A Classification and Comparison of Data Mining Algorithms for Wireless Sensor Networks, and of Concept Modeling Approaches for Systems of Wireless Sensor Networks (based on Natural Language Processing)

Nemanja Kojić Goran Rakočević
Dragan Milićev Veljko Milutinović
School of Electrical Engineering
University of Belgrade

Staša Vujičić Stanković
School of Mathematics
University of Belgrade

Wireless Sensor Networks

- Wireless Sensor Networks (WSN)
have matured enough,
so that the relevant information
(for a number of applications)
can be generated in a way which is economical,
because sensors have become inexpensive.
- Data Mining (DM) techniques can be effectively applied to WSN systems, to improve the results.

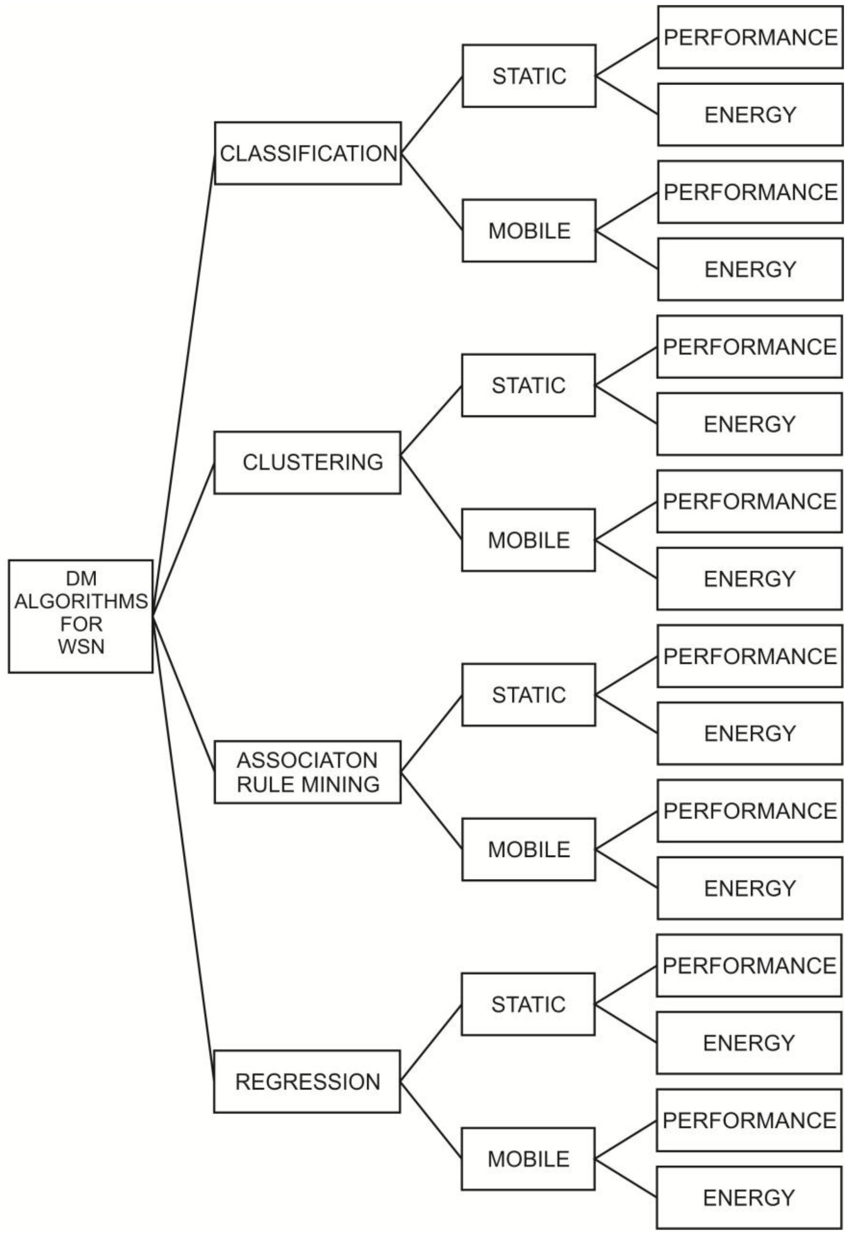
Data Mining

- On the level of a single (e.g., national) Wireless Sensor Network, the major research problems are related to Data Mining, along the following four problem areas:
  - Classification
  - Clustering
  - Regression, and
  - Association rule mining

