A Dynamic Trust Weight Allocation Technique for Data Reconstruction in Mobile Wireless Sensor Networks

Concepción Sánchez Alemán[1], Niki Pissinou[2], Sheila Alemany[2], Georges A. Kamhoua[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, gkamh001}@fiu.edu

Abstract—Data accuracy and low energy consumption in mobile wireless sensor networks (MWSN) are crucial attributes for real-time applications. Although there are many existing methods to reconstruct data for wireless sensor networks, there are few developed for highly mobile environments. We propose Dynamic Trust Weight Allocation Technique (DTWA), a novel in-network data reconstruction method that determines the trust level in the data accuracy of each candidate node by evaluating spatio-temporal correlations, trajectory behavior, quantity and quality of data, and the number of hops traveled by the received data from the original source. DTWA is capable of evaluating second-hand data when there is no first-hand data available and selecting second-hand data when this last is more accurate than the first-hand data. Our results demonstrate that data reconstructed using DTWA depicts significantly lower Root Mean Square Error (RMSE) compared to the IMC method when tested for both low and high incomplete dataset scenarios.

Index Terms—Dynamic trust allocation, data reconstruction, first-hand data, second-hand data, energy-aware, mobile wireless sensor networks

1 Introduction

Mobile wireless sensor networks (MWSN) have applications in various domains. A common application is in military systems and homeland security, where as an example, drones that are sent out on a mission are able to communicate with each other, exchange sensed data, and autonomously make decisions to accomplish their collective goals. Another example is in vehicular systems, where autonomous vehicles follow a common leader, separated by small inter-vehicle gaps, forming road trains to improve the vehicular flow and reduce accidents [1]. Besides the domains mentioned above, MWSN has applications in many other fields, such as environmental monitoring, health care, and the Internet of Things (IoT).

MWSN consist of a large number of sensor nodes deployed over a wide area, where sensors share information, collaborate to perform operations, and autonomously make decisions [2]. Having accurate data is important, as wrong data can lead to wrong decisions. The mobility of nodes in the network increases the difficulty in preserving data accuracy due to data collision [3], sensor isolation, and short-term connectivity of the network [4]. Figure 1 shows an example of a MWSN, where sensor nodes embedded into vehicles and/or devices carried by people communicate to exchange information, while moving in a determined pattern.

Additionally, sensor nodes generally have limited memory, low computational capacity, and are battery operated with restricted and irreplaceable power sources. Since sensor nodes in MWSN tend to collect and transmit data continuously, nodes are expected to function properly for long periods of time. Nevertheless, when their energy is exhausted, sensor nodes are no longer able to communicate or complete their functions and therefore propel the problem of energy consumption. It is known that in MWSN the maximum amount of energy is consumed by the communication process, which includes the transmission and reception of data. The second greatest energy consumer is computational operations in in-network processing. However, compared to the communication process, the computational operations expend much less energy [5]. To mitigate the negative effects these limitations impose on mobile networks, the development of a light-weight method that seeks to reduce energy consumption while providing highly accurate data is necessary.

Unlike current highly effective data reconstruction methods (where nodes train models and upload them to the sink for nodes to utilize this same model to reconstruct sensed values), our novel DTWA method is designed for real-time applications in mobile environments. In mobile environments, the delays produced by model training processes and the constant availability of a sink are unrealistic. The major contributions we have made with this work can be summarized as follows:

- We proposed a light-weight low-complexity data reconstruction method, ideal for mobile environments where energy preservation is imperative.
- We presented a novel data reconstruction method to identify a set of sensors to help during the prediction of missing values. Our method dynamically assigns weights for trust in data accuracy to first-hand and second-hand data in highly incomplete datasets.
- We conducted extensive experiments on real world datasets to verify the efficiency and accuracy of our

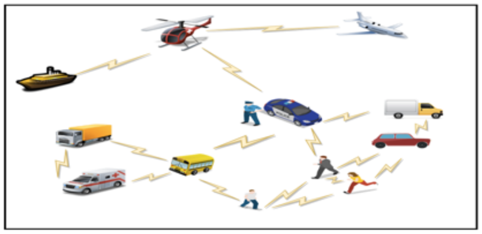


Fig. 1: Example of a Mobile Wireless Sensor Network

proposed method (DTWA). It was shown that our proposed method performs better than IMC method by achieving consistent high data accuracy when tested for highly incomplete datasets.

