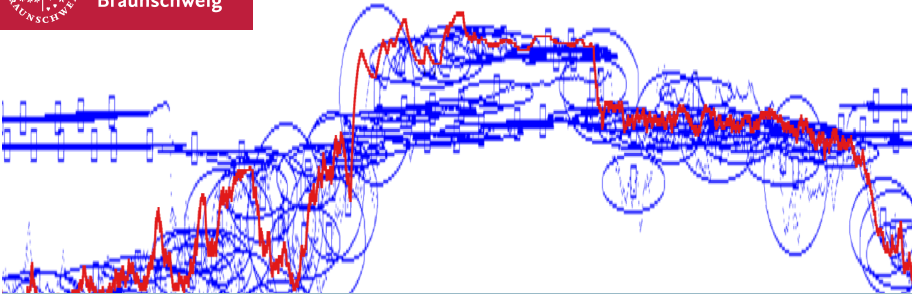
Institute of Operating Systems
and Computer Networks

Technische
Universität
Braunschweig



A Clustering-Based Characteristic Model for Unreliable Sensor Network Data

WF-IoT 2015

Ulf Kulau, Tobias Breuer and Lars Wolf, December 14, 2015

Technische Universität Braunschweig, IBR

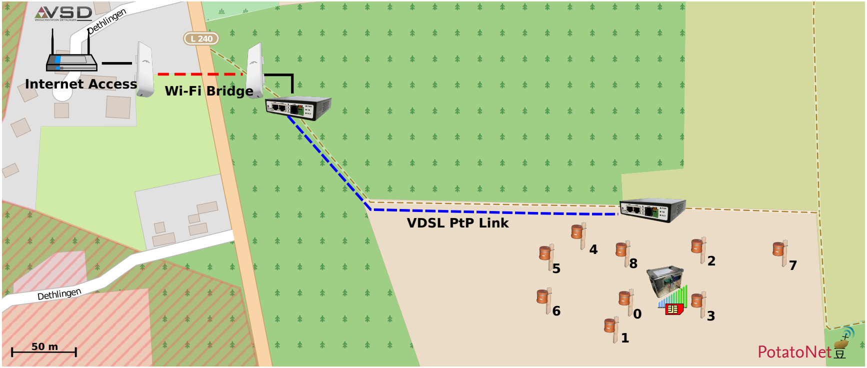
Introduction – Background

Many WSN/IoT applications deployed in challenging areas

- Harsh environmental conditions
  $\rightarrow$ Reliability of nodes decreases
  $\rightarrow$ Correctness of data-collection might be affected

Introduction – PotatoNet

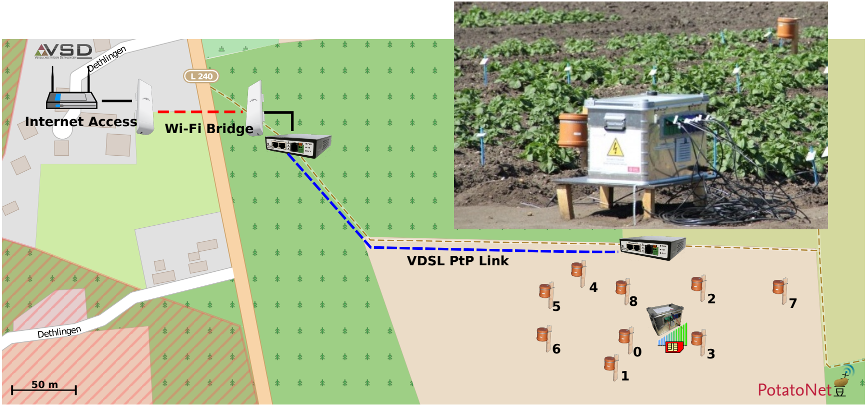
Outdoor WSN testbed – Central box and field-nodes



