the data cleaning are all done by each sensor. Each sensor node has an internal pre-process for detection of dirty data. Outliers are discarded and all dirty data is considered as missing data. Sensors are assumed to be (1) mobile, (2) to cooperate, and (3) to have *a priori* knowledge of the area where they are deployed. We assume that each sensor node has a unique identity and its sensing task takes place asynchronously.

The data exchange will only take place between any two nodes if these nodes are within transmission range. In other words, no data exchange will take place via multi-hop communication. Data exchanges follow the Round Robin Scheduling Technique as explained in [12]. With the data from its surrounding sensor nodes, each node should be able to approximate the missing values. However, finding a trustworthy sensor becomes a challenge.

Figure 1 shows an example of a MWSN where sensor nodes embedded into vehicles and/or devices carried by people move in a determined pattern. At first glance, a group of sensors sharing similar locations may appear to have similar behaviors. But this may not be the case. For example,

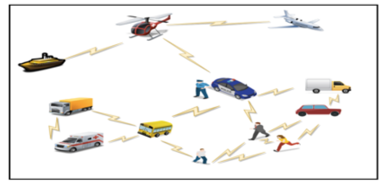


Fig. 1: Example of a mobile wireless sensor network

the speed at which the node with dirty data travels compared to the speed of the neighboring nodes may be a factor in determining the most trustworthy node. Moreover, selecting a single node can reduce the accuracy of the estimated values as the data of the selected node could be corrupted or imprecise. Selecting a set of candidate sensors can help to minimize the error during the estimation of the missing data. Once we have selected the set of most trustworthy sensors, the cleaning is performed using their data to calculate and replace the missing values.

A. Data Gathering

Our data cleaning process requires a time window T which is partitioned into two phases: sensing phase T_s and cleaning phase T_c; at the same time T is divided into t time instants. Individual time instants are referred to as t_j where $j = 1, 2, ..., t$.

1) Sensing Phase (T_s): When the sensor is performing the cleaning of its data, nodes with dirty data will select the fraction of time, T_e, within T_s to carry out the evaluation of candidate sensors. The time partition can be defined as: $T_e = t_d - k, ..., t_d - 2, t_d - 1, t_d, t_d + 1, t_d + 2, ..., t_d + p$, where t_d is the time instant where there is a missing value. The initial and final time instants in T_e are defined as $t_o = t_d - k$ and $t_f = t_d + p$ accordingly. At the end of every t_j sensors will update their internal tables with the sensed value x_{ij}, its position $(x, y)_{ij}$, angle of travel θ_{ij}, and average speed s_{ij}, where $i = 1, 2, ..., n$, and n is the total number of sensors in the network. To avoid error propagation, once a sensor node detects a dirty sample within its sensed values, the value is marked with a flag. No dirty data will be considered during the process of data cleaning. Table I shows the internal table for sensor i at each t_j during T_s.

TABLE I: Sensor i Internal Table for T_s

Time t_j	1	2	...	j
Position $(x, y)_{ij}$	$(x, y)_{i1}$	$(x, y)_{i2}$	...	$(x, y)_{ij}$
Sensed Value x_{ij}	x_{i1}	x_{i2}	...	x_{ij}
Average Speed s_{ij}	s_{i1}	s_{i2}	...	s_{ij}
Angle of Travel θ_{ij}	θ_{i1}	θ_{i2}	...	θ_{ij}
Dirty Flag f_{ij}	f_{i1}	f_{i2}	...	f_{ij}

2) Cleaning Phase (T_c): The cleaning phase begins after the sensing phase. In this period if sensors have data to be cleaned, at every time t_j, sensors will search for neighboring nodes within transmission range by broadcasting a cleaning status message (CSM) containing its identification and current location. After the CSM has been received, sensors estimate if there is enough time to send its data to neighboring sensors before the neighboring sensors move out of transmission range, as in [13]. If there is enough time to send the data to at least one node with dirty data, sensors will transmit their information in the form of data streams containing their internal tables. Once sensors have exchanged their data, in the event of future encounters, the received request will be ignored. To begin the evaluation, sensors will only evaluate neighboring candidates who were within one hop during t_d.

B. Beta-Based Candidate Reduction

Our beta-based candidate reduction chooses a set of sensors whom are the most correlated with respect to their trajectory behavior. We focus in the consideration of sensors within transmission range at time t_d. Candidate sensors with the least similar behavior (i.e., the sensors with beta values below or above the lower and upper pre-defined boundaries L_b and U_b) are discarded. The ideal candidate sensor would return a beta value of 1 for both betas, β_s and β_θ, meaning the sensor had the exact trajectory behavior as the sensor containing dirty data.