The related work can be reviewed in Section 2. The assumptions and methodology can be found in Section 3. Section 4 provides our experimental results, and Section 5 discusses conclusions and future work.

2 RELATED WORK

Many techniques have been proposed for predicting missing data in wireless sensor networks. Kong et al. [6] proposed Environmental Space Improved Compressive Sensing (ESTICS). ESTICS employed compressive sensing while combining spatio-temporal correlation to reconstruct complete information from a portion of data. Gill et al. [7] proposed a context-aware model-based technique for cleaning environmental data from sensor nodes. Using geographical and meteorological datasets, they created statistical models for system training. Then, data was analyzed by comparing the observed value with the predicted value. When the observed value exceeded the error threshold in relation to the predicted data, the predicted data was used to replace the observed value.

Lei et al. [8] estimated the missing values in incomplete sensor data by selecting spatially correlated sensors and updating the training sensor dataset with the data selected. This procedure assisted in obtaining a more suitable candidate sensor and refined the regression model by querying a record within the training sensor dataset. In [9], Zhang et al. presented a reliability-based technique, where the reliability of each sensor is adapted according to its performance. At every iteration the consistency is updated based on the difference of the prediction made and the real sensed value. A method for selecting a reliable sensor was presented by Zhang et al. [10] using a statistical model based on sensed data and latent variables, such as the sensors' faulty state. Kurasawa et al. [11] proposed using multivariate linear regression to predict the missing data using multiple attribute correlation. The method analyzed spatio-temporal relationships and used machine learning to build training datasets through the back-end, sending data back to the sensor periodically.

Although [6], [7], [10], and [11] exhibited high levels of effectiveness, the importance of models that perform in-network computations and capable of handling large volumes of trajectory data is not considered. Moreover, the authors did not take into consideration the energy consumption to perform the heavy computations that most of these models propose. Light-weight techniques are crucial, as sensors deployed in mobile environments tend to work unattended with limited power and computational capacity. Despite the fact that [6], [8], and [10] were proposed for wireless environments, and [7] and [11] for mobile networks, they relied on the presence of a sink and/or a back-end for their data processing. MWSN tend to present delays related to continuous data transmissions and delays related to data processing even when there is no data to reconstruct, as in [9]. In addition, authors in [6] did not consider network failure due to sink isolation produced by the sensors near sink that deplete their energy faster due to heavy traffic, creating energy holes and network failure.

To address these problems, we have designed an effective, light-weight method for mobile environments, where the network topology is temporary. Our dynamic system assigns trust scores for data accuracy, based on each candidate sensor node's current condition. Data training and additional transmissions-related delays, as well as the energy expenditure, are reduced in this real-time approach.

3 ASSUMPTIONS AND METHODOLOGY

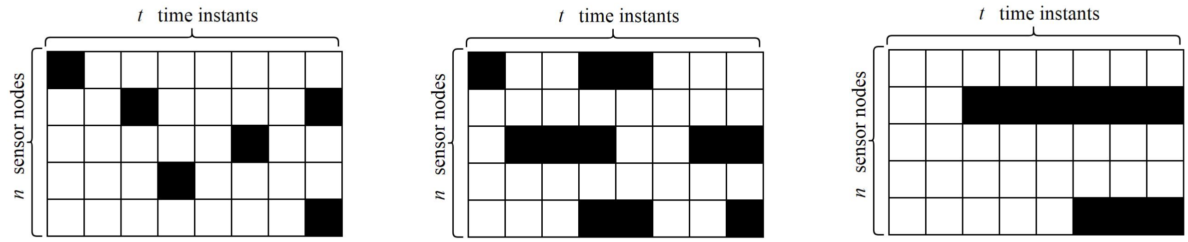
It is assumed that each sensor node has an internal preprocess for detection and elimination of dirty data, resulting in missing data. Data that is not detected as missing is considered valid and to be used for reconstruction. Our proposed method assumes a decentralized, in-network computational method. In other words, the selection of the candidate sensor(s) to perform the reconstruction, and the reconstruction itself are done by each node. The sensing task takes place asynchronously, and the sensors are considered to be mobile, cooperative, and to have *a priori* knowledge of the area. The data exchange can take place between any two nodes via multi-hop communication and follow the Round Robin Scheduling Technique, as described in [12].