https://www.ibr.cs.tu-bs.de/projects/potatonet/

Introduction – PotatoNet

Outdoor WSN testbed – Central box and field-nodes



https://www.ibr.cs.tu-bs.de/projects/potatonet/

Introduction – Proposed Approach

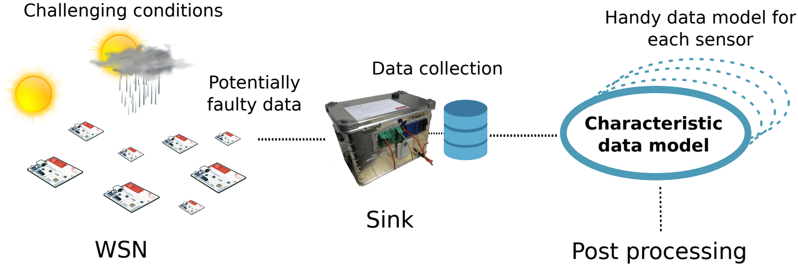
Convenient data handling – System overview

- Collection of *many* (potentially faulty) data elements at the sink

Introduction – Proposed Approach

Convenient data handling – System overview

- Collection of *many* (potentially faulty) data elements at the sink
  $\rightarrow$ Generate a more handy data model for processing and storage

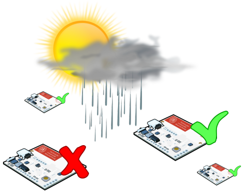


Data model – Motivation

Characteristics of data can be used to...

- detect errors

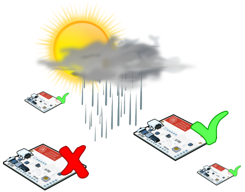
  → Unreliable sensing (challenging environment, undervolting, ...)

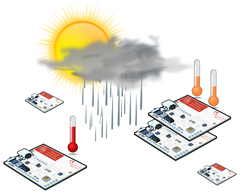


Data model – Motivation

Characteristics of data can be used to...

- detect errors
  $\rightarrow$ Unreliable sensing (challenging environment, undervolting, ...)
- detect redundant sensing
  $\rightarrow$ Neighboring nodes are potentially redundant (shared sensing)





Data model – Motivation

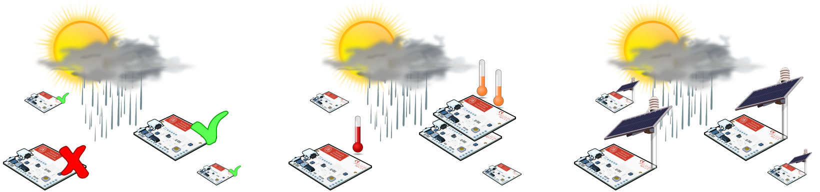
Characteristics of data can be used to...

- detect errors

  → Unreliable sensing (challenging environment, undervolting, ...)
- detect redundant sensing

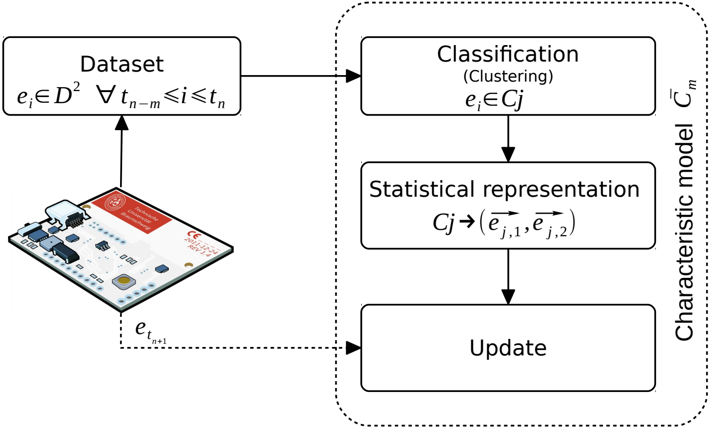
  → Neighboring nodes are potentially redundant (shared sensing)
- predict further system states