1) Trajectory Speed Beta: The similarity in trajectory speed variations along the sensors' trajectory path for a pair of nodes is measured by the speed beta, β_s, defined as:

$$\beta_s = \frac{cov(s_c, s_d)}{var(s_d)} \tag{1}$$

where s_c and s_d are the speed values of the candidate sensor during T_e and the sensor with dirty data respectively.

2) Angle of Travel Beta: The angle of travel represents the direction in which the sensor node is moving at the time instant the sensing is taking place. For a trajectory, the angle of travel beta, β_θ, is given by the equation:

$$\beta_\theta = \frac{cov(\theta_c, \theta_d)}{var(\theta_d)} \tag{2}$$

where θ_c and θ_d are the angle of travel values from a point of interest (relative to a given axis) of the candidate sensor and the sensor with dirty data respectively.

C. m-Candidate Selection: Spatial Autocorrelation

The similarity in trajectory behavior evaluated above does not provide information in regards to the spatial correlation among the sensor nodes under analysis. Since the environmental data exhibits two main features: time stability and space correlation [14], we employ local Moran's I to identify spatial clusters most spatially autocorrelated during time t_d. Local Moran's I identifies clustered sensor nodes with positive index values and outliers with negative index values. When evaluating the local Moran's I for the sensor containing dirty data, the missing value at time t_d is assigned the average of the sensed values collected in T_e. Local Moran's I is defined by:

$$I_i = \frac{\sum_{a=1}^{n} w_{ia}(z_a - \bar{z})(z_i - \bar{z})}{s^2 \sum_{a=1}^{n} w_{ia}} \tag{3}$$

where $s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$, z is the z-score of the sensor evaluated at time t_d, and weight $w_{ia} = \alpha^{d_{ia}}$ is an exponential function of the Euclidean distance d_{ia} between i and a. In this model, α is a hyper-parameter specified using cross-validation.

After local Moran's I is calculated for all sensor nodes, the sensor containing dirty data selects m sensors with the smallest $|I_d - I_{c_i}|$ values as the set of most trustful candidate sensors. I_d and I_{c_i} are the local Moran's I of the sensor containing the dirty data and the candidate sensors, c_i, respectively.

D. Diversified Trust Portfolio Distribution

Based on the technique proposed in [6], instead of devoting all our trust into one candidate sensor with a low beta, we diversify the trust throughout a set of candidate sensors. Our DTP approach delivers a trust portfolio by assigning weights, w_i, to the different candidate sensors, c_i, based on the risk to trust each sensors' data. To find the weight that needs to be assign to each candidate sensor, it is necessary to calculate relative observation error:

$$E(w_1, w_2, ..., w_m) = \frac{\sqrt{\sum_{j \in \{t.\}_o^f \setminus t_d} (\frac{x_{dj} - \sum_{i=1}^{m} x_{cj} \times w_{ci}}{x_{dj}})^2}}{|T_e| - 1} \tag{4}$$

where m is total number of candidate sensors, x_{dj} and x_{cj} are the sensed values of the sensor containing dirty data and the candidate sensor at time j respectively; and $|T_e| - 1$ represents the cardinality of the set of time instants selected for evaluation excluding t_d.

Since this is a continuous function and the domain of the function is compact, there is a minimum and a maximum. The diversified trust portfolio is generated by minimizing $E(w_1, w_2, ..., w_m)$ such that $w_i \leq 0$ for any $i = 1, 2, ..., m$.

E. Missing Value Approximation

After the weights have been distributed among the candidate sensors to minimize the error, the estimated value, R, is computed as follows:

$$R = \sum_{i=1}^{m} x_{c_i j} \times w_{c_i} \tag{5}$$

where $x_{c_i j}$ is the sensed value of the candidate sensor c_i at the dirty time j, and w_{c_i} is the weight assigned by the trust portfolio to the candidate sensor c_i.