3.1 Data Loss Patterns

It is known that MWSN exhibit different types of data loss patterns [6]. The most commonly employed data loss patterns in real-time applications are:

- **Element random loss pattern:** Elements are dropped independently and randomly during the transmission. This can be as a result of noise and collision.
- **Element frequent loss in a row pattern:** Sensed data from a single node has higher probabilities of loss. This can be produced by unreliable links.
- **Successive element loss in a row:** Sensor stops sensing at some point in time and produces no more sensed values until the end of the simulation. This could be a result of sensors losing energy or malfunctioning.

DTWA is designed to reconstruct real-life data loss patterns which include a combination of all the aforementioned patterns. Figure 2 gives a graphic view of the described data loss patterns in MWSN.



(a) Element Random Loss

(b) Element Frequent Loss in a Row

(c) Successive Element Loss in a Row

Fig. 2: Data Loss Patterns in Mobile Wireless Sensor Networks

3.2 Data Gathering

In our DTWA approach, data is gathered and reconstructed in a time window T, which is partitioned into (1) sensing phase T_s and (2) reconstruction phase T_r; at the same time T is divided into t time instants. Each time instant is appointed as t_j, where $j = 1, 2, ..., t$.

3.2.1 Sensing Phase (T_s)

Each node will sense and collect data. At the end of every t_j, sensors will update their internal tables, as shown in Table I, using the sensed value (x_{ij}), its position (l_{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. When the node detects a dirty sample within its sensed values, the value is marked with a flag as missing. When a sensor is reconstructing its data, it will select the sub-partition of time, T_e (evaluation time), within T_s to carry out the evaluation of candidate sensors. The sub-partition of time can be described as: $T_e = t_m - k, ..., t_m - 2, t_m - 1, t_m, t_m + 1, t_m + 2, ..., t_m + p$, where t_m is the time instant where there is a missing value. The initial and final time instants in T_e are $t_o = t_m - k$ and $t_f = t_m + p$, respectively. $|T_e|$ is user-defined and, k and p are dynamic values based on the position of t_m.

TABLE 1: Sensor i Internal Table for T_s

Time t_j	1	2	...	j
Location l_{ij}	l_{i1}	l_{i2}	...	l_{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}
Missing Flag f_{ij}	f_{i1}	f_{i2}	...	f_{ij}

3.2.2 Reconstruction Phase (T_r)

In this period, if sensors have data to be reconstructed, at every time t_j, sensors will broadcast a data request message containing its identification and current location. Receiving sensors estimate if there is enough time to send their data to the requesting sensor before the two move out of transmission range. Once sensors confirm the feasibility of the transmission, sensors will transmit their information together with the information from any other node received during T_r.

3.3 Dynamic Trust Weight Allocation (DTWA)

Our Dynamic Trust Weight Allocation (DTWA) method quantifies the level of trust in data accuracy for each of the candidate sensors without the usage of predefined thresholds. The DTWA scheme revolves around common factors and conditions faced by each node. When a sensor node contains a missing value, it must evaluate influencing factors to identify which node(s) data accuracy to trust. DTWA describes trustworthy nodes as nodes containing the highest quantity of high-quality, spatio-temporally correlated data with significant resemblance in trajectory behavior in relation to the evaluating node. To compute the total trust (τ), we evaluate the following parameters:

- **Confidence level (ϕ_c):** To evaluate the trustworthiness of the accuracy of the data provided by a node, taking into consideration the number of correct observations provided.
- **Spatio-Temporal Closeness (ϕ_x):** To prioritize first-hand data yet, consider second-hand data while taking into account the Euclidean distance between two nodes.
- **Pearsons Correlation Coefficient (ρ):** To quantify the strength of a linear relationship between the collected sensed values for the pair of sensors.
- **Normalized Speed Beta (β_s):** To quantify the similarity in trajectory speed variations for a pair of sensors.
- **Normalized Angle of Travel Beta (β_θ):** To measure the similarity in direction of a pair of sensors. The angle is calculated at the time instant the sensing is occurring, and are values from a point of interest relative to a given axis.

After all parameters have been evaluated, the total trust in data accuracy of an individual node can be computed. Figure 3 shows an example of a MWSN, in this scenario, all sensor nodes may have provided their sensed data and depicted a strong correlation among their sensed values. Although nodes may receive the data of each candidate node via one-hop communication and share similar locations/trajectories at time t_1, the change of speed after time t_1 can be the determining factor if a node is attempting reconstruction of data after time t_1.

