A System for Distributed Event Detection in Wireless Sensor Networks

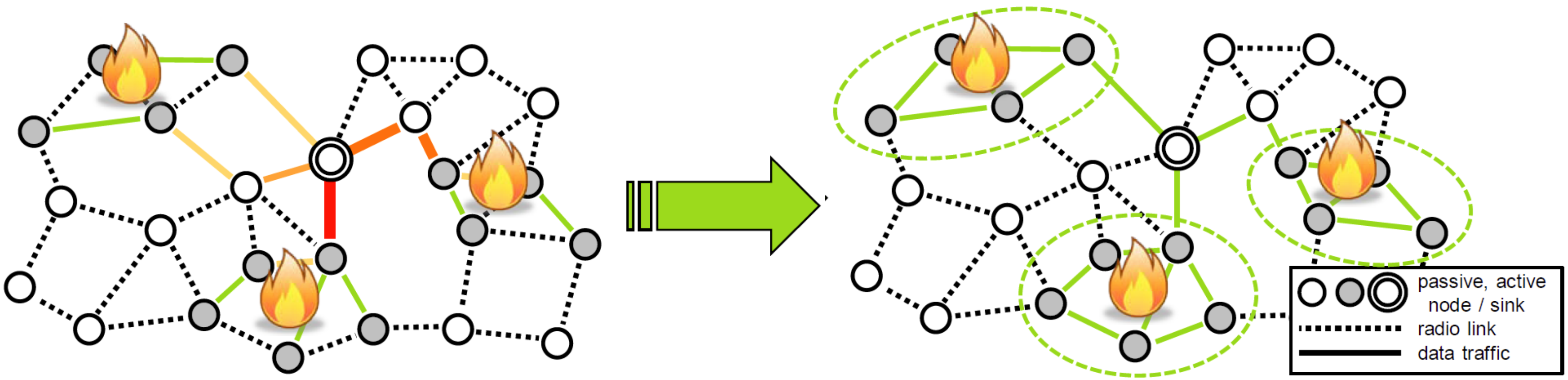
CSCI 780 Wireless Sensor Networks

Gang Zhou

Computer Science
College of William and Mary

Based on slides from Georg Wittenburg etc.

Motivation



- General-purpose event detection
  - Decide locally whether an application-specific event occurred (e.g., "There is a fire!")
  - Only transmit confirmed events to the base station
  - Reduce communication between nodes and base station and extend network lifetime

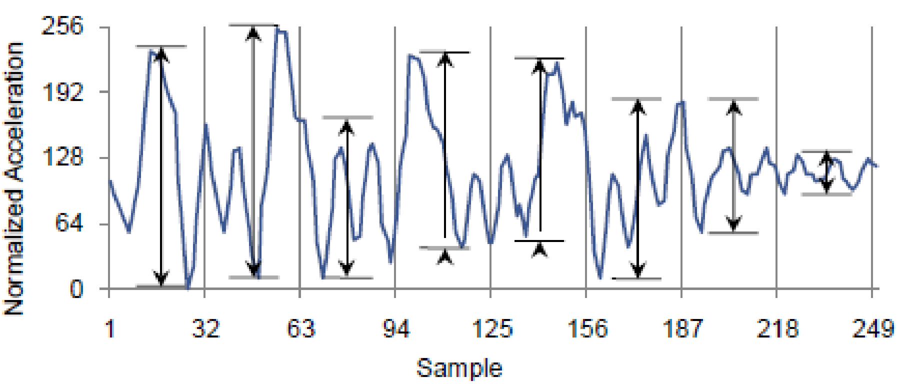
Use Case: Fence Monitoring

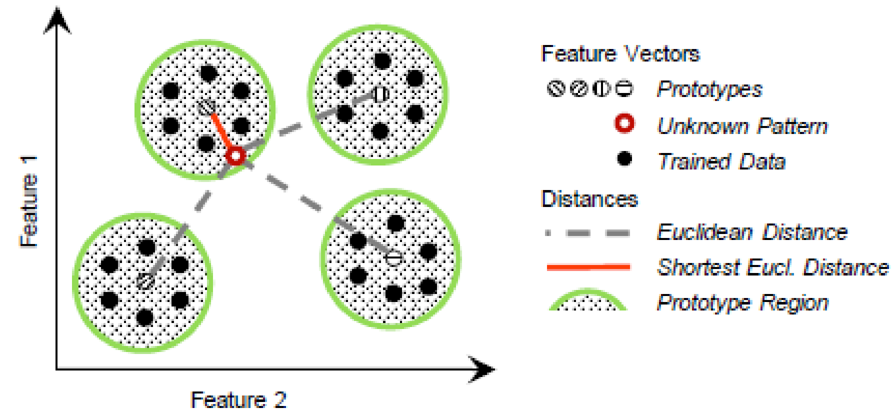

- Sensor nodes attached to fence measure acceleration to detect security-relevant events (e.g., intruder over fence)
- Use cases: access control, perimeter security, more??
- Suitable properties:
  - No mobility
  - Can be large deployment, i.e., long route to base station

Outline

- Motivation
- Basic Approach: Pattern Matching
- System Design
- Performance evaluation
- Conclusion
- Discussion