Algorithm 1 Context-Aware m-Candidate Selection

Input: Time of missing value t_d, Internal tables for the sensor d containing dirty data and each candidate sensor at one-hop
Output: $\{C'\}$ Set containing m most trustworthy candidate sensors

```
for each c_i ∈ C do
    β_s ← Equation (1)
    β_θ ← Equation (2)
    if L_b ≤ β_s ≤ U_b and L_b ≤ β_θ ≤ U_b then
        C' ← c_i //Sensors that meet the trajectory behavior requirements
    end if
    I_d ← Equation (3) //Local Moran's I of sensor d
    for each c_l ∈ C' do
        I_l ← Equation (3)
        δ ← {c_l, I_l}
    end for
    C' ← m sensors with the smallest |I_d − I_{c_i}| values
end for
return C'
```

IV. PERFORMANCE EVALUATION

To evaluate the effectiveness of our approach, a number of environmental sensor nodes were placed in an area. The sensors select an initial sensor nodes' position, speed, and rest times, and continues to choose random destination points and speeds as per the steady state distributions of the random waypoint model outlined in [16]. Once the sensor arrives at the chosen point, it stops for a randomly selected time interval and continues to select another point and speed randomly.

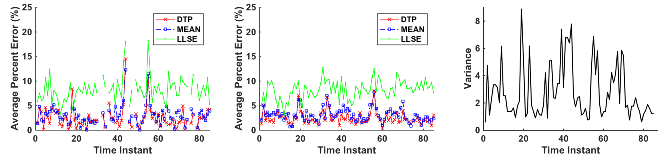
We utilized Bonnmotion as our mobility generator together with MATLAB to simulate the environment and test our proposed approach. The simulation employs 50 minute time windows divided into two phases: an initial sensing phase of 42.5 minutes, and a cleaning time phase of 7.5 minutes. At the same time each 50 minute time window is divided into 30 second time instants, in which sensors will begin to sense randomly. Our assessment is executed in two environments, outdoor and indoor.

Our predefined values for L_b, U_b, α and m are -0.50, 2.00, 0.10 and 5 respectively, which are hyper-parameters for our selected dataset determined using cross-validation. The node densities tested are 1 and 2.5 nodes per 100 m^2. The amount of sensors employed reach up to 2,250 nodes with up to 95,625 dirty samples. The efficiency of the proposed technique was evaluated by calculating the average percent error of all cleaned samples at each time instant. In addition, we calculated the cleaning level percentage for thresholds ranging from 0.50% to 10%. A sample is considered to be successfully cleaned if the absolute value of the percentage error between the value of reference and the calculated value falls below the specified error threshold. To contrast our results with existing techniques we selected Mean [17] and LLSE [18]. We specifically employed Mean's point, smooth and merge steps to detect sensor value outliers and correct missing data.

A. Intel Lab Data

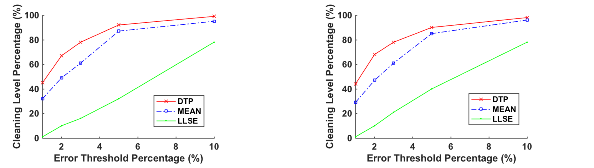
During our indoor environment simulation we used the Intel Indoor experiment by Intel Berkeley Research Lab [19], where 54 Mica2Dot sensors were deployed to collect various environmental data in an area of 120 m^2. The humidity data was employed to establish the values of reference to perform the simulation. The mean speed and transmission range employed by each sensor were 2 miles per hour and 5 meters respectively.

From Figure 2(c) it can be observed when the simulation was done in a low-density indoor environment, the variance of the data collected by all sensors fluctuates drastically. The discontinuities in the graphs occur when there were no missing values to be cleaned. Figures 2(a) and 2(b) show the average percent error of cleaned data by DTP stayed below Mean and LLSE when the simulation was performed with 20% and 50% of dirty data. Figures 3(a) and 3(b) confirm the performance of DTP as the cleaning level percentage reaches above the 98% under the 10% error threshold and above the 44% under 1% error threshold.



(a) 20% Dirty Data (b) 50% Dirty Data (c) Data Variance

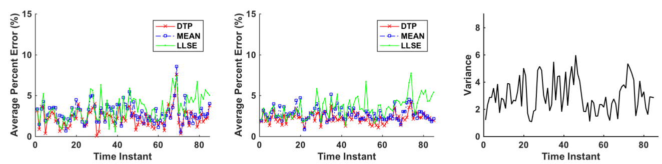
Fig. 2: 1 node per 100 m^2 for Intel Lab Data



(a) 20% Dirty Data (b) 50% Dirty Data

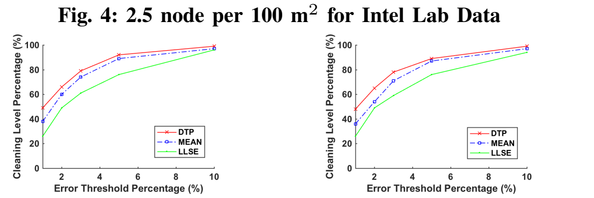
Fig. 3: Cleaning level percentage at 1 node density for Intel Lab Data

When the node density was increased to 2.5 nodes per 100 m^2, the variance of the data collected increased in stability, shown in Figure 4(c). Since LLSE takes into account the co-variance among each sensors' data, its performance improved with the increase of the data stability, shown in Figures 4(a) and 4(b). Figures 5(a) and 5(b) display our DTP approach kept a consistent performance over 99% of cleaning level percentage at 10% of error threshold and over 49% when tested for 1% error threshold.