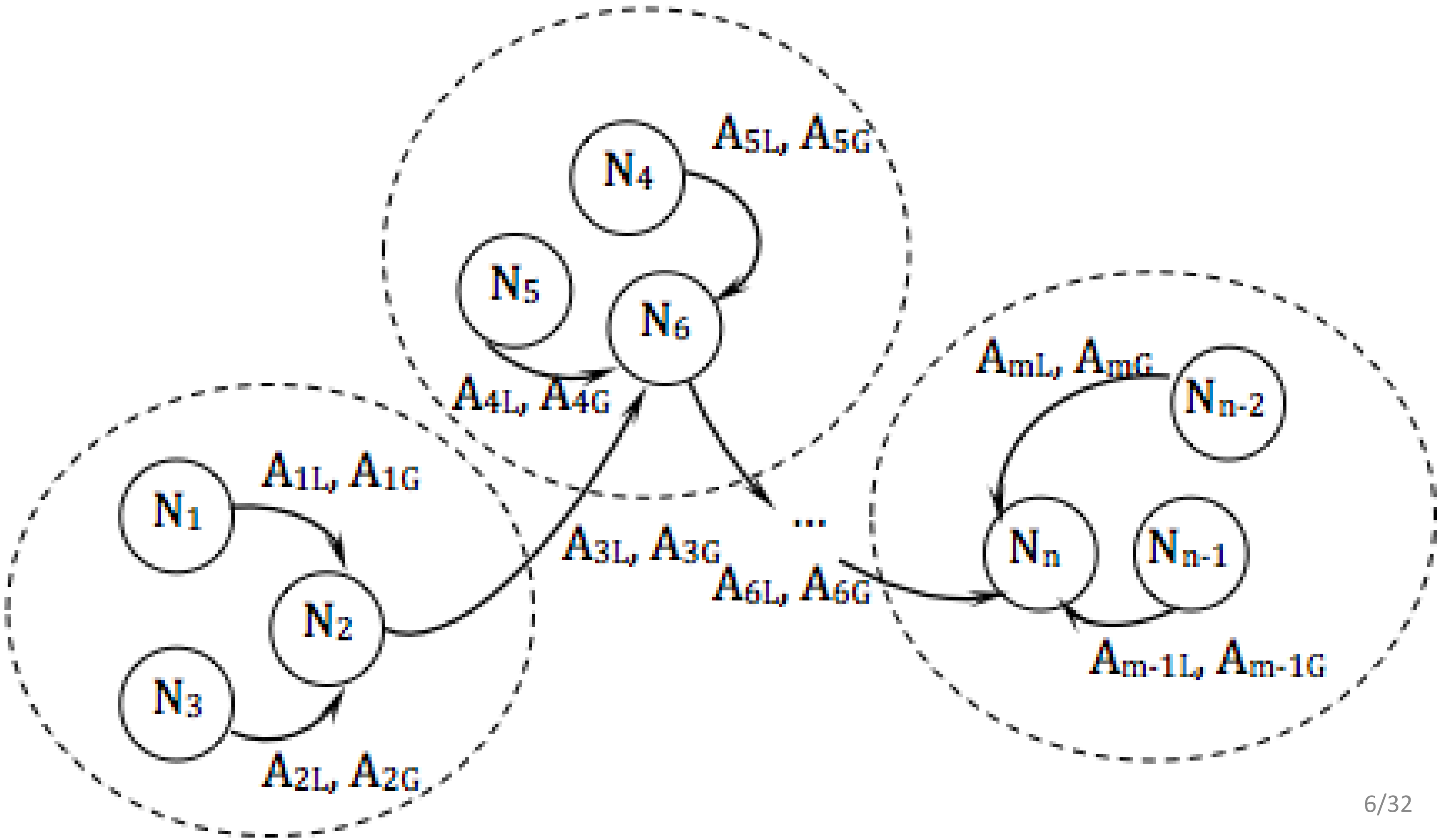
Natural Language Processing
&
Concept Modeling

- Once the set of single (e.g., national) Wireless Sensor Networks is connected into a system of Wireless Sensor Networks, the major research problems are related to Natural Language Processing (NLP) and Concept Modeling (CM):
  - Different Wireless Sensor Networks utilize different terminologies (or even different ontologies) to refer to the same concepts.

Classification Tree (DM@WSNs)



1. Classification Static Performance: ClaSP



Communication pattern in a WSN of the type ClaSP

Legend:

- $\mathbf{N_x}$ – A WSN node parameterized by a couple [$N_{xG,}$ N_{xL}].
- $\mathbf{N_{xG}}$ – Node's global parameter: The cluster a node belongs to.
- $\mathbf{N_{xL}}$ – Node's local parameter:
  Set of features observable through the node's sensors.
- $\mathbf{A_x}$ – An arch denoting a communication line in the network, parameterized by a couple [$A_{xG,}$ A_{xL}].
- $\mathbf{A_{xL}}$ – Arch's local parameter: A couple [N_s, N_d], denoting the source and destination node, respectively.
- $\mathbf{A_{xG}}$ – Arch's global parameter:
  Defining whether the arch denotes communication within the cluster, or between two different clusters.

Training Algorithm in a Wireless Sensor Network of the Type ClaSP

<table>
<tr><th>DISTIRBUTED FIXED-PARTITION
SVM TRAINING DFP-SVM</th><th>WEIGHTED DISTIRBUTED FIXED-PARTITION
SVM TRAINING WDFP-SVM</th></tr>
<tr><td colspan="2">Divide samples into clusters, where each cluster is of the same size.
For every cluster do
Define the cluster head (A sensor which receives data
from all other sensors in the cluster,
performs data fusion, and transmits the results to the base station.)