Background: Pattern Mattching

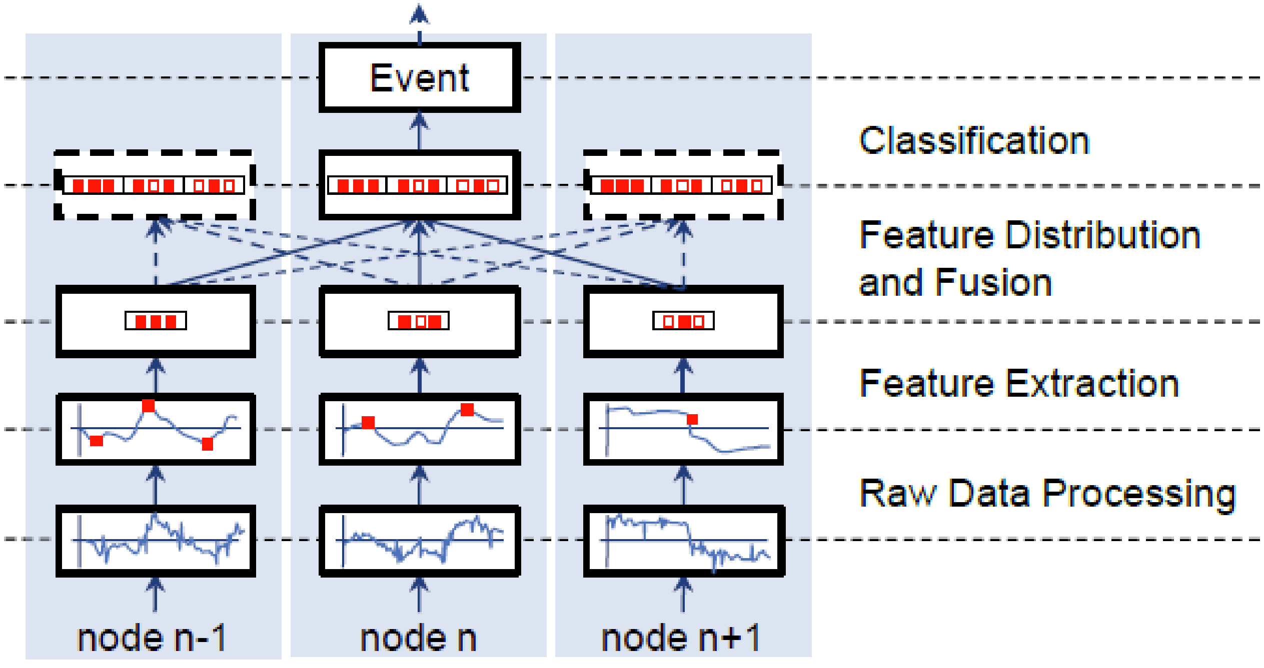




- Feature extraction
  - Extract set of descriptive features from sampled raw data
  - Example: for each time interval, the difference between the minimal and maximal acceleration value is computed
- Classification
  - Use extracted features to deduce previously trained event
  - Example: four prototype vectors established by averaging training data;

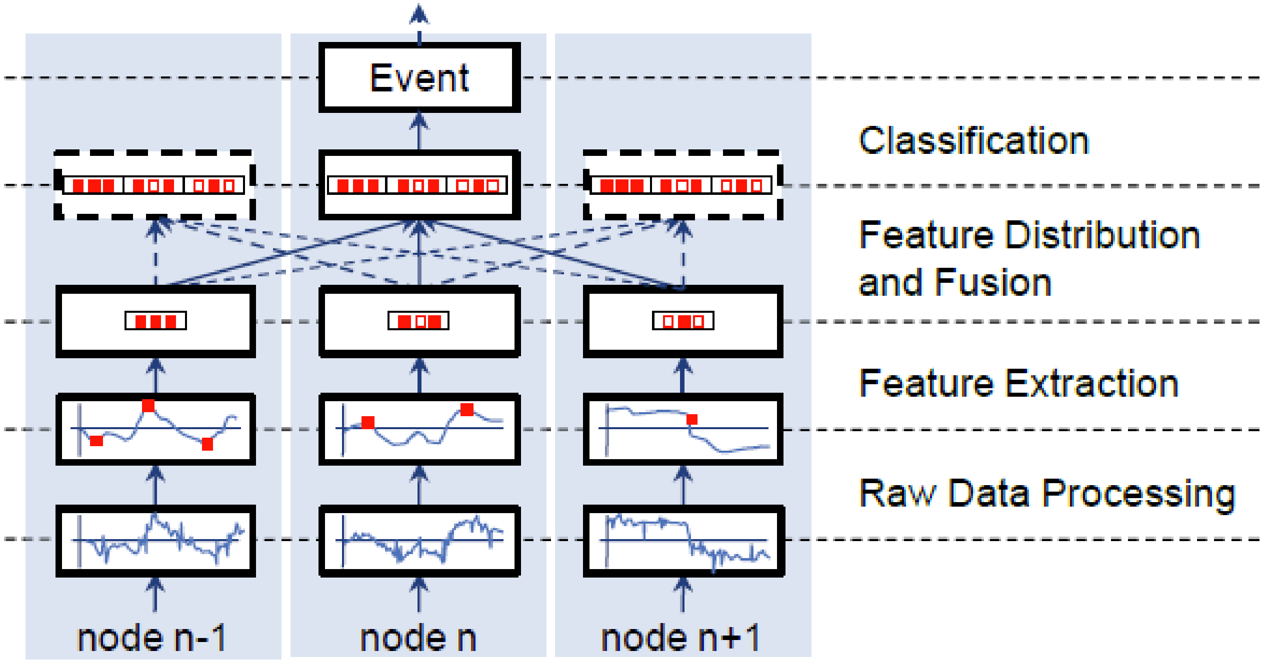
    Classify feature vector by finding nearest prototype vector