3.3.1 Trust Parameters Evaluation

The degree of trust evaluation is an adaptive mechanism to assess the certainty in data accuracy. Each evaluating node performs this evaluation for each candidate sensor with the data provided. To carry out this evaluation, we consider the following parameters:

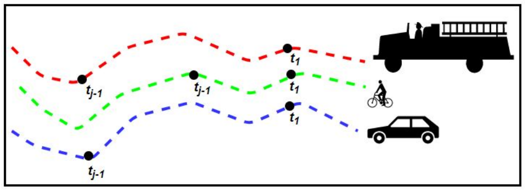


Fig. 3: Trajectory Behavior Example

A. Confidence Level

The confidence level quantifies the number of samples provided versus the number of missing samples during the evaluation time T_e. This value ranges from [0,1], where 0 indicates that the node did not provide any valid sensed sample and 1 represents that the node provided all valid samples and no missing samples. The confidence level can be calculated using a modified formula from [13], and is given by:

$$\phi_c = 1 - \sqrt{\frac{12 \times \delta \times \xi}{(\delta + \xi)^2(\delta + \xi + 1)}} \tag{1}$$

where δ is the number of time instances containing complete, accurate data, and ξ is the number of time instances containing missing data for the candidate sensor, y.

B. Spatio-Temporal Closeness

The spatio-temporal closeness parameter quantifies how close two sensor nodes were in t_m. It considers not only the distance among the two nodes, but whether the data was received via one-hop or multi-hop. In other words, if the data received is first-hand or second-hand. The spatio-temporal closeness can be determined as shown below:

$$\phi_x = \begin{cases} (\alpha)^d & \text{if received via one-hop} \\ (\gamma)^d & \text{if received via multi-hop} \end{cases} \tag{2}$$

where $\alpha > \gamma$. They are hyper-parameters that are dependent on whether data was received via one-hop or multi-hop and are determined using cross validation. d is the Euclidean distance between the sensors at the time to be reconstructed.

C. Pearson's Correlation Coefficient

Pearson's correlation coefficient [14] calculates the correlation between the sensed values during the evaluation period T_e. For a pair of nodes, a positive correlation indicates sensed values are directly proportional. A negative correlation indicates sensed values are inversely proportional. The closer Pearson correlation coefficient approaches 1, the stronger the positive linear relationship. DTWA considers only positive Pearson's correlation coefficients ranging from [0,1] and negative coefficients ranging from [-1,0) are assigned 0 at evaluation time. Namely, sensors with inverse relationships are given no trust. The Pearson's Correlation coefficient can be computed using the following formula:

$$\rho = \frac{cov(x, y)}{\sigma_x \times \sigma_y} \tag{3}$$

where x and y are the sensed values during T_e of the sensor with data to be reconstructed and a candidate sensor, respectively. σ_x and σ_y are the standard deviations for x and y, respectively.

Trajectory Behavior Similarity

To compare the trajectory behavior between the baseline node and each candidate node, we utilize Beta analysis [15], [16]. Specifically, we compute Speed Beta (β_s), and Angle of Travel Beta (β_θ). Betas quantify how similar or dissimilar each node behaves in relation to the evaluating node. The ideal candidate sensor would return a beta value of 1 for both betas, β_s and β_θ, meaning that both sensors had the exact trajectory behavior. The similarity in trajectory variations along the sensors' trajectory path for a pair of sensors is computed as:

$$\beta_u = \frac{cov(u_c, u_m)}{var(u_m)} \tag{4}$$

where u_c and u_m are the speed or angle of travel values of the candidate sensor during T_e and the sensor with missing data, respectively. The beta values are normalized to be transformed into weight coefficients. The new normalized values represent the probability that the value could appear in the given historical data. To obtain the normalized beta values, β'_u, we compute the following formula:

$$\beta'_u = \frac{\beta_u - min(\beta_u)}{max(\beta_u) - min(\beta_u)} \tag{5}$$

where β_u contains the non-normalized beta values. $min(\beta_u)$ and $max(\beta_u)$ are the minimum/maximum values of all the collected beta values that a sensor contains.