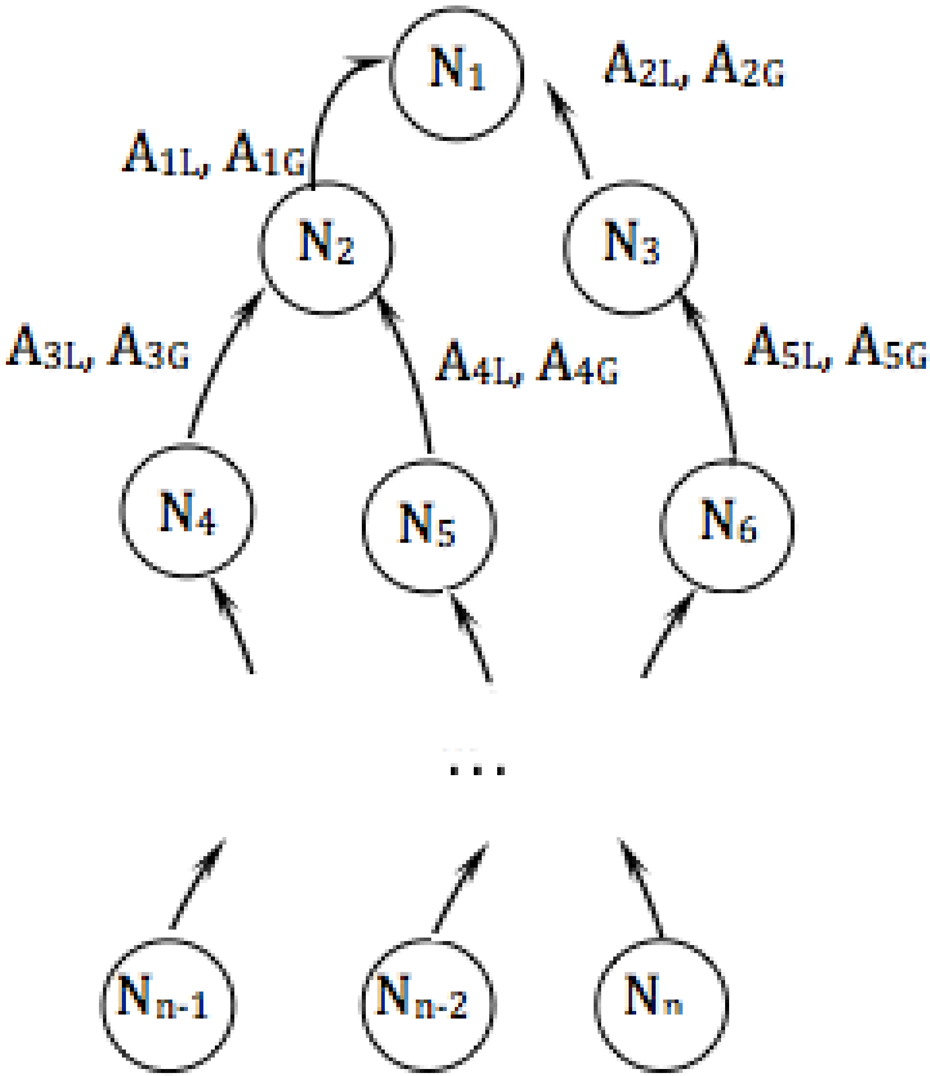
Each sensor transmits its measurement sample vector to the cluster head.
Cluster head combines the measurement sample vector with an estimation
calculated (equations 1, 4) for the previous cluster head,
to make a new SVM.

Cluster head sends the estimated SVM to the cluster head
that is the next one in order.</td></tr>
<tr><td>Cost function for estimation,
equation for DFP-SVM:

$$\min_{\mathbf{w},\xi} \Phi(\mathbf{w},\xi) = \frac{1}{2} \| \mathbf{w} \|^2 + C\sum_{i=1}^{n} \xi_i, \quad (1)$$
where the parameter C defines the cost of constraint violation giving weight to measurements in the set.</td><td>Cost function for estimation,
Equation for WDFP-SVM:

$$\Phi(\mathbf{w},\xi) = \frac{1}{2} \| \mathbf{w} \|^2 + C(\sum_{i\in I} \xi_i + L\sum_{i\in S} \xi_i), \quad (4)$$
where the parameter L increases the cost for the old support vectors giving more weight to former measurements.</td></tr>
</table>

2. Classification Mobile Performance: ClaMP



Communication Pattern in a Wireless Sensor Network of the Type ClaMP

Legend:

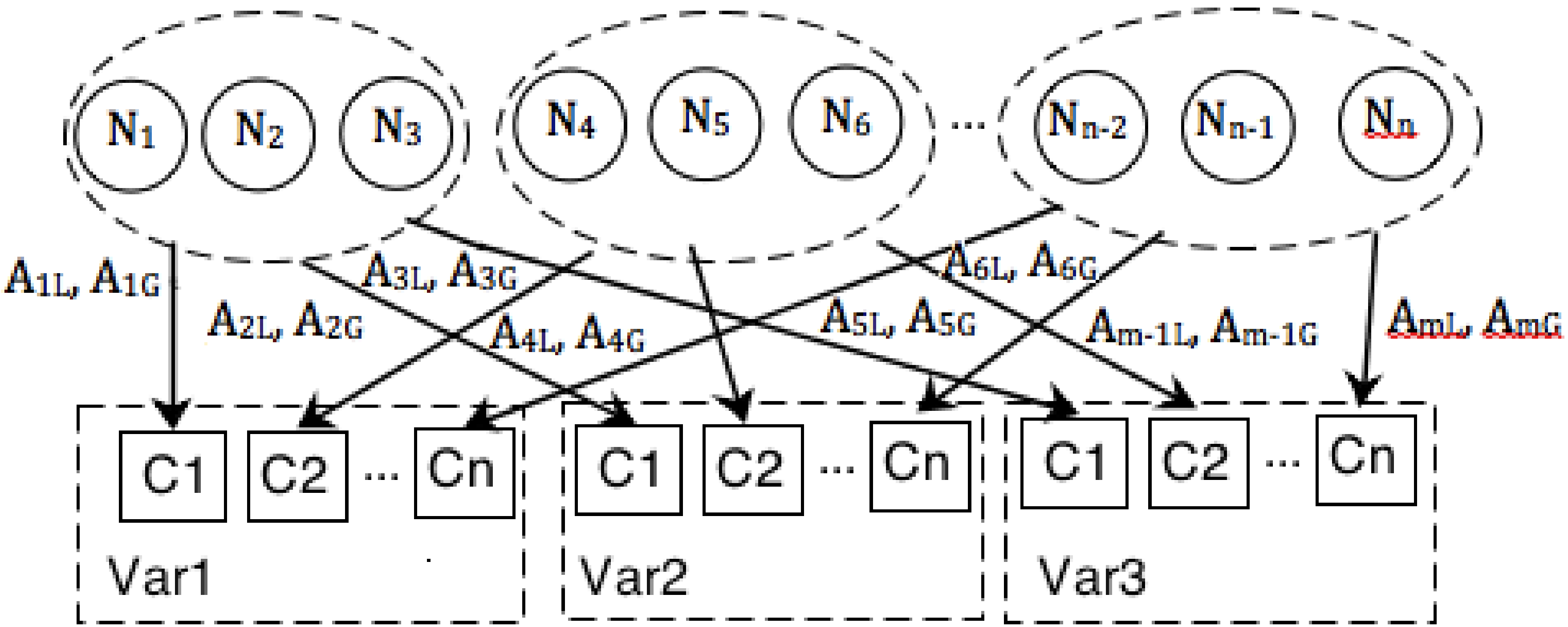
- **N_x** – A WSN node parameterized by a couple [$N_{xG,}$ N_{xL}]
- **N_{xG}** – Node's global parameter:
  The set of weights in weighted voting schemes.
  N/A in the simple voting scheme
- **N_{xG}** – Node's local parameter:
  Set of features observable through the node's sensors
- **A_x** – An arch denoting a communication line in the network parameterized by a couple [$A_{xG,}$ A_{xL}]
- **A_{xL}** – Arch's local parameter: a couple [N_s, N_d], denoting the source and destination node, respectively
- **A_{xG}** – Arch's global parameter: N/A.

Training Algorithm in a Wireless Sensor Network of the Type ClaMP

TRAINING A DISTRIBUTED WSN CLASSIFICATOR

For each node **do**
The node takes the readings from its local sensors.
The node's local predictor is used with the readings to make a local prediction.
The node sends the local prediction to the central server.
If Voting scheme other than Simple voting is used
Find the appropriate weight associated with the node and its local prediction.
End If
End For
Sum the votes across all of the nodes to reach a global prediction.

3. Clustering Mobile Energy: CluME



Communication pattern in a Wireless Sensor Network of the type CluME

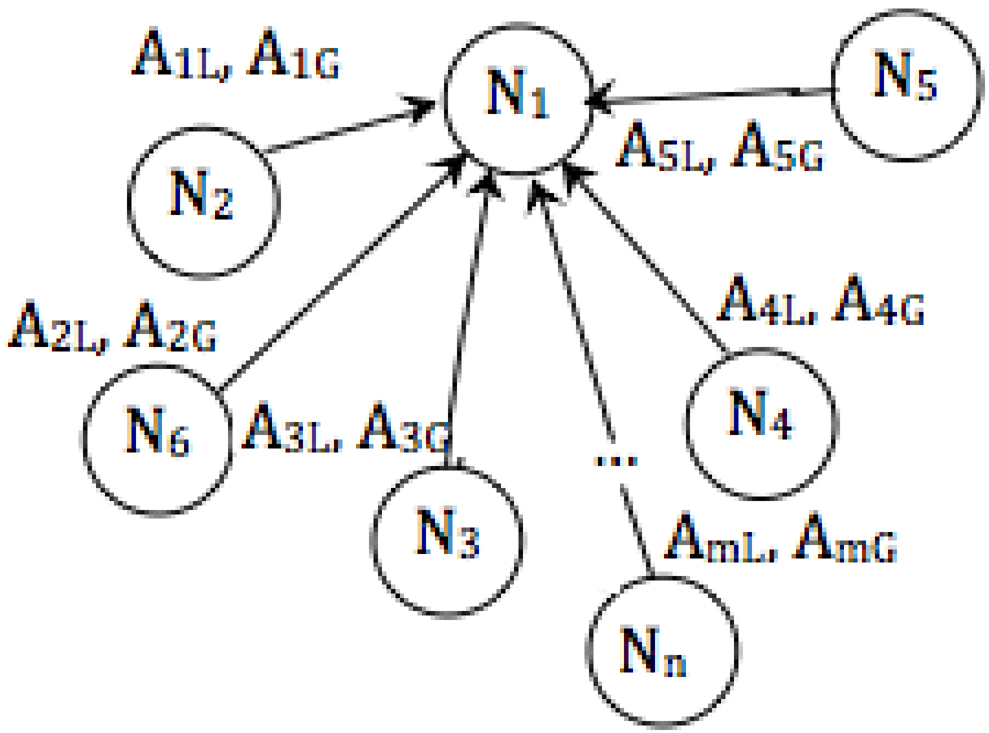
Legend:

- **N_x** – A WSN node parameterized by a couple [N_{xG}, N_{xL}]
- **N_{xG}** – Node's global parameter: The cluster a node belongs to
- **N_{xL}** – Node's local parameter:
  Set of features observable through the node's sensors
- **A_x** – An arch denoting that the readings from the sensors in the originating node cluster belong to the appropriate cluster of values [A_{xG}, A_{xL}]
- **A_{xL}** – Arch's local parameter: A couple [N_s, N_d],
  denoting the source and destination node, respectively
- **A_{xG}** – Arch's global parameter:
  Defining whether the arch denotes communication within the cluster, or between two different clusters

Training Algorithm in a Wireless Sensor Network of the Type CluME

Calculating Principals Componets from streaming data (SPIRIT approach)
Cluster the sensor data along each sensor attribute separatly
Construct a bipartite graph, with the set of sensor nodes and the set of the data cluster as vertex sets, so that a vertiex form a node to a cluster denotes that the vaules from the node belong to the cluster
Find all complete bipartite subgraphs

4. Regression Mobile Performance: RMP



Communication pattern in a WSN of the Type RMP

Legend:

- **N_x** – A WSN node parameterized by a couple [$N_{xG,}$ N_{xL}]
- **N_{xG}** – Weights corresponding to the node's readings
- **N_{xL}** – Node's local parameter: Set of features observable through the node's sensors
- **A_x – A_Y** – An arch denoting a communication line in the network

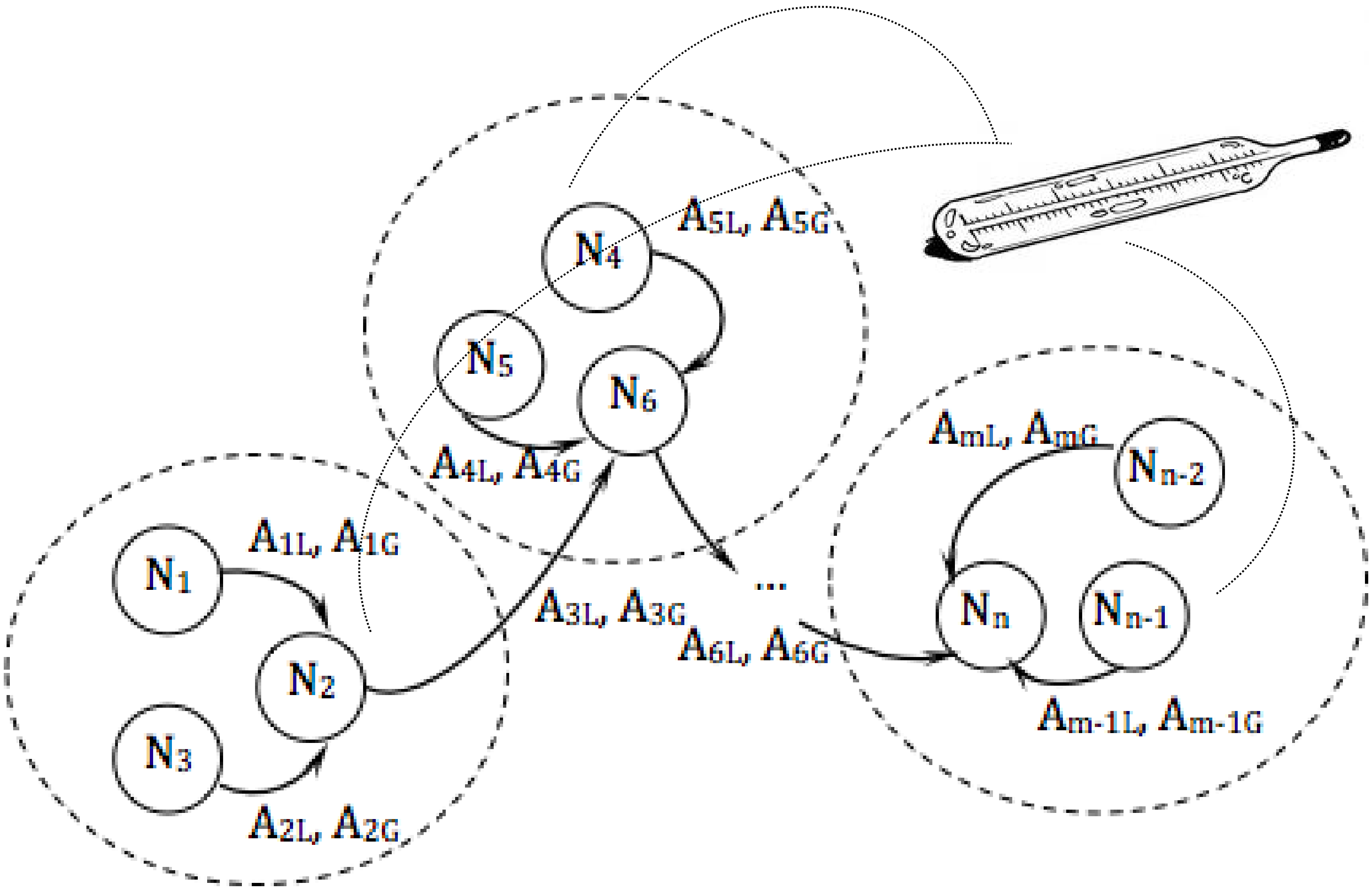
Training Algorithm in a Wireless Sensor Network of the type RMP

Calculating Principals Componets from streaming data (SPIRIT approach)