Applying Pattern Matching in WSN



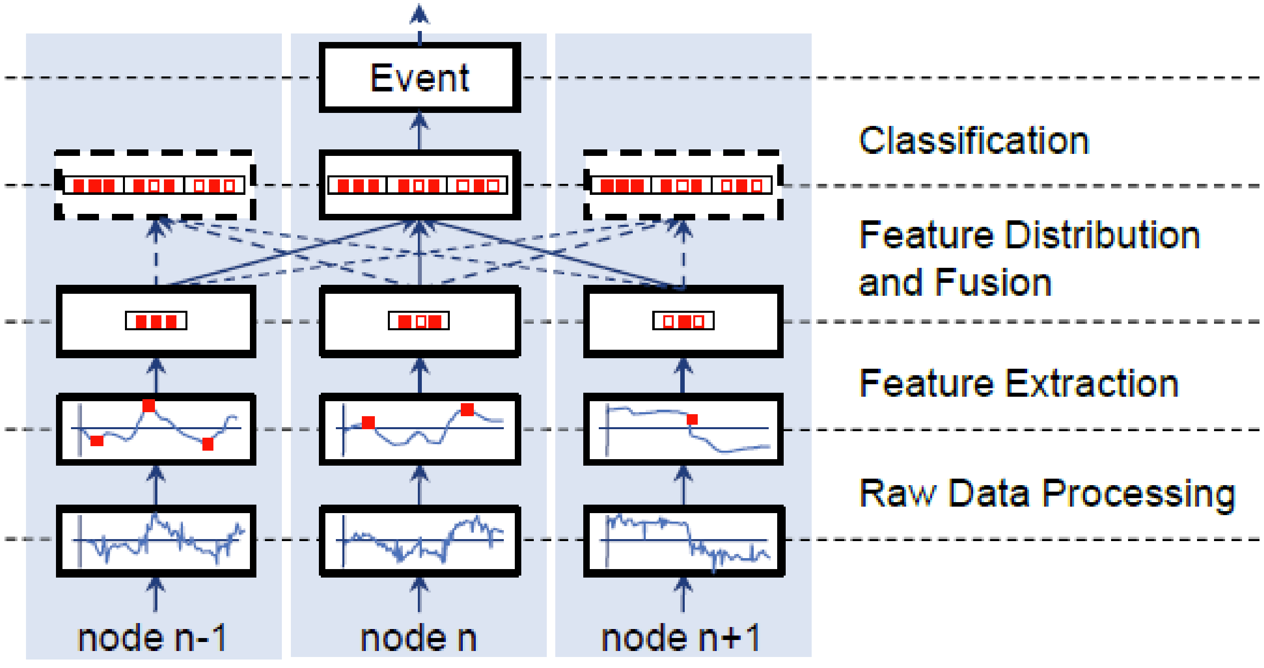
- Raw Data Processing:
  - Filter, segment, and normalize the data
  - Control sampling frequency
    - Preserve energy in phases of inactivity

Applying Pattern Matching in WSN



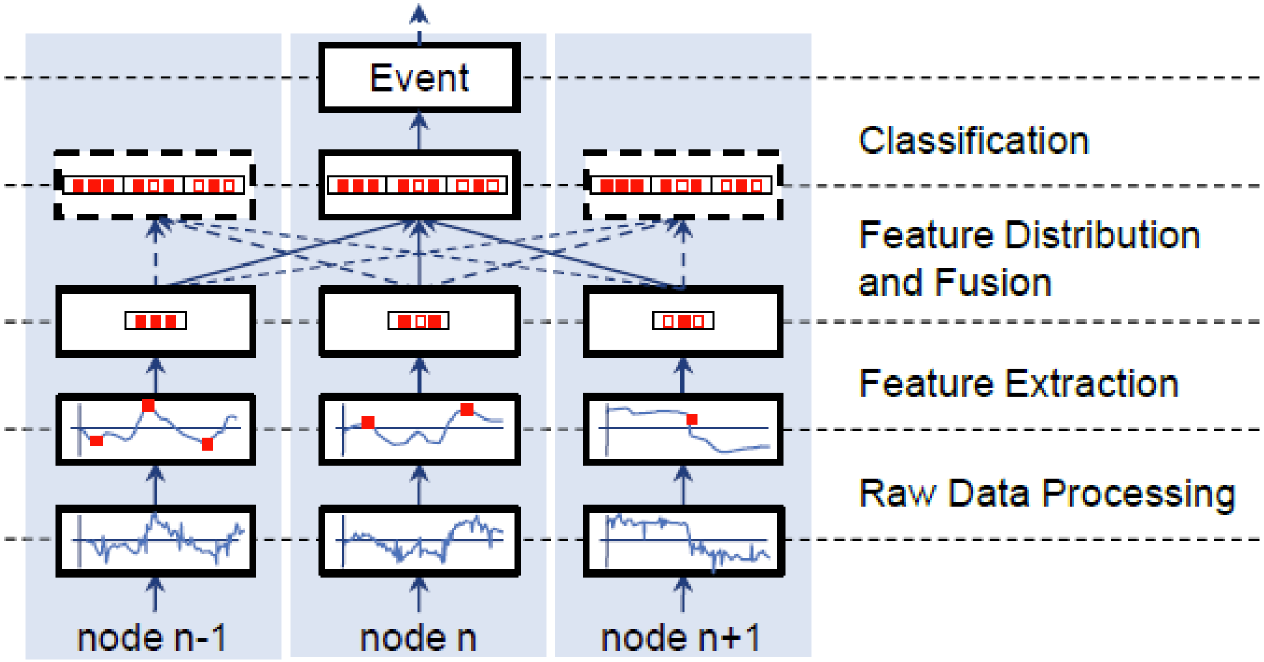
- Feature Extraction:
  - Extract application-specific set of features from raw data
  - Selection of appropriate features is part of training
    - LOOCV algorithm is used, will explain later

