**Overview & Motivations**

**Traditional Approaches:**
- Publicly accessible datasets (benchmarks)
- Synthetic datasets
- Simple data collection (natural faults)

**Over-the-Air Programming (OTAP) / Network Reprogramming:**
- High reprogramming latency/overhead
- A need for selective reprogramming (faulty/compromised nodes)

**SNMiner Approach:**
- Facilitates fault modeling (direct interaction with the deployed sensor network)
- Simulates malicious activities within a real-world deployment
- Avoids network reprogramming (no interruption of operation)
- Constantly re-evaluates classification accuracy

---

**Operation (Functional Diagram)**

**Fault Modeling**

**Fault/Compromise Models**
- Naive: sensors emit falsified readings in a continuous manner
- Smart: sensors emit falsified readings intermittently

---

**Feature Extraction**

- Every $T_{p}$, extract a statistical feature tuple: $\{f_{1}, f_{2}, f_{3}, f_{4}, f_{5}, f_{6}\}$ [4]
  - $f_{1} = \frac{(\text{mean})_{\text{deviation}} - \sum (\text{mean})_{\text{deviation}}}{\text{mean}}$ represents the variance of a sensor aspect $\text{aspect}_{\text{deviation}}$ where $\text{mean}$ is the number of samples for each node over one $T_{p}$, $M$ is the number of nodes, and $K$ is the number of attached sensors
  - $f_{2} = \frac{\text{absolute deviation}}{\text{deviation}}$ represents the absolute deviation of a sensor aspect $\text{aspect}_{\text{deviation}}$ where $\text{deviation}$ is the absolute deviation of node $m$ among all nodes’ $m$th readings
  - Example feature tuple for three attached sensors: temperature, light, and acoustic

---

**Data Collection and Visualizations**

**Central Sensor DB**

**Sensor Network Deployment**

**Synthetic/Benchmark Datasets**

---

**Evaluation of Model Accuracy**

**Re-Evaluating The Ensemble**
- Uses JBoost implementation
- Ensemble classifiers: Adaboost, LogitBoost (usually have identical performance for anomaly detection in sensor networks)
- Base classifiers: decision stumps, decision trees, adaptive decision trees, combination of decision trees.

**Observations:**
- Exploiting the existence of other sensors on board makes it easier to identify faulty nodes.
- Simple base classifiers (i.e. decision stumps) have a comparable performance to other tree-based ensembles.
- Trade-off between number of iterations and complexity of the base classifier. A smaller number of iterations reduces over-fitting.

**Evaluation of Model Accuracy**

**Decision Stumps**

- Decision Stumps

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Deviation</th>
<th>Fault Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Large</td>
<td>Continuous</td>
</tr>
<tr>
<td>5</td>
<td>I.A</td>
<td>Continuous</td>
</tr>
<tr>
<td>12</td>
<td>I.A</td>
<td>Very Small</td>
</tr>
<tr>
<td>15</td>
<td>I.A</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

**Decision Trees**

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Deviation</th>
<th>Fault Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Large</td>
<td>Continuous</td>
</tr>
<tr>
<td>5</td>
<td>I.A</td>
<td>Continuous</td>
</tr>
<tr>
<td>12</td>
<td>I.A</td>
<td>Very Small</td>
</tr>
<tr>
<td>15</td>
<td>I.A</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

---

**Future Work**

- Fault Modeling
- Incorporate additional fault models.
- Support additional sensors.

**Model Evaluation**
- Automatically obtain the best classifier for the current sensor network deployment
- Other classification techniques: SVM, Bagging, etc.
- Avoid over-fitting

---

**References**

- Tao Zhang, Giovani Rimon Abuaitah, Bin Wang, and Zhiqiang Wu. 
- TBD and D. Culler. 
  Design of an application-supportive management system for wireless sensor networks. 
  In *DCSF, pages 1 - 14, 2006.*
- TBD and D. Culler. 
  Design of an application-supportive management system for wireless sensor networks. 
  In *DCSF, pages 1 - 14, 2006.*
- TBD and D. Culler. 
  Design of an application-supportive management system for wireless sensor networks. 
  In *DCSF, pages 1 - 14, 2006.*