3.3.2 Total Trust Computation

Combining the five parameters described above results in our total trust formula. The total trust is computed as follows:

$$\tau = \frac{(w_1\phi_c) + (w_2\phi_x) + (w_3\rho) + (w_4\beta'_s) + (w_5\beta'_\theta)}{w_1 + w_2 + w_3 + w_4 + w_5} \tag{6}$$

where w_1, w_2, w_3, w_4 and w_5 are user-defined weights assigned to each parameter dependent on each node preferences. Once the sensor containing missing data has computed the total trust value (τ) for each candidate sensor, it will select the sensor(s) with the highest value (τ), as shown in Algorithm 1. Finally, the selected sensors' information is employed to approximate the missing data.

3.3.3 Missing Value Approximation

If at least n_c candidate nodes with $\phi_c = 1$ exist, the diversified trust portfolio and data predictions are made using Equations (7) and (8). If that does not exist, then a linear a regression is employed. Equations (9) and (10) are employed for the prediction. Where n_c is a user-defined parameter that specifies the minimum quantity of candidate sensors are preferred to employ the diversified trust portfolio technique.

A. $\exists\, n_c$ Candidate Sensors with $\phi_c = 1$

Based in the portfolio selection technique proposed in [17], instead selecting one single candidate node, we diversify the trust throughout a set of candidate sensors. Choosing a set of candidate sensors can help to minimize the error between the predicted data and the real values. The diversified trust portfolio assigns weights to multiple candidate sensors based on the risk to trust each sensors' data [16]. To find the weight to be assigned to each candidate sensor, it is necessary to minimize the relative observation error:

$$E(w_1, w_2, ..., w_{n_c}) = \frac{\sqrt{\sum_{j\in\{t.\}_o^f \setminus t_m} \left(\frac{x_{dj} - \sum_{i=1}^{n_c} x_{cj} \times w_{ci}}{x_{dj}}\right)^2}}{|T_e| - 1} \tag{7}$$

where n_c is total number of candidate sensors, x_{dj} and x_{cj} are the sensed values of the sensor missing data and the candidate sensor at time j, respectively; w_{ci} is the weight variable to be minimized and $|T_e| - 1$ represents the cardinality of the set of time instants selected for evaluation, excluding t_m. Noting that t_m is the time containing the missing data we are approximating.

This is a continuous function with a compact domain, so there is a guaranteed minimum and maximum. To find the combination of weights that will depict the highest precision when approximating the missing value, the diversified trust portfolio is generated by minimizing $E(w_1, w_2, ..., w_{n_c})$ such that $w_i \geq 0$ for any $i = 1, 2, ..., n_c$. After the weights have been spread out among the candidate sensors to minimize the error, the estimated value, R, is calculated as follows:

$$R = \sum_{i=1}^{n_c} x_{c_i m} \times w_{c_i} \tag{8}$$

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

B. $\nexists\, n_c$ Candidate Sensors with $\phi_c = 1$

If there are no sensors that do not contain an missing values in T_e, then we select one candidate sensor with the highest trust value τ. To approximate the value at the time with missing data, we calculate the r-correlation coefficient [18] as below:

$$r = \frac{\sum_{i=t_o}^{t_f} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=t_o}^{t_f} (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{9}$$

where $b_1 = r\frac{\sigma_x}{\sigma_y}$, and $b_0 = \bar{y} - (b_1 \times \bar{x})$. The values for x and y are the sensed values for the sensor containing missing data and the candidate sensor, respectively. $\bar{x}$ and $\bar{y}$ are the mean of the sensed values, and y_m is the sensed value of the selected candidate sensor at the time where data is missing. Then, the value is approximated as:

$$R = b_0 + (b_1 \times y_m) \tag{10}$$

4 Experimental Results

To evaluate the performance of our method, a number of mobile sensor nodes were placed in an area. The sensors followed the steady state random waypoint mobility model [19]. We utilized the Bonnmotion Mobility Scenario Generation and Analysis Tool to generate sensor mobility and MATLAB to simulate our testing environment and perform the data cleaning computations. In this assessment, sensors move with a mean speed and transmission range of 20 miles per hour and 15 meters, respectively. For our 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. We employed a 50-minute time window divided into two phases: sensing phase of 42.5 minutes and reconstruction phase of 7.5 minutes. Every time instant t_j had a duration of 30 seconds and sensing occurred asynchronously during every time instant of the sensing phase.

