Intelligent Protocols Based on Signal Change Detection

Leon Reznik, Gregory Von Pless, Tayeb Al Karim

Department of Computer Science, Rochester Institute of Technology, Rochester
NY 14623-5608
Sensor Networks

- Sensor networking is an emerging technology.

- In sensor networks, we have small sensor nodes which are densely deployed in an area.

- The sensor nodes equipped with transceiver such that they can send and receive information from neighboring nodes in the form of a wireless network.

- Main problem: limited resources
Sensor nodes: MICA Mote by Crossbow Inc. and Telos rev. B by MoteIV

- Humidity
- Temperature
- Light
- 802.15.4 Compliant
  - 50m range
- USB Capable
- TinyOS
- 10kB RAM
- 8MHz processor
Main Idea: New sensor networks protocols with CONTENTS BASED efficiency, security, and reliability enhancement

What can we get from it?
- data compression for communication
- security enhancement
- reliability improvement
- accuracy increase

How?

Idea: measurement results (or sensor signal) change detection
Security enhancement

• 1. Detection of the malicious measurement result change or an addition of a new node
• 2. Alerting the administrator
• 3. Possible correction of the malicious change
Reliability improvement

- 1. Possibility of detecting measurement instrument big error or malfunctioning (big error here is defined as an error which is significantly bigger than a normal measurement error)
- 2. Possibility of correcting a sensor big error
Idea No. 1: Comparison with the threshold

Problems:
- hard to find the right value
- generally low reliability, which can be improved by a priori study of the signals and environment
- Sensor network specific: a long priori study is not possible in many applications
Alarm Application - Design

1. User Sets the Thresholds

2. Telos Rev B Motes sends Temperature, Humidity and Light information (Wireless)

3. Data to computer via USB

4. Alerts sent via email
Idea No.2 to improve detection reliability: Compare measurement results against association information

Sensor networks specific:
sensor signals networking

\[ \text{measure2} \sim \text{aver.} (\text{measure1, measure3}) \]
\[ \text{measure3} \sim \text{measure4} \]
Was this smart enough?

Protocol implementation: software
Sensor Network Anomaly Detection System (SNADS)

- designed to become modular, extensible, robust, scalable and portable
- versatile cross platform tool
- modularity is mainly achieved via a central signaling system
- components are replaced and added on the fly: achieve scalability
- database subsystem provides a simple interface for data logging and searching
- intelligent agents generate new association information and modify existing one
- anomaly detection: detect and possibly correct measurements
Screenshot of SNADS
Agent No. 1: Artificial Neural Networks

- The artificial neuron is a mathematical construct that emulates the more salient function of biological neurons, namely this signal integration and threshold firing behavior. Just as in the biological case, such neurons are bound together by various connection weights that determine how the outputs from one neuron are to be algebraically weighted before arriving at receiving neurons. The intelligence within these collective structures of artificial neurons (i.e., ANNs) is stored within these sundry algebraic connection weights.
All of the information stored within an artificial neural network (i.e., its virtual computer programs) takes the form of connection strengths between neurons. These are values by which the signals from one artificial neuron to another are multiplied before being summed up within the receiving neuron. Important to note is that these weights are not 'hand wired' into these networks by computer nerds. Instead, special computer programs mathematically 'spank' the net until it consistently yields the correct outputs for any given set of inputs.
ANN

- **ANN Training** - we successively apply all known inputs to the net (here the Exclusive Or data) propagating signals in the forward direction, observe network output, and then backwardly propagate corrections to the respective connections in the net. We continue this process until the net yields the correct output for all known test cases. At this point we say that we have a neural network model of some conceptual space. Applications of inputs unencountered during the network's training phase should yield reasonable estimates for network outputs (i.e., the model's predictions). The most important aspect of this process is that the network discovers on its own what the underlying rules actually are.
ANNs are able to GENERATE knowledge - Artificial Neural Networks have taken a rap for being 'black boxes'. That is, they give the right results, but don't explain why they do so. In reality, they internally develop connection traces that embody the rules behind the conceptual space they are training on. Here we see a network learning three implicit rules hidden within a database of numbers.
Smart change detection introduction (continued)

• Method summary
  – Based on function prediction
  – Predicted sensor values are compared to actual next values
  – Uses a modified Multi-Layer Perceptron for prediction
How to make ANN train and work faster under limited resources? (WSN question)
How to make ANN train and work faster under limited resources? (WSN question)
Network Specifications - Introduction

- The modified time-based MLP can be broken up into two parts: the TimeNets, and the MainNet, as shown below.
Network Specifications - Overview

- Each TimeNet acts as a time-based MLP for a single sensor
- Network learns correlations between sensors while separating some knowledge of individual sensors
- Training is performed using backpropagation
Experiments

- **MLP vs. MTBMLP**
  - Ran 4 identical sine curves through both MLP and MTBMLP

- **Uncorrelated Detection**
  - 1800 tests (3 types * 5 thresholds * 4 training times * 30 attempts)
  - 4 curves, each “randomly” generated
  - 1 curve changes – either flat lines, changes frequency, or changes amplitude
  - Change occurs after 4000 iterations; total iterations = 8000

- **Correlated Detection**
  - 2400 tests (2 types * 2 curve sets * 5 thresholds * 4 training times * 30 attempts)
  - 6 curves, in equivalent pairs
  - 2 or 3 curves change
  - Change occurs after 4000 iterations; total iterations = 8000
Results – MLP vs. MTBMLP
Results – MLP vs. MTBMLP

• Learns more quickly than a standard time-based MLP with same inputs and topology
• TimeNet structure eliminates a good deal of weights, making all operations on the network faster
• MTBMLP is more robust – when one or two sensors are lost, other sensors are still properly predicted
Results – Uncorrelated Detection
Results – Correlated Detection
Application

• The problem
  – Novelty in light intensities
  – On / Off
  – Flickering
Application - 2
Application - 4

![Image of a software interface showing graphs and data]

- **Sensor Values and Predictions**
  - Fluctuations are likely caused by interference on the sensor board.

- **Error Values and Predictions**
  - Error generated here is far larger than in the fluctuating fluctuations.

Here the flicker rate changes, no novelty is detected.
Application - 5

Note that the network learns the general trend of the sensor, even if it cannot predict the erratic values.

Here we see the initial error caused by the lights beginning to flicker.

Fluctuations in flicker rate most likely caused those errors.