Applying Pattern Matching in WSN



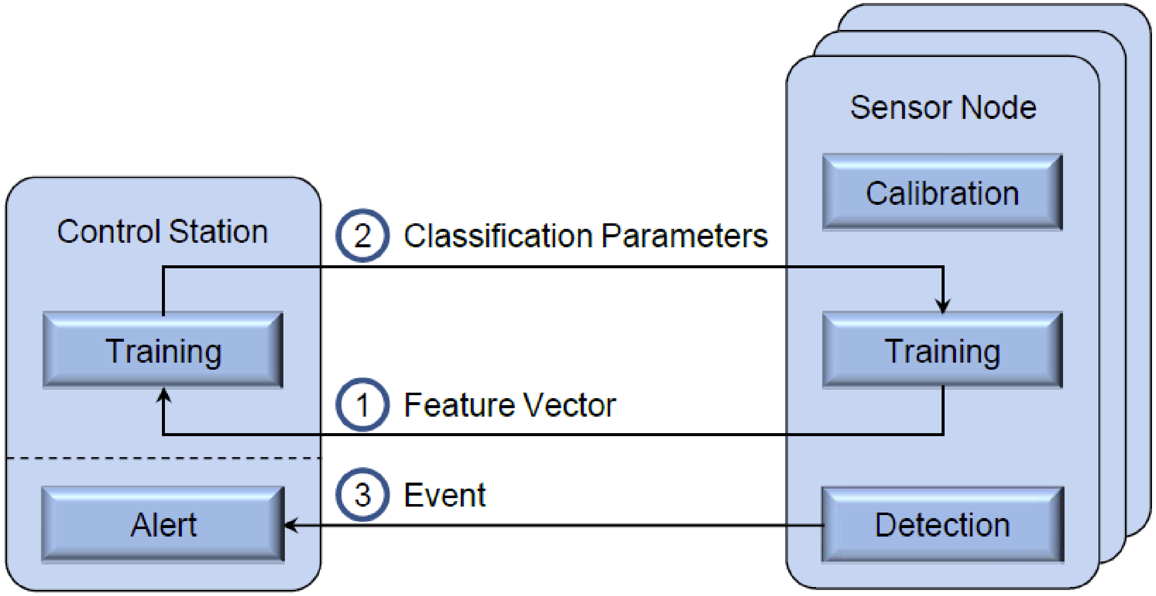
- Feature Distribution / Fusion:
  - Broadcast features to n-hop neighborhood (usually n= 1)
  - Retransmit features in case of transmission failures

Applying Pattern Mattching in WSN



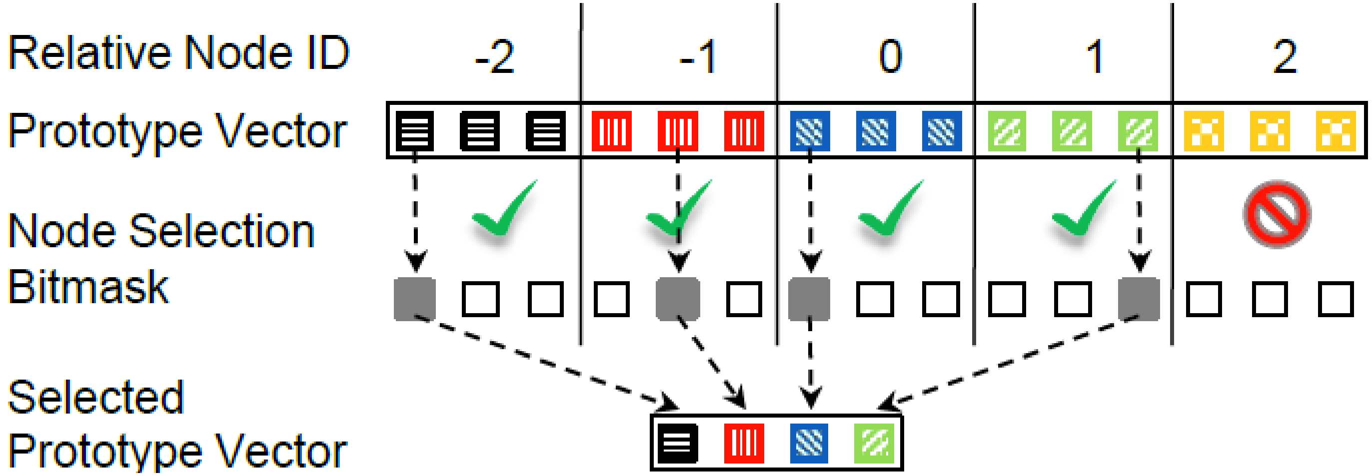
- Classification / Reporting:
  - Combine local and received features into feature vector
  - Classify feature vector
- Report event to base station or locally log event for user-initiated retrieval
- Base station can further fuses classification reports

System Design



- Training
  - Expose sensor network to series of training events
  - Extract all supported features and transmit them to control station
- Setup
  - Select best subset of features, calculate prototype vector for each event
  - Configure nodes to only extract/transmit selected features, setup prototype vectors
- Detect and report events