The values for α and γ are 1 and 0.01, respectively, and are the hyper-parameters for our selected dataset evaluated using cross-validation. The weight parameters w_1, w_2, w_3 w_4 and w_5 employed to calculate total trust are equal to 1, as we consider all parameters to be equally important. $|T_e|$ and n_c are 10 and 5, respectively. The simulation was performed with 900, 450 and 250 nodes in an area of 90,000 m^2. The percentage of missing samples representing the amount of data reconstructed for our simulations were 20%, 50%, and 70%. The performance of the proposed technique was evaluated by calculating the Root Mean Square Error (RMSE) of all reconstructed samples at each time instant. RMSE is a quadratic scoring rule that measures the average magnitude of the error and is beneficial in penalizing large errors. To evaluate the effectiveness of our results, we compared the RMSE values at each time instant with the effective recent technique, IMC [9].

Table 2 displays the average percentages of missing data

Algorithm 1 Candidate(s) Selection for DTWA

Input: Time of missing value t_m, Internal tables for: the sensor d and each candidate sensor c_i, user-defined potential number of candidate sensors n_c
Output: The set of trustworthy candidate sensors C'

```
for each c_i ∈ C do
    φ_c(c_i) ← Equation (1)                      // Confidence Level
    if φ_c(c_i) = 1 then
        C_complete ← c_i      // Set of sensors with complete data in T_e
    end if
    φ_x(c_i) ← Equation (2)              // Spatio-Temporal Closeness
    ρ(c_i) ← Equation (3)                 // Sensed Value Correlation
    β_s(c_i) ← Equation (4)                              // Speed Beta
    β'_s(c_i) ← Equation (5)                  // Normalized Speed Beta
    β_θ(c_i) ← Equation (4)                    // Angle of Travel Beta
    β'_θ(c_i) ← Equation (5)        // Normalized Angle of Travel Beta
    τ(c_i) ← Equation (6)                         // Total Trust Value
end for
if |C_complete| ≥ n_c then
    c_τmaxset ← Set of sensors with greatest τ values from C_complete
else
    c_τmaxset ← Single sensor with greatest τ value in C
end if
return C'(τmaxset) // Return sensor(s) corresponding to τmaxset
```

TABLE 2: Average Data Loss and Collection Statistics

Sensor Count n	250	450	900
First-Hand	42.07%	31.42%	36.71%
Second-Hand	57.93%	68.58%	63.29%
Element Random Loss	21.64%	22.51%	22.09%
Element Frequent Loss in a Row	58.20%	56.95%	66.51%
Successive Element Loss in a Row	20.16%	20.54%	11.40%

reconstructed utilizing first-hand and second-hand data throughout our simulation. It also presents the average distribution of missing data generated by each of the data loss patterns simulated to assess our method. As element frequent loss in a row is the most common data loss pattern in MWSN, up to 66.51% of the missing data was lost using that data loss pattern. Up to 22.51% and 20.54% of the missing data was generated using the element random loss pattern and the successive element loss in a row pattern, respectively.

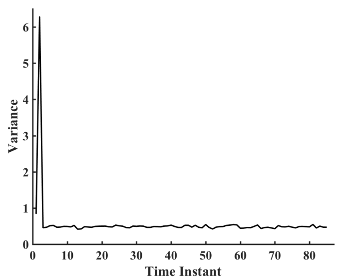
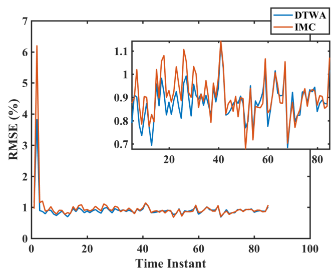
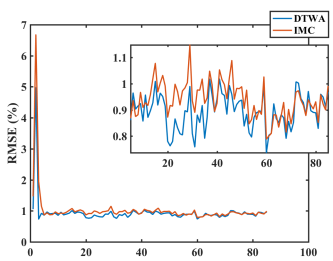


Fig. 4: Data Variance of Data Collected by Sensors

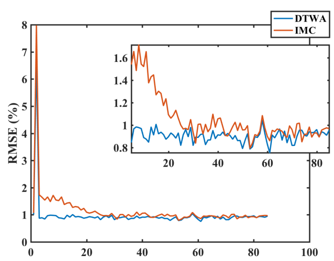
Figure 4 shows a sample of the variance of the data collected by the 450 sensors' simulation. The variance of the data collected in all of our simulations had similar behavior with a spiked variance in the initial few time instances and a constant lower variance once all sensors have collected a few values. This is due to the steady state behavior of our sensors in our simulation. From the behavior of the data collected and the RMSE at each simulation, it is inferred that the RMSE and the variance are directly proportional.