Initialise the k hidden variables W
to unit vectors w_1 = [10⋯0]T, w_2 = [010⋯0]T, etc.
Initialise di (i = 1, ...k) to a small positive value.
While x_{t+1} arrives
    Update $x'_1 = x_{t+1}$.
    For $1 \leq i \leq k$.
        Calculate y_i, d_i, $\mathbf{e}_i$, $\mathbf{w}_i$, $x_{t+1.}$
    End_For
    Update $x'_{i+1} = x_i - y_i \mathbf{w}_{i.}$
End_While

$y_i = \mathbf{w}_i^T x'_i$ ($y_{t+1,i}$ = projection onto w_i) $d_i = \lambda d_i + y_i^2$ (energy ∝ i-th eigenval. of $X^T_t X_t$) $\mathbf{e}_i = x'_i - \mathbf{w}_i y_i$ (error, ei ⊥ wi) $\mathbf{w}_{i} = \mathbf{w}_i + \mathbf{w}_i \mathbf{e}_i / d_i$ (update PC estimate)

5. Clustering Static Energy: CluSE



Communication pattern in a WSN of the type ClaSP

Legend:

- $\mathbf{N_x}$ – A WSN node parameterized by a couple [$N_{xG,}$ N_{xL}]
- $\mathbf{N_{xG}}$ – Node's global parameter: The cluster a node belongs to
- $\mathbf{N_{xL}}$ – Node's local parameter:
  Set of features observable through the node's sensors
- $\mathbf{A_x}$ – An arch denoting a communication line in the network,
  parameterized by a couple [$A_{xG,}$ A_{xL}]
- $\mathbf{A_{xL}}$ – Arch's local parameter: A couple [N_s, N_d],
  denoting the source and destination node, respectively
- $\mathbf{A_{xG}}$ – Arch's global parameter:
  Defining whether the arch denotes communication
  within the cluster, or between two different clusters.

Training Algorithm in a Wireless Sensor Network of the type CluSE

BASIC ALGORITHM

For each node **do**
    sensor becomes a *volunteer clusterhead – VC* with probability *p*.
    If node became VC
        node advertises itself as a *clusterhead – CH*
        to the sensors within its radio range.
        For (all the sensors that are no more than *k* hops away from the CH) **do**
            forward the advertisement
        End_For
    End_If
    If a node that receives a *CH* advertisements is not itself a *CH*
        the node joins the cluster of the closest *CH*.
    Else_If the node is not a clusterhead and sensor has not joined any cluster
        the node becomes a *forced clusterhead – FCH*.
    End_If
    If node does not receive a *CH* advertisement within time duration *t*
        the node become a *FCH.*
    End_If
End_For

* where *t* units is the time required for data to reach the clusterhead from any sensor *k* hops away.

HIERARCHICAL ALGORITHM

For each node **do**
    the node becomes a *level-1 clusterhead* – *level-1 CH* with probability p_1.
    If node became VC
        node advertises itself as a *clusterhead* – *CH*
        to the sensors within its radio range.
        For (all the sensors that are no more than *k* hops away from the CH) **do**
            forward the advertisement
        End_For
    End_If
End_For
For each node that receives an advertisement **do**
    sensors joins the cluster of the closest *level-1CH*
End_For
For each node that does not receive an advertisement **do**
    the sensors become *forced level-1 CHs*.
End_For
For each node **do**
    communicate the gathered data to *level-1 CHs*.
End_For