(a) 20% Dirty Data (b) 50% Dirty Data (c) Data Variance

Fig. 4: 2.5 node per 100 m^2 for Intel Lab Data



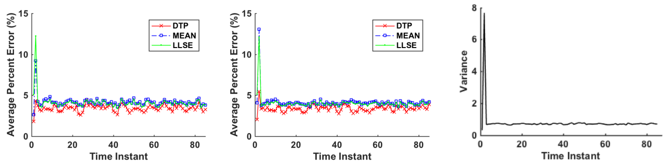
(a) 20% Dirty Data (b) 50% Dirty Data

Fig. 5: Cleaning level percentage at 2.5 node density for Intel Lab Data

B. Melbourne Weather Data

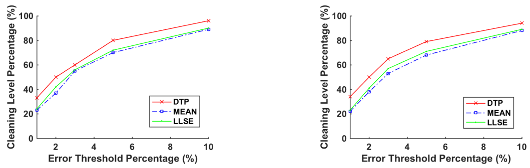
For our outdoor environment simulation we used the Melbourne dataset [20], where sensors collected temperature at 8 different locations from February 23-28, 2015. This collected data was used to generate the values of reference when evaluating the performance of our proposed method. In this assessment, sensors move in an area of 90,000 m^2 with a mean speed and transmission range of 20 miles per hour and 15 meters respectively.

From Figure 6(c), it can be observed that the variance of the data collected by all sensors display minimal fluctuations. The average percent error cleaned, shown in Figures 6(a) and 6(b), demonstrate DTP still maintains a lower average percentage error than Mean and LLSE methods. Figures 7(a) and 7(b) support the high efficiency of DTP by displaying over a 95% of cleaning level percentage for 10% error threshold and 49% for 1% error threshold.



(a) 20% Dirty Data (b) 50% Dirty Data (c) Data Variance

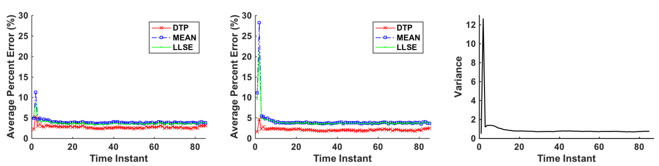
Fig. 6: 1 node per 100 m^2 for Melbourne Weather Data



(a) 20% Dirty Data (b) 50% Dirty Data

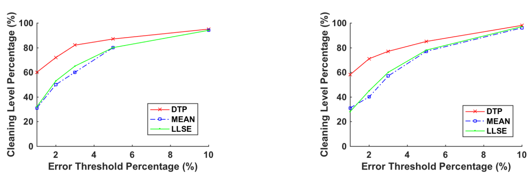
Fig. 7: Cleaning level percentage at 1 node density for Melbourne Weather Data

Similar to Figure 6(c), Figure 8(c) shows a high variance on the data initially, while the performance of Mean and LLSE are affected by this variance, DTP keeps a consistent high performance as seen in Figures 8(a) and 8(b). The cleaning level percentage was able to reach 98% under 1% error threshold and 59% for 10%. The consistent results of DTP are justified by its dependency on sensors' spatial autocorrelation and trajectory behavior rather than on the collected data only.



(a) 20% Dirty Data (b) 50% Dirty Data (c) Data Variance

Fig. 8: 2.5 node per 100 m^2 for Melbourne Weather Data



(a) 20% Dirty Data (b) 50% Dirty Data

Fig. 9: Cleaning level percentage at 2.5 node density for Melbourne Weather Data

V. Conclusions

Our unique Diversified Trust Portfolio (DTP) approach for cleaning data in MWSN has demonstrated to be an effective method that utilizes the spatio-temporal correlated data integrated with the analysis of contextual parameters of each sensors trajectories to analyze the relationships among sensors. This constitutes an effective online system for selecting a trustworthy set of sensors to help during the in-network data cleaning process. By selecting a set of sensors, DTP is able to find the combination of weights to more accurately estimate the data. This diversified trust technique reduces the risk involved in trusting a single sensor node's data. DTP demonstrated its outstanding capabilities to consistently achieve high data accuracy in comparison to two reputable data cleaning methods.