(a) 20% Missing Data



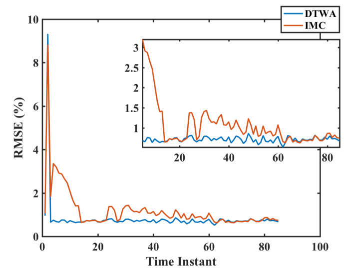
(b) 50% Missing Data



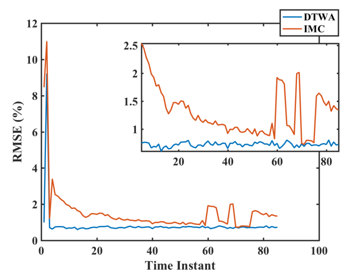
(c) 70% Missing Data

Fig. 5: 250 sensors collecting Melbourne Data

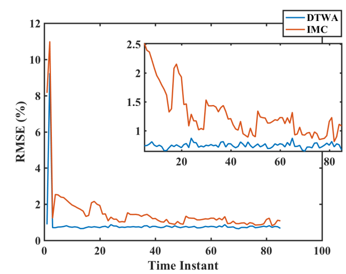
The RMSE of the values reconstructed by DTWA contrasted against IMC for 250 sensors' simulation is shown in Figure 5. When tested for 20% of incomplete data, IMC shows to be highly competitive compared to DTWA. However, when tested for 70%, it is evident that it is much harder for IMC to learn which sensor is most trustworthy, while DTWA is resilient to high quantities of missing data. This is due to the fact that IMC only considers spatio-temporal and sensed value correlations while DTWA considers additional parameters that influence the data reconstruction process in real-life applications. Figures 5, 6, and 7 contain insets to more clearly see the RMSE at each time instant after the initial spike in variance.



(a) 20% Missing Data



(b) 50% Missing Data



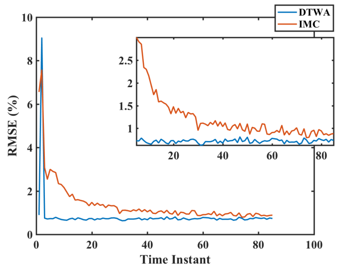
(c) 70% Missing Data

Fig. 6: 450 sensors collecting Melbourne Data

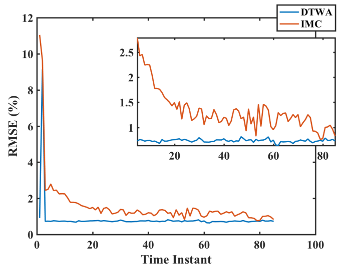
The increase in the number of sensors complicates the selection of trustworthy sensor(s) in data reconstruction techniques. Figures 6 and 7 demonstrate that DTWA performed better compared to IMC by predicting the missing data with higher precision, even when the number of sensors and their interactions were increased. It was easier for DTWA to approximate the values because the prediction of the missing data depends on the evaluation of different behavior parameters in both first- and second-hand data. The likelihood of finding an optimal sensor with the most valid data and similar trajectory behaviors increases as the number of sensors increases. Utilizing the trust parameters evaluation, DTWA selected second-hand data in up to 68% of its predictions, as shown in Table 2. IMC shows spikes in their RMSE throughout the simulations because, at particular time instances, the amount of available valid data is minimal. DTWA depicts a low and stable RMSE regardless of how many missing values there were at a specific point in time.

5 Conclusions and Future Work

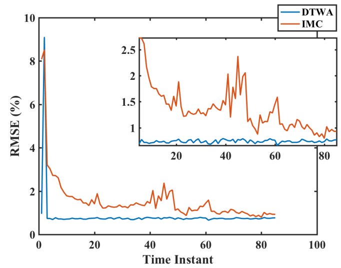
DTWA is a novel and effective light-weight in-network technique designed to reconstruct highly incomplete datasets in mobile environments. Our scheme quantifies the level of trust in data accuracy for each candidate sensor and revolves around common factors and conditions faced by each node in real-world applications. DTWA's accuracy is



(a) 20% Missing Data



(b) 50% Missing Data



(c) 70% Missing Data

Fig. 7: 900 sensors collecting Melbourne Data

obtained from the selection of sensor(s) with the highest quantity of high-quality, spatio-temporally correlated data and with significant resemblance in trajectory behavior. DTWA can be easily tailored to different scenarios in MWSN, and the flexibility of the modification of weights given to each attribute can contribute to meet specific user requirements in diverse scenarios. The dynamic adaptive features of DTWA makeS it suitable for evaluating the certainty of data accuracy for neighboring sensor nodes in scenarios with large quantities of missing data and sensor count, such as in IoT.