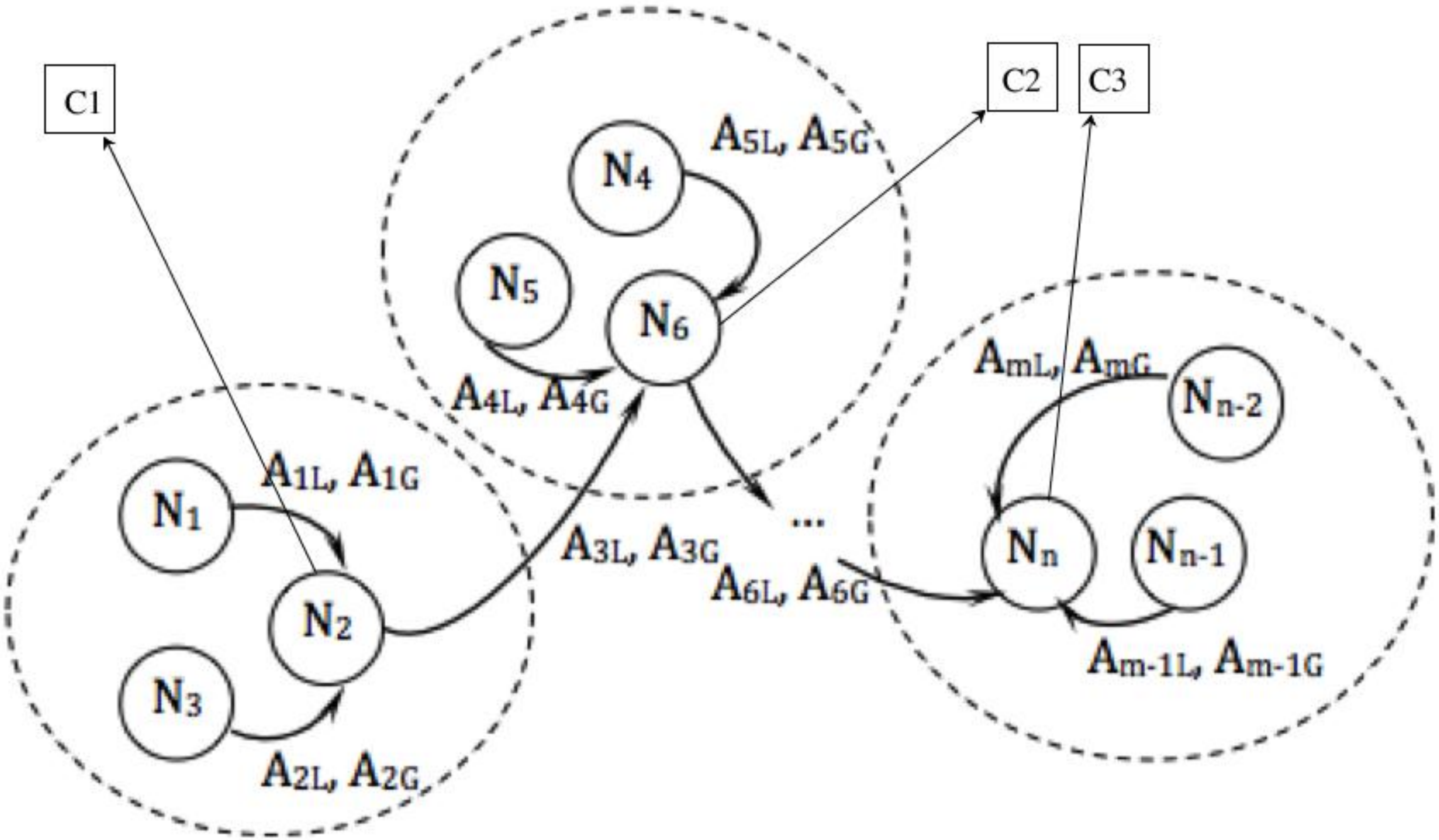
```
i:=1;
while (i<h) do
      for each (level-i CH) do
             level-i CH elect themselves as level-(i+1) CHs with a probability pi+1
            and broadcast their decision of becoming a level-(i+1) CH
            to all the sensors within ki+1 hops.
       End_For
       For (all the level-i CHs that receive the advertisements from level-(i+1) CHs) do
                          level-i CHs joins the cluster of the closest level-(i+1) CH.
       End_For
       For (all the level-i CHs do not receive an advertisement from level-(i+1) CHs) do
                the level-i CHs become forced level-(i+1) CHs.
       End_For

       For (all the level-i CHs) do
                aggregate the data and communicate the aggregated data
                or estimates based on the aggregated data to level-(i+1) CHs
       End_For
              i:=i+1;
End_While
```

The *level-h CHs* communicate the aggregated data or estimates based on this aggregated data to the processing center.

* where *h* is the number of levels in the clustering hierarchy with level 1 being the lowest level and level *h* being the highest.

6. Clustering Static Energy: CluSE



Communication pattern in a WSN of the type CluSE

Legend:

- **Nx** – A WSN node parameterized by a couple [NxG, NxL]
- **NxG** – Node's global parameter: The cluster a node belongs to
- **NxL** – Node's local parameter:
  Set of features observable through the node's sensors
- **Ax** – An arch denoting a communication line in the network, parameterized by a couple [AxG, AxL]
- **AxL** – Arch's local parameter: A couple [Ns, Nd],
  denoting the source and destination node, respectively
- **AxG** – Arch's global parameter:
  Defining whether the arch denotes communication within the cluster, or between two different clusters.
- $\mathbf{C_x}$ – gateway to the internet/outside world

Training Algorithm in a Wireless Sensor Network of the Type CluSE

Generating Clusters in a static, energy aware, clustering apporach

The clusterhead (CH) generates a prediction-model inside a prediction model unit.

While next time unit **do**

The CH sends a prediction-model to all the sensors in the cluster.

while (the prediction-model is valid) **do**

For each node **do**

recieve a prediction-model from the CH

compare the sensor's reading

with the reading predicted by the prediction-model.

If they differ for more than some preconfigured margin of error

send sensor's readings to the CH.