Feature Selection

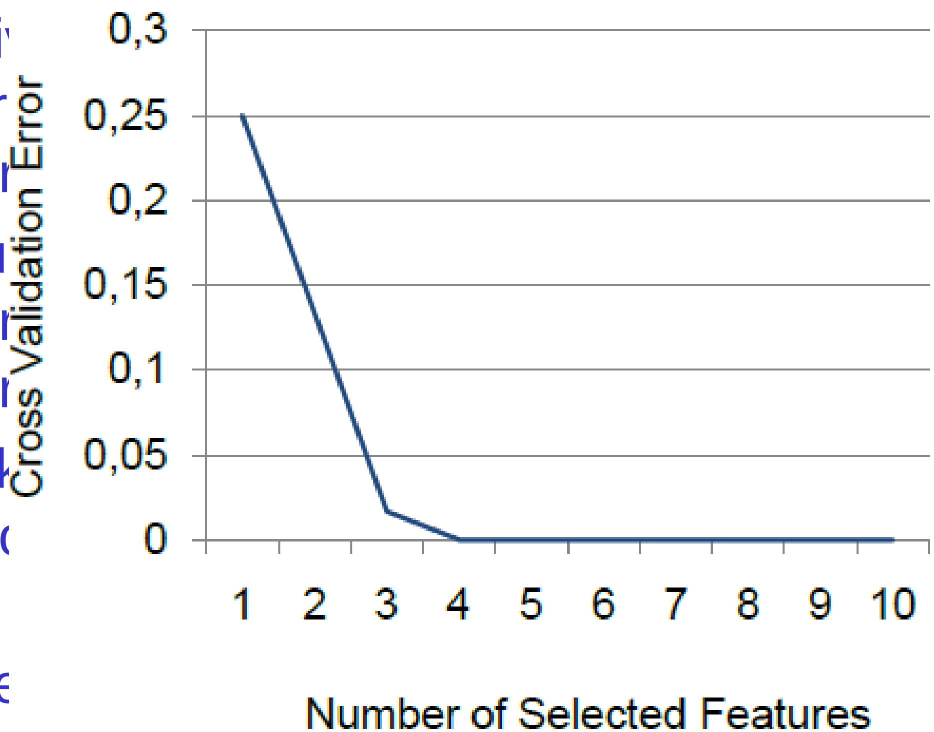


- Two selection steps:
  - ➢ If #feature vectors received from a node is too small, discard all features from that node
- Select only high quality features
  - ➢ With LOOCV

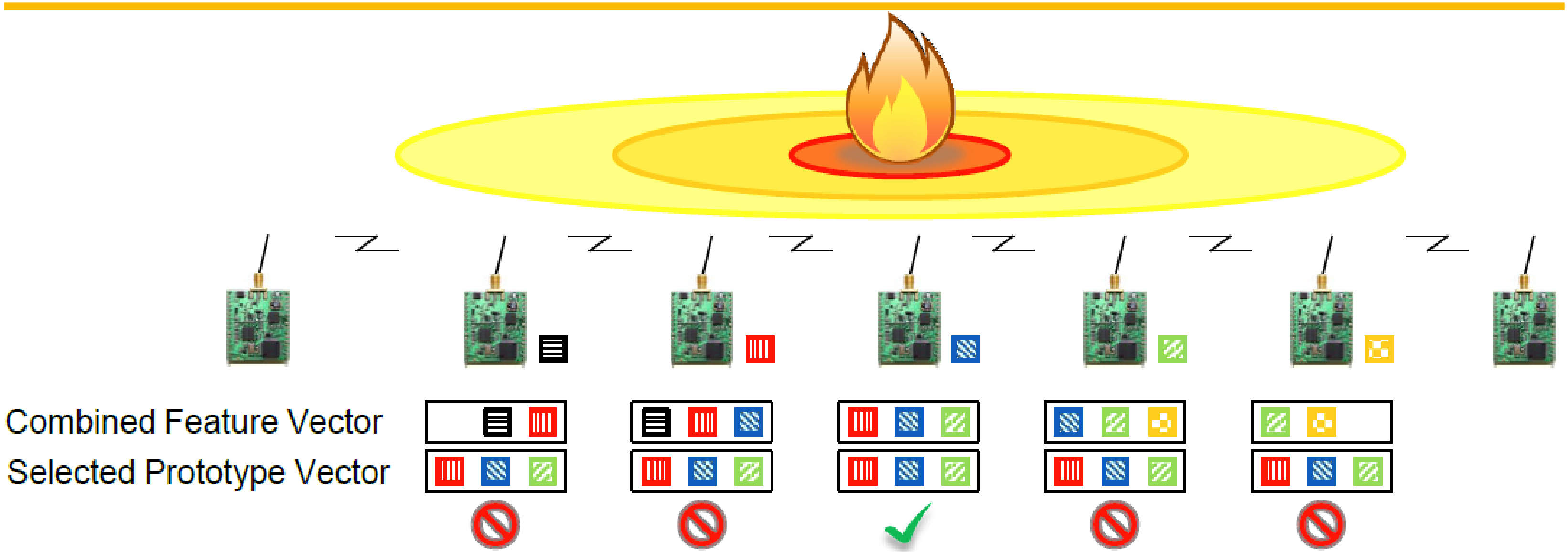
Feature Selection

- Leave-one-out Cross Validation (LOOCV):
  - Iterati
    featur
    vector
  - Calcu
    vector
    vector
  - Check
    error (
    using
  - Iterate
    vectors to pick, average
    classification errors

- Feature selection algorithm:
  - empty set of
  - select one
    th largest
    in LOOCV
  - eature to set
    d features
  - ntil no
    feature
    results in noteworthy
    reduction of error



Example



- Setup:
  - Nodes in a line, one feature per node
- Nodes are configured to recognize one single event
  - Identified by prototype vector with three features from: the left, local, and right nodes
- Event detection (all nodes):
  - Sample and process raw data
  - Extract feature
  - Distribute and calculate feature vector
  - Perform classification
- Central node detects (and reports) event; other nodes ignore event

Experimental Evaluation --- Setup

- Sensor nodes attached to fence of construction site
  - One node per fence element (3.5m wide, 2m high)
- ScatterWeb MSB sensor node:
  - TI MSP430 16-bit microcontroller (5 KB RAM, 55 KB flash)
  - ChipCon 1020 radio transceiver (operating at 868 MHz)
  - Freescale Semiconductor MMA7260Q 3-axis accelerometer
- Four different events:
  - Trained and evaluated with 15 samples per event at the same location

