Pattern Matching and Detection in Extremely Resource-Constrained Wireless Sensor Networks

Michael Zoumboulakis

Birkbeck College, University of London

March 2013

Birkbeck
UNIVERSITY OF LONDON

londonknowledgelab
exploring the future of learning with digital technologies

Outline

Outline

Outline

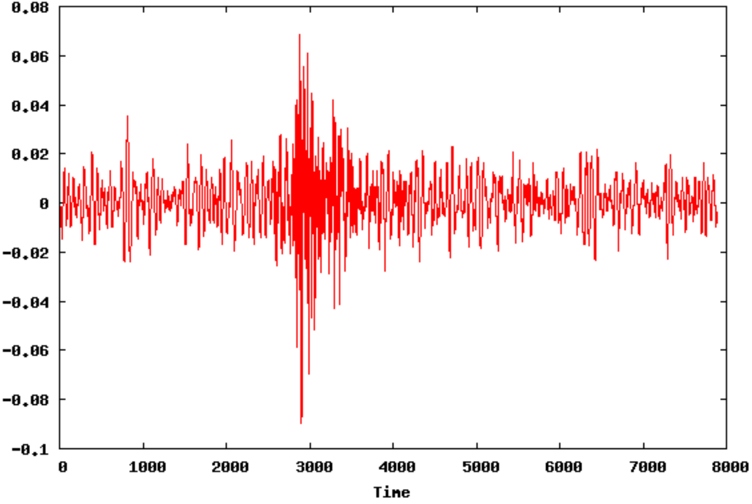
Outline

Outline

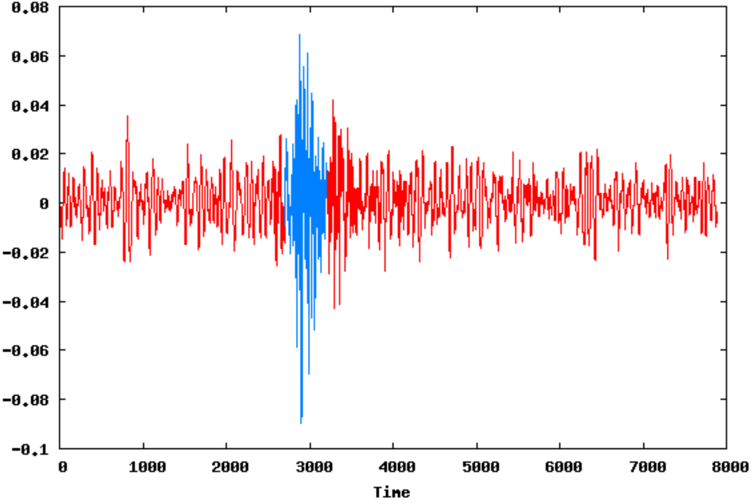
Introduction

Pattern: An ordered list of sensor observations revealing interesting or unusual activity in the monitored object.

Example


0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
1000
2000
3000
4000
5000
6000
7000
8000
Time

Example


0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
1000
2000
3000
4000
5000
6000
7000
8000
Time

Characteristics

- Lists of sensor observations
- *Pattern Matching* refers to the problem of finding occurrences of one or more user-submitted template patterns.
- *Pattern Detection* refers to the problem of discovering sustained changes or local anomalies.

Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.

Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
  - Precision Agriculture, Volcanic Monitoring, SHM, Pervasive Healthcare, Disaster Detection, etc.

Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
  - Precision Agriculture, Volcanic Monitoring, SHM, Pervasive Healthcare, Disaster Detection, etc.
- Only a few in-network approaches are available.

Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
  - Precision Agriculture, Volcanic Monitoring, SHM, Pervasive Healthcare, Disaster Detection, etc.
- Only a few in-network approaches are available.
- The alternative of using thresholds and a programmatic description of patterns as series of logically combined conditions is neither efficient nor scalable.

On-line data-mining

Central Idea: Use a well-understood & established algorithm for the online conversion of sensor readings to characters. Reduce matching and detection to a string matching problem.

Why?

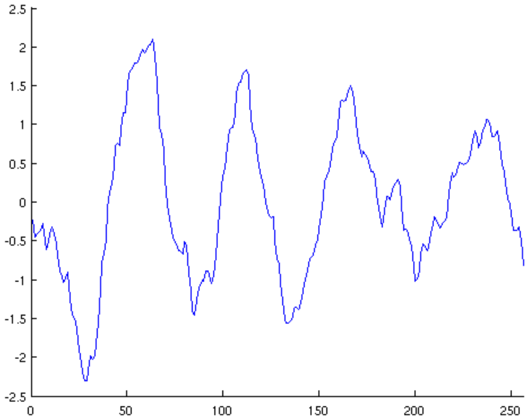
On-line data-mining

Central Idea: Use a well-understood & established algorithm for the online conversion of sensor readings to characters. Reduce matching and detection to a string matching problem.