  → Energy efficiency/budget can be predicted to schedule tasks



Course of action

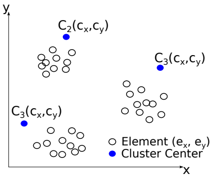
Generation of the characteristic data model



Data model – Classification of data

Classification using k-means algorithm

1. Generate potential clusters C_j with a random center $C_j(c_x, c_y)$.

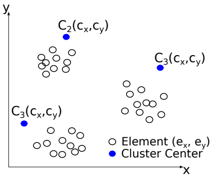


Step 1

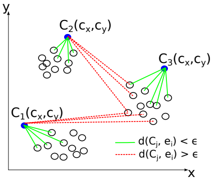
Data model – Classification of data

Classification using k-means algorithm

1. Generate potential clusters C_j with a random center $C_j(c_x, c_y)$.
2. Calculate the distance of each element $e_i \in \mathbb{D}^2$ to the clusters C_j



Step 1

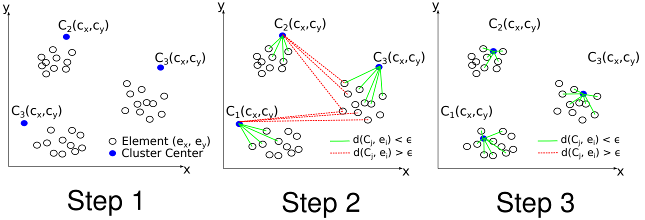


Step 2

Data model – Classification of data

Classification using k-means algorithm

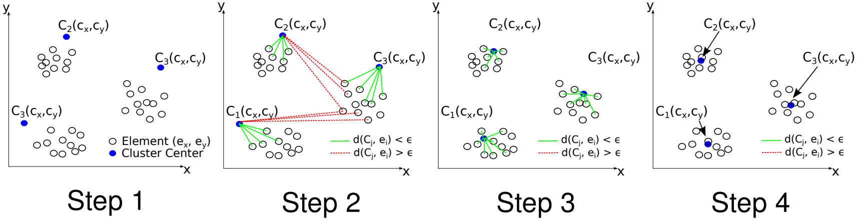
1. Generate potential clusters C_j with a random center $C_j(c_x, c_y)$.
2. Calculate the distance of each element $e_i \in \mathbb{D}^2$ to the clusters C_j
3. Assign elements to a cluster C_j and recalculate centroid of C_j



Data model – Classification of data

Classification using k-means algorithm

1. Generate potential clusters C_j with a random center $C_j(c_x, c_y)$.
2. Calculate the distance of each element $e_i \in \mathbb{D}^2$ to the clusters C_j
3. Assign elements to a cluster C_j and recalculate centroid of C_j
4. Repeat step 2 and 3 – add more C_j if necessary



Data model – Statistical representation

Characteristic Model

Model which is representative for all data elements but more handy
$\rightarrow$ Convert the clusters dataset to variables by *Principal Component Analysis*

Data model – Statistical representation

Characteristic Model

Model which is representative for all data elements but more handy
$\rightarrow$ Convert the clusters dataset to variables by *Principal Component Analysis*

1. Covariance matrix of the clusters elements

$$Cov_{C_j} = \begin{pmatrix} Cov(X_{C_j}, X_{C_j}) & Cov(X_{C_j}, Y_{C_j}) \\ Cov(Y_{C_j}, X_{C_j}) & Cov(Y_{C_j}, Y_{C_j}) \end{pmatrix}$$

Data model – Statistical representation

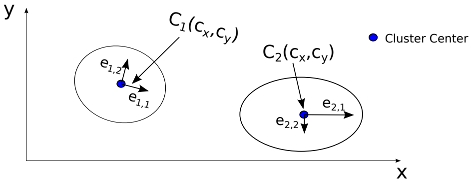
Characteristic Model

Model which is representative for all data elements but more handy
$\rightarrow$ Convert the clusters dataset to variables by *Principal Component Analysis*