Experimental Evaluation --- Events

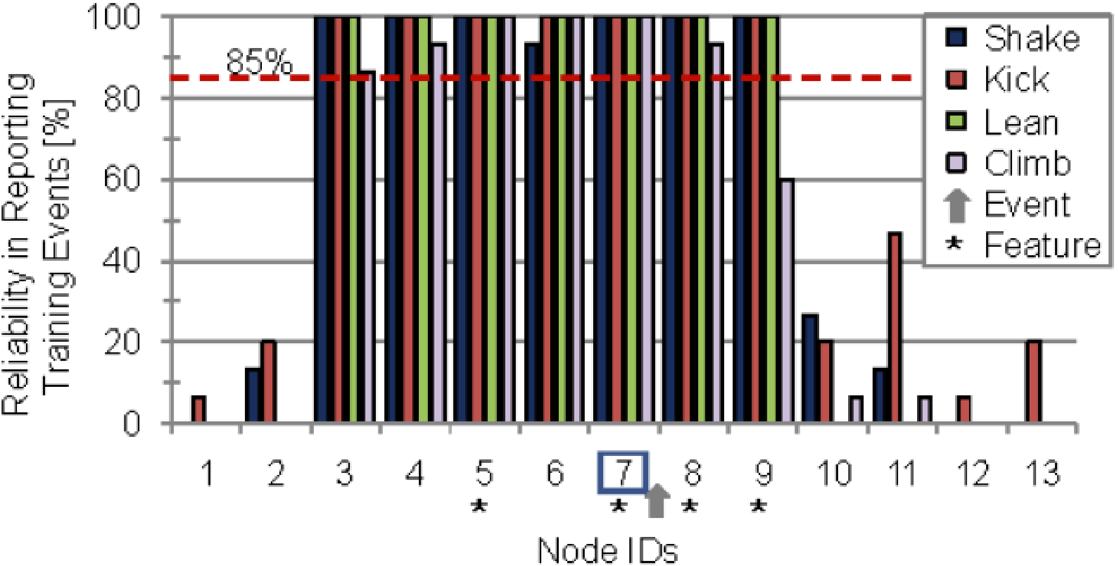
Shake

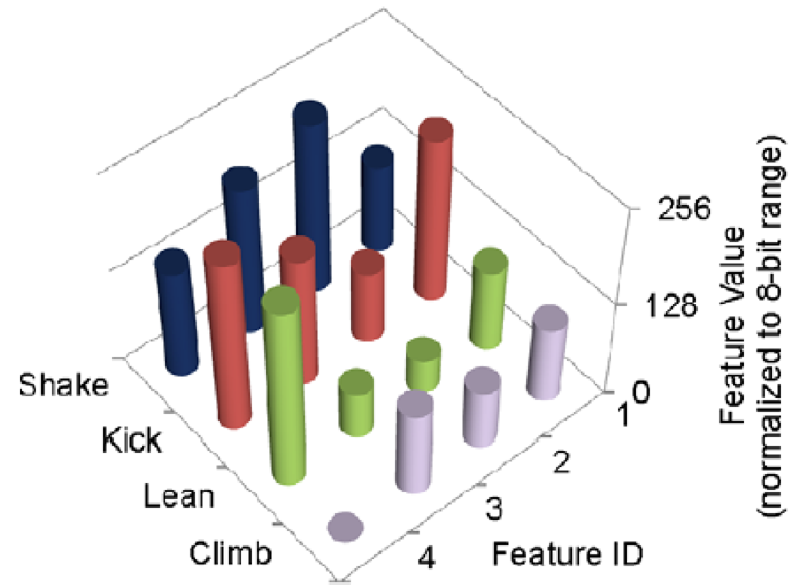
Kick

Lean

Climb

Results --- Feature Selection



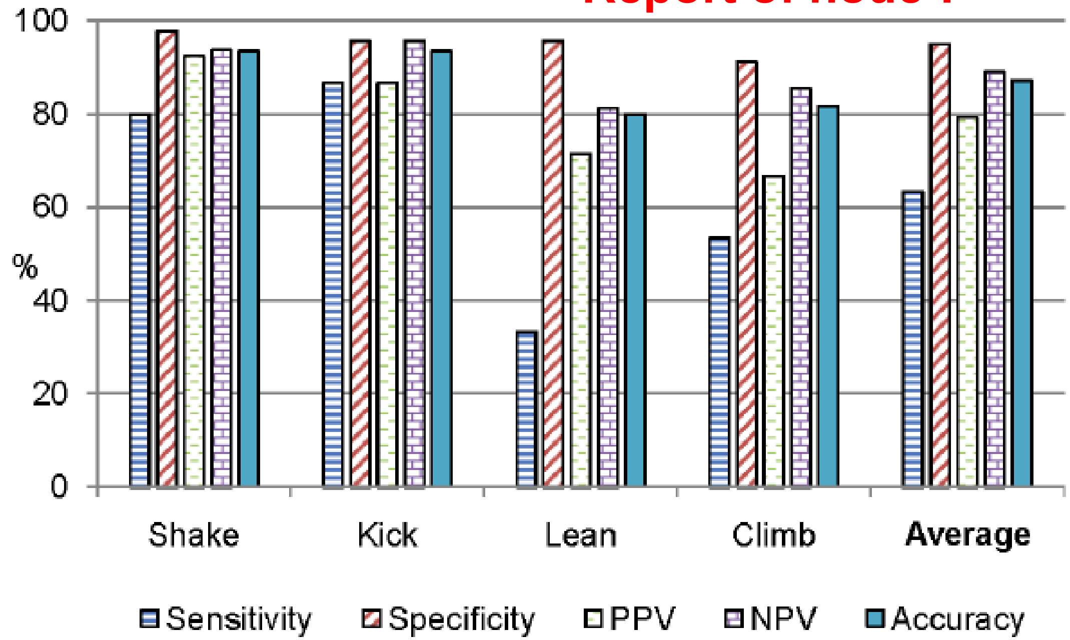


- Seven nodes were reproducibly affected by events
- Features from nodes #3 to #9 are deemed reliable enough
  - >85% reports
- Quality-based feature selection results in 4 features
- Selected features:
  - ID #1: Histogram-based feature from node #5
  - IDs #2 to #4: intensity features from nodes #7~ #9