When compared to IMC, another useful light-weight algorithm, DTWA demonstrated its outstanding capabilities to consistently achieve high data accuracy with vast quantities of missing data. Since IMC showed to outperform LLSE and the Mean methods [21], [22], and DTWA outperformed IMC, we can derive that DTWA can achieve better data accuracy than the two well-known methods, LLSE and the Mean. Contrary to various current methods, the evaluation of trust in DTWA is not affected by the past interactions, which addresses the newcomer problem. DTWA is also an energy-aware method, as sensors will only compute predictions when there is missing data. Our future work will focus on studying the application of low complexity future-data prediction models for highly mobile environments.

Acknowledgments

We would like to acknowledge the support of the National Science Foundation REU and RET under grant CNS-1560134, the Department of Defense, and the National Bureau of Science, Technology and Innovation of the Republic of Panama (SENACyT).

References

[1] L. Terruzzi, R. Colombo, and M. Segata, "Poster: On the effects of cooperative platooning on traffic shock waves," *IEEE Vehicular Networking Conference (VNC)*, 2017.

[2] H.-D. Ma, "Internet of things: Objectives and scientific challenges," *Journal of Computer science and Technology*, vol. 26, no. 6, pp. 919–924, 2011.

[3] S. Sharma, R. K. Bansal, and S. Bansal, "Issues and challenges in wireless sensor networks," in *International Conference on Machine Intelligence and Research Advancement (ICMIRA)*. IEEE, 2013, pp. 58–62.

[4] J. Park, M. Gofman, F. Wu, and Y.-H. Choi, "Challenges of wireless sensor networks for internet of thing applications," in *International Journal of Distributed Sensor Networks*. SAGE Publications Sage UK: London, England, 2016.

[5] A. S. H. Abdul-Qawy, N. M. Almurisi, A. P. Kumar, and T. Srinivasulu, "Major energy dissipation sources in the iot-based wireless networks," *International Journal of Electronics, Electrical and Computational System*, vol. 6, no. 9, pp. 155–161, 2017.

[6] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *International Conference on Computer Communications (INFOCOM)*. IEEE, 2013, pp. 1654–1662.

[7] S. Gill, B. Lee, and E. Neto, "Context aware model-based cleaning of data streams," in *Signals and Systems Conference (ISSC)*. IEEE, 2015, pp. 1–6.

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

[9] Y. Zhang, C. Szabo, and Q. Z. Sheng, "Cleaning environmental sensing data streams based on individual sensor reliability," in *International Conference on Web Information Systems Engineering*. Springer, 2014, pp. 405–414.

[10] Y. Zhang, C. Szabo, and Q. Sheng, "An estimation maximization based approach for finding reliable sensors in environmental sensing," in *International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2015, pp. 190–197.

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

[12] R. Sharma, G. Jain, and S. Gupta, "Enhanced cluster-head selection using round robin technique in wsn," in *International Conference on Communication Networks (ICCN)*. IEEE, 2015, pp. 37–42.

[13] A. M. Shabut, K. P. Dahal, S. K. Bista, and I. U. Awan, "Recommendation based trust model with an effective defence scheme for manets," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2101–2115, 2015.

[14] K. Pearson, *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[15] A. Damodaran, "Estimating risk parameters." [Online]. Available: http://hdl.handle.net/2451/26789

[16] C. Sanchez Aleman, N. Pissinou, S. Alemany, K. Boroojeni, J. Miller, and Z. Ding, "Context-aware data cleaning for mobile wireless sensor networks: A diversified trust approach," *International Conference on Computing, Networking and Communications*, in press.

[17] H. Markowitz, "Portfolio selection," *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[18] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2012, vol. 821.

[19] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

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

[21] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 3, p. 23, 2010.

[22] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, "Declarative support for sensor data cleaning," in *International Conference on Pervasive Computing*. Springer, 2006, pp. 83–100.