1. Covariance matrix of the clusters elements

$$Cov_{C_j} = \begin{pmatrix} Cov(X_{C_j}, X_{C_j}) & Cov(X_{C_j}, Y_{C_j}) \\ Cov(Y_{C_j}, X_{C_j}) & Cov(Y_{C_j}, Y_{C_j}) \end{pmatrix}$$

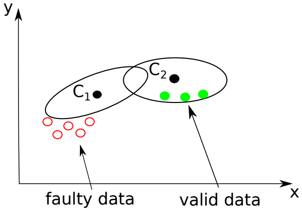
2. Solve $Cov_{C_j} - \lambda E = 0$ to get the eigenvectors $\vec{e_1}$, $\vec{e_2}$
   $\rightarrow$ Normalized eigenvectors represent the cluster C_j statistically



Characteristic model – Continuous update

Online update – Adding new sampled data

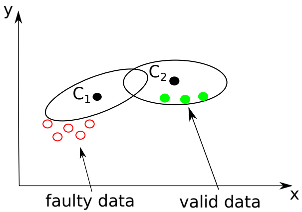
- Incoming data elements are added to pre-clustered dataset

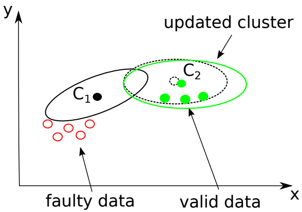


Characteristic model – Continuous update

Online update – Adding new sampled data

- Incoming data elements are added to pre-clustered dataset
- Check if new data is valid
  1. Add new data elements to corresponding clusters
  2. Recalculate the involved covariance ellipses

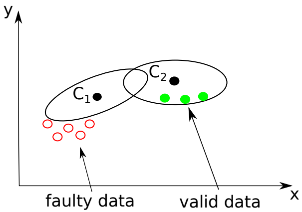


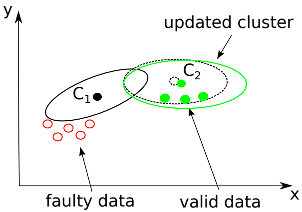


Characteristic model – Continuous update

Online update – Adding new sampled data

- Incoming data elements are added to pre-clustered dataset
- Check if new data is valid
  1. Add new data elements to corresponding clusters
  2. Recalculate the involved covariance ellipses





- Data model gets too tolerant after enduring update
  $\rightarrow$ Recalculate the characteristic model periodically

Characteristic model – Detection of invalid data

Check if an element is part of the characteristic model

- Point-in-cluster
  $\rightarrow$ Check if a new element fits to an ellipse

Characteristic model – Detection of invalid data

Check if an element is part of the characteristic model

- Point-in-cluster
  $\rightarrow$ Check if a new element fits to an ellipse
- Cluster-in-cluster
  $\rightarrow$ Cluster several new data elements and check intersection

Characteristic model – Detection of invalid data

Check if an element is part of the characteristic model

- Point-in-cluster
  $\rightarrow$ Check if a new element fits to an ellipse
- Cluster-in-cluster
  $\rightarrow$ Cluster several new data elements and check intersection
- Manhattan distance (echelon form)

$$d_{Man}(C_j, e_i) = \left\| \begin{pmatrix} c_x - e_x \\ c_y - e_y \end{pmatrix} \right\|_1$$

Characteristic model – Detection of invalid data

Check if an element is part of the characteristic model

- Point-in-cluster
  $\rightarrow$ Check if a new element fits to an ellipse
- Cluster-in-cluster
  $\rightarrow$ Cluster several new data elements and check intersection
- Manhattan distance (echelon form)

$$d_{Man}(C_j, e_i) = \left\| \begin{pmatrix} c_x - e_x \\ c_y - e_y \end{pmatrix} \right\|_1$$

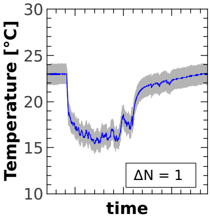
- Mahalanobis distance (considers covariance)

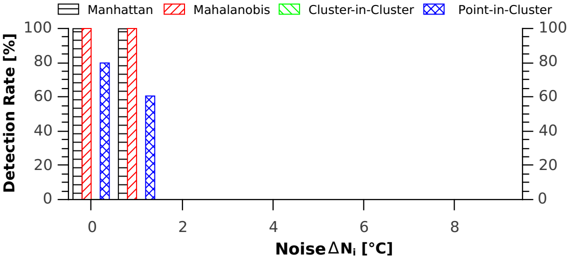
$$d_{Maha}(C_j, e_i) = \sqrt{\begin{pmatrix} c_x - e_x \\ c_y - e_y \end{pmatrix}^T Cov_{C_j}^{-1} \begin{pmatrix} c_x - e_x \\ c_y - e_y \end{pmatrix}}$$

Interim evaluation

Robustness of the characteristic model

- Calculate the characteristic model $\overline{C_m}$ of sample data
  1. Add noise $[0, N_i]$ to the sample data
  2. Feed $\overline{C_m}$ with noised data iteratively
  3. Update model

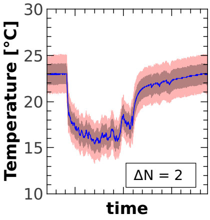


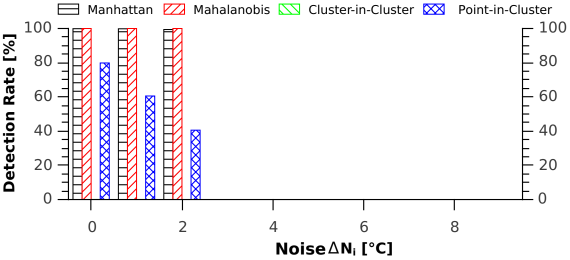


Interim evaluation

Robustness of the characteristic model

- Calculate the characteristic model $\overline{C_m}$ of sample data
  1. Add noise $[0, N_i]$ to the sample data
  2. Feed $\overline{C_m}$ with noised data iteratively
  3. Update model

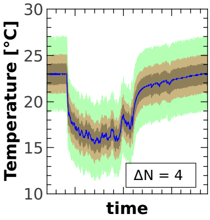


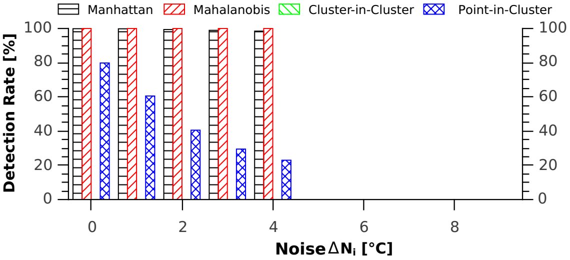


Interim evaluation

Robustness of the characteristic model

- Calculate the characteristic model $\overline{C_m}$ of sample data
  1. Add noise $[0, N_i]$ to the sample data
  2. Feed $\overline{C_m}$ with noised data iteratively
  3. Update model

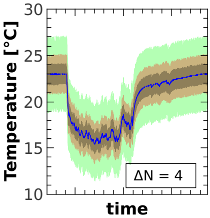


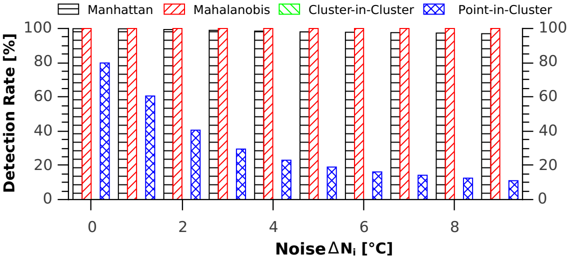


Interim evaluation

Robustness of the characteristic model

- Calculate the characteristic model $\overline{C_m}$ of sample data
  1. Add noise $[0, N_i]$ to the sample data
  2. Feed $\overline{C_m}$ with noised data iteratively
  3. Update model