To the best of our knowledge, this is one of the first works that employs the risk premium calculations and portfolio diversification techniques in the modeling of trust in data accuracy for MWSN. Our future work will be focused on analyzing the application of other economic theories to model trust exploiting semantic correlated data.

Acknowledgements

We would like to acknowledge the National Science Foundation (NSF), the Department of Defense (DoD), the Army Research Office and the National Bureau of Science, Technology and Innovation of the Republic of Panama (SENACyT) for funding and supporting this project.

References

[1] Wireless Sensor Network Project Team. IEC Market Strategy Board. IoT: Wireless Sensor Networks (White Paper), July 2015.

[2] S. Sharma, R. K. Bansal, and S. Bansal, Issues and Challenges in Wireless Sensor Networks, International Conference on Machine Intelligence Research and Advancement, pp. 58-62, Katra, India, 2013.

[3] J. Park, M. Gofman, F. Wu, and Y.-H. Choi, Challenges of wireless sensor networks for Internet of thing applications, International Journal of Distributed Sensor Networks, vol. 12, no. 8, 2016.

[4] X. Chu, I. Ilyas, S. Krishnan and J. Wang, Data Cleaning, International Conference on Management of Data (SIGMOD 2016).

[5] A. Damodaran, "Estimating Risk Parameters" Available: http://people.stern.nyu.edu/adamodar/pdfiles/papers/beta.pdf

[6] H. Markowitz, Portfolio Selection, The Journal of Finance, vol. 7, no. 1, pp. 77-91, Mar. 1952.

[7] J. Lei, H. Bi, Y. Xia, J. Huang, and H. Bae, An in-network data cleaning approach for wireless sensor networks, Intelligent Automation & Soft Computing, vol. 22, no. 4, pp. 599 604, 2016.

[8] S. Gill, B. Lee, and E. Neto, Context aware model-based cleaning of data streams, Irish Signals and Systems Conference (ISSC), 2015.

[9] Z. Feng, and Z. Yanmin, "A Survey on Trajectory Data Mining: Techniques and Applications." IEEE Access 4, pp. 2056-2067, 2016.

[10] H. Kurasawa, H. Sato, A. Yamamoto, H. Kawasaki, M. Nakamura, Y. Fujii, H. Matsumura, "Missing sensor value estimation method for participatory sensing environment", International Conference on Pervasive Computing and Communications (PerCom), Budapest, Hungary, 2014 pp. 103-111.

[11] Y. Zhang, C. Szabo, and Q. Z. Sheng, An Estimation Maximization Based Approach for Finding Reliable Sensors in Environmental Sensing, International Conference on Parallel & Distributed Systems, 2015.

[12] R. Sharma, G. Jain, and S. Gupta, Enhanced Cluster-head selection using round robin technique in WSN, International Conference on Communication Networks (ICCN), Canada, 2015, pp. 37-42.

[13] L. Y. Njilla and N. Pissinou, Dynamics of data delivery in mobile ad-hoc networks: A bargaining game approach,Symposium on Computational Intelligence for Security and Defense Applications, 2015.

[14] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, Data loss and reconstruction in sensor networks, International Conference on Computer Communications, Turin, Italy, 2013, pp.1654 - 1662.

[15] M. Yan, Q. Tong, R. Wang, C. Luo, W. Land, Y. Gao, Y. Pan, "Outliers detection of cultivated land quality grade results based on spatial autocorrelation," International Conference on Agro-Geoinformatics, Tianjin, China, 2016, pp.1-5.

[16] W. Navidi, T. Camp, "Stationary Distributions for the Random Waypoint Mobility Model," ACM Transactions on Mobile Computing, vol. 3, no. 1, pp. 99 108, 2004.

[17] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, Declarative support for sensor data cleaning, Lecture Notes in Computer Science Pervasive Computing, pp. 83100, 2006.

[18] A. B Sharma. L. Golubchik, and R. Govindan, "Sensor Faults: Detection Methods and Prevalence in real-world datasets", ACM Transactions on Sensor Networks, vol.6, no, 3, pp. 1-39, 2010.

[19] Intel Indoor Test Data Available: http://db.csail.mit.edu/labdata/labdata.html Wireless Sensor Network Project Team. IEC Market Strategy Board. Internet of Things: Wireless Sensor Networks (White Paper), July 2015.

[20] Melbourne Data Available: https://data.melbourne.vic.gov.au/Environment/Sensor-readings-with-temperature-light-humidity-ev/ez6b-syvw