Experimental Evaluation --- Metrics

- Sensitivity = TP / (TP+FN)
  - Proportion of correctly detected events in all events of that type
- Specificity = TN / (TN+FP)
  - Proportion of correctly ignored events in all events of another type
- Positive Predictive Value (PPV) = TP / (TP+FP)
  - Proportion of correctly detected events in all detections of that type

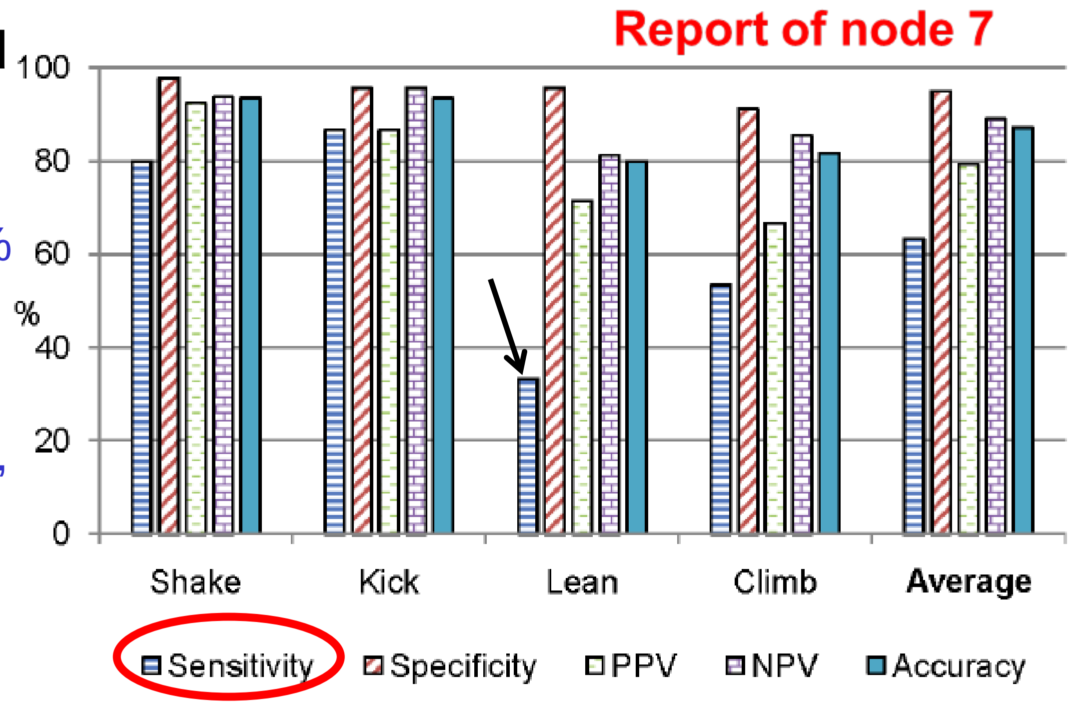
Report of node 7



- Negative Predictive Value (NPV) = TN / (TN+FN)
  - Proportion of correctly ignored events in all detections of another type
- Accuracy = (TP+TN) / (TP+TN+FP+FN)
  - Proportion of true results in the population

Experimental Evaluation --- Feature Fusion

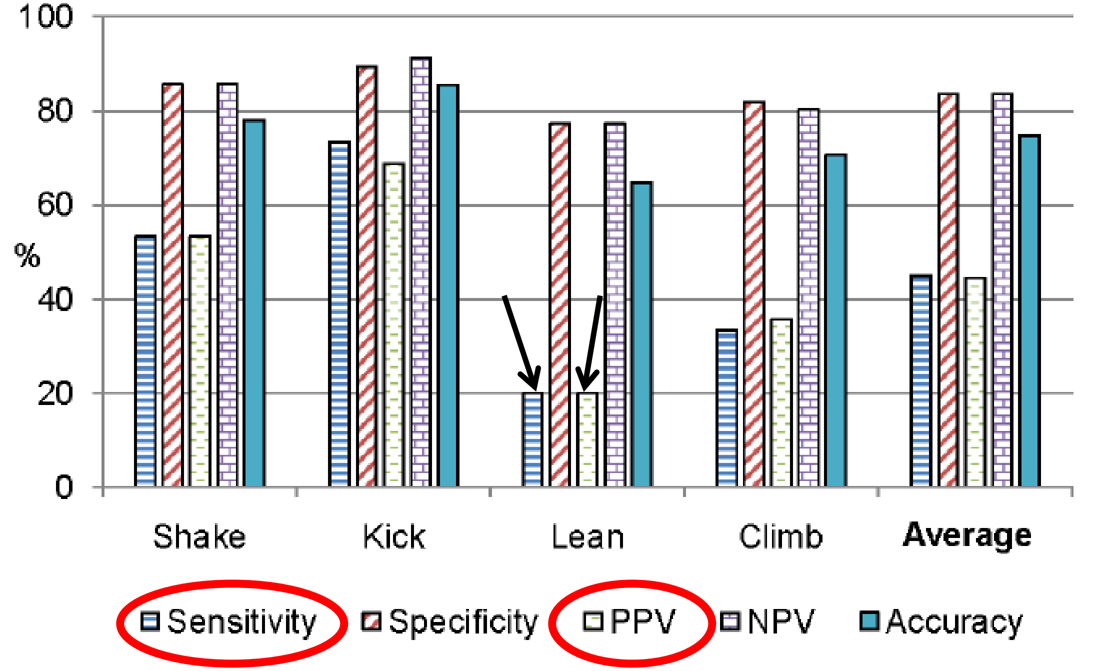
- Shake and kick events detected reliably
  - All metrics above 80%, shack and kick has accuracy of 93.3%
- Detection of lean or climb events not as accurate
  - Sensitivity is comparatively low, while specificity remains high
  - Too many events are falsely rejected due to prototype regions being too small
  - Training runs were too similar to each other, prototype regions only enclose part of required space
- Overall accuracy of 87.1% for all events