End_If

End_for

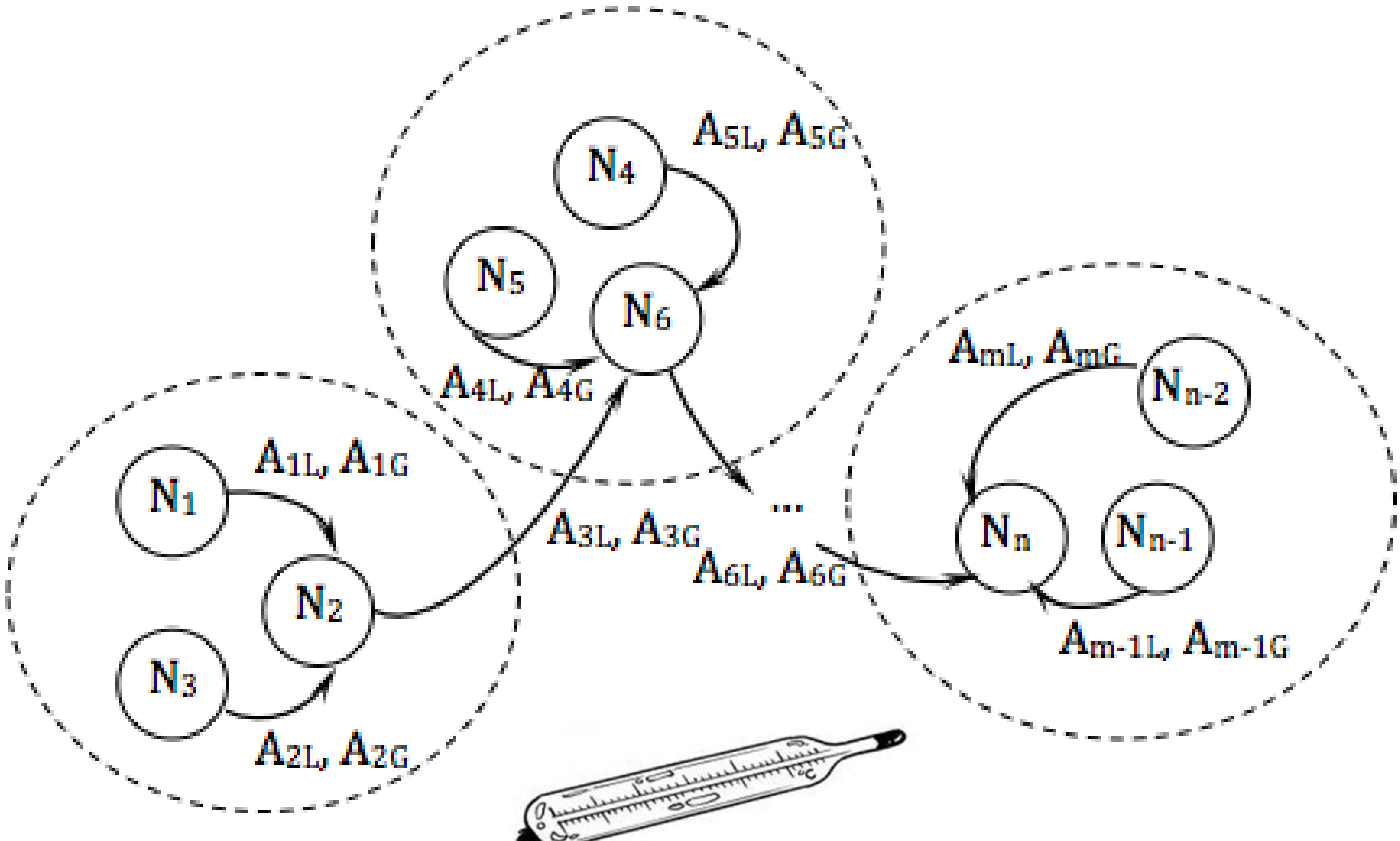
The CH collects updates from the sensors and the prediction model generation unit computes new prediction model.

End_while

If the prediction-model resulted in fewer update
CH sends encoded set of prediction models, followed by the updates necessary to override wrong predictions to an access point.
End_If

The access points collectively maintain a database of current readings of all the sensors in the sensor fields, so the user interest in monitoring a query may register its interest with the appropriate access point.

7. Association rule mining Static Energy: ArmSE



Communication pattern in a WSN of the type ClaSP.

Legend:

- $\mathbf{N_x}$ – A WSN node parameterized by a couple [$N_{xG,}$ N_{xL}]
- $\mathbf{N_{xG}}$ – Node's global parameter: The cluster a node belongs to
- $\mathbf{N_{xL}}$ – Node's local parameter:
  Set of features observable through the node's sensors
- $\mathbf{A_x}$ – An arch denoting a communication line in the network, parameterized by a couple [$A_{xG,}$ A_{xL}]
- $\mathbf{A_{xL}}$ – Arch's local parameter: A couple [N_s, N_d],
  denoting the source and destination node, respectively
- $\mathbf{A_{xG}}$ – Arch's global parameter:
  Defining whether the arch denotes communication within the cluster, or between two different clusters.

Training Algorithm in a Wireless Sensor Network of the Type ArmSE

```
Static, energy aware, Associaation Rule Mining Alghoritam

for (all rounds of sensor readings) do
    begin
        checkBuffer();
        update();
        estimateValue();
    end.
checkBuffer()
while (the current session lasts) do
        record the data received from a particular sensor to corresponding field in the Buffer
for (all fields in the Buffer) do
        check if there is a missing value
if missing value exists
        estimateValue() for that missing value
Else
        send OK signal to queries
        update()
End_If
```

* The Buffer is the data structure to store the arriving readings associated with the corresponding sensors.

<u>update()</u>

// The purpose of this algorithm is to update the Cube and the Counter every time a new round (without missing values) of sensor readings is stored in the Buffer.

For all sensor readings in the Buffer **do**
  update 1-itemsets
  add new nodes at the front of the Cube
  discard the oldest nodes at the back of the Cube
  update the Counter
End_for
For all sensor readings in the Buffer **do**
    generate 2-itemsets between the sensor readings in the particular round
  add new nodes at the front of the Cube
  discard the oldest nodes at the back of the Cube
  update the Counter
End_For

* The Cube keep track of all existing 1- and 2-itemsets in each round, which are stored in the corresponding nodes and slices.
* The Counter data structure speeds up the estimation of a missing value.

<u>estimateValue()</u>

for all missing values **do**
  estimate the missing value
  store it in the Buffer
End_For
update()

Thank you for your attention!

A Classification and Comparison of Data Mining Algorithms for Wireless Sensor Networks, and of Concept Modeling Approaches for Systems of Wireless Sensor Networks (based on Natural Language Processing)

Nemanja Kojić Goran Rakočević
Dragan Milićev Veljko Milutinović
School of Electrical Engineering
University of Belgrade

Staša Vujičić Stanković
Duško Vitas
University of Belgrade