Functionality and evaluation

Real world experiment

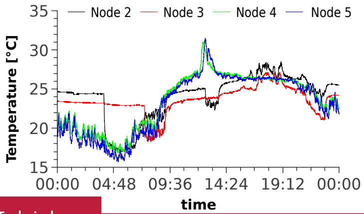
- Temperature sensing with 4 nodes with 1Hz sampling rate (5 days)

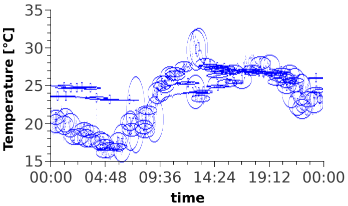
Functionality and evaluation

Real world experiment

- Temperature sensing with 4 nodes with 1Hz sampling rate (5 days)
  $\rightarrow$ Exemplary results of 24h measurement

Node ID	Samples	Runtime	Iterations	Cluster
2	83162	429ms	17	22
3	83635	521ms	19	22
4	83592	581ms	22	28
5	83635	457ms	23	24





Sample application – Redundancy analysis

Detect redundancy between data

Overlapping areas of characteristic models imply redundant sensing

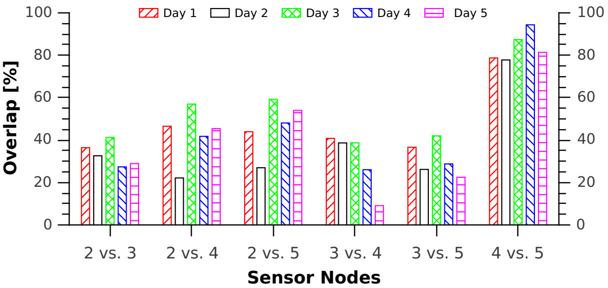
- Experimental setup: *node 4* and *node 5* located side by side
- Calculate intersection $S_{i,j} = \overline{C_i} \cap \overline{C_j}$

Sample application – Redundancy analysis

Detect redundancy between data

Overlapping areas of characteristic models imply redundant sensing

- Experimental setup: *node 4* and *node 5* located side by side
- Calculate intersection $S_{i,j} = \overline{C_i} \cap \overline{C_j}$



Summary and outlook

Goal: Convenient data handling of (potentially faulty) WSN data

- Unreliable sensing: environmental conditions, undervolting, ...

Summary and outlook

Goal: Convenient data handling of (potentially faulty) WSN data

- Unreliable sensing: environmental conditions, undervolting, ...
- Generation of a handy characteristic data model
  1. Classification of data (clustering)
  2. Statistical representation (PCA)

Summary and outlook

Goal: Convenient data handling of (potentially faulty) WSN data

- Unreliable sensing: environmental conditions, undervolting, ...
- Generation of a handy characteristic data model
  1. Classification of data (clustering)
  2. Statistical representation (PCA)
- Initial evaluations
  - Robustness against noisy data
  - Real world experiment (redundancy analysis)

Summary and outlook

Goal: Convenient data handling of (potentially faulty) WSN data

- Unreliable sensing: environmental conditions, undervolting, ...
- Generation of a handy characteristic data model
  1. Classification of data (clustering)
  2. Statistical representation (PCA)
- Initial evaluations
  - Robustness against noisy data
  - Real world experiment (redundancy analysis)
- Next steps

  → Integrate the modelling of data to our PotatoNet testbed

Thank you for your attention! Questions?

Ulf Kulau

kulau@ibr.cs.tu-bs.de