- Sensitivity = TP / (TP+FN)
- Specificity = TN / (TN+FP)

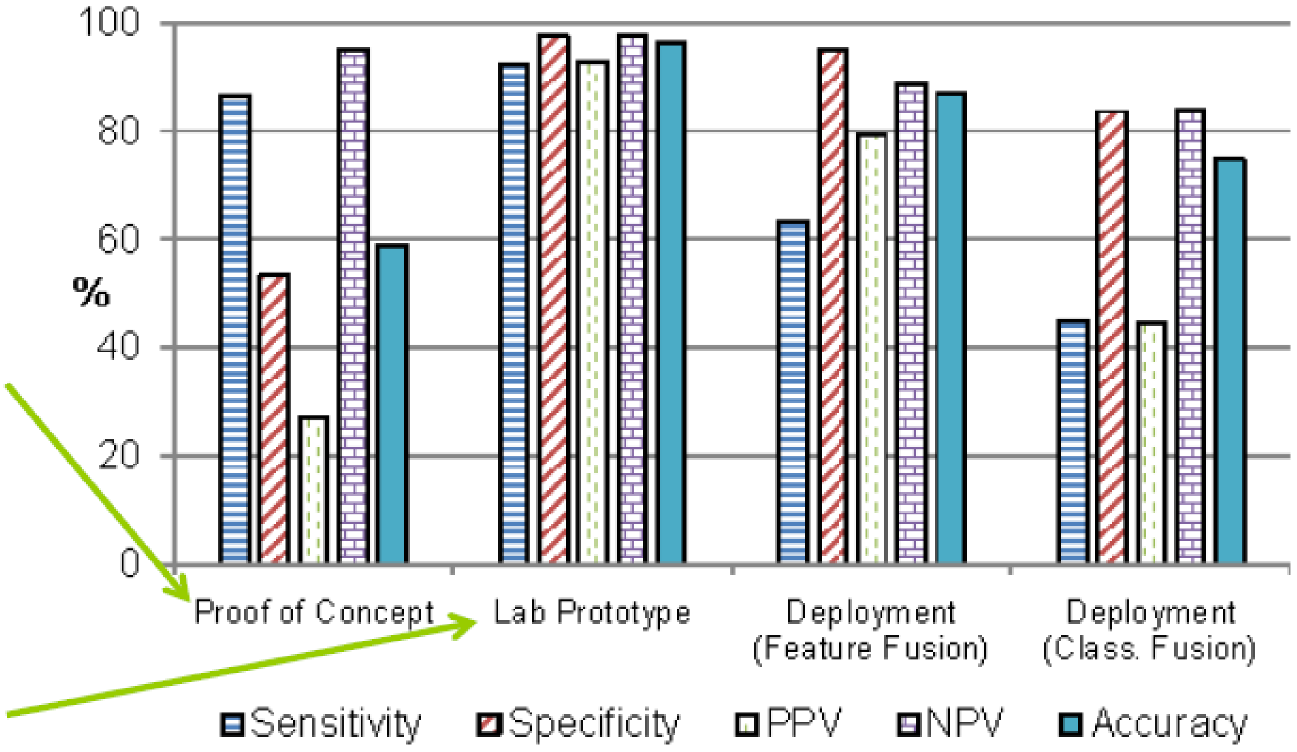
Results --- Classification Fusion

- Sensitivity and PPV decrease considerably
  - Sensitivity=TP/(TP+FN)
  - PPV=TP/(TP+FP)



- Classification Fusion: base station counts incorrect classification if
  - correct classification is falsely rejected on central node 7, while incorrect classification is reported from another node
  - node reports incorrect classification with higher confidence (shorter distance to prototype vectors) than that of correct classification
- Overall accuracy of 74.8%

Comparison with Prior Work



- Improvement over proof-of-concept implementation
  - Improvement of 28.8% (feature fusion, classification fusion was not supported)
- Unable to reach same level of accuracy as lab experiments
  - Manual feature section, accuracy of 96.3%

Conclusion

- Build a system for distributed event detection in WSNs
  - No external coordination or processing required
  - Trainable to detect different classes of application-specific events
- Has real deployment and the event detection accuracy shows improvements over prior work.
- But, further performance improvement is still needed in future.

Discussion

- How to make the design more energy efficient?
  - Reduce sampling, duty cycling …
- This is a general solution that has two common assumptions about the setup of the deployed WSN. What are they?
  - Sensor nodes must be placed uniformly within the deployment area
    - Is this really enough?
  - The physical effects of the events to be observed must propagate evenly in all parts of this area.
    - Is this really needed?
- Do we need to conduct training in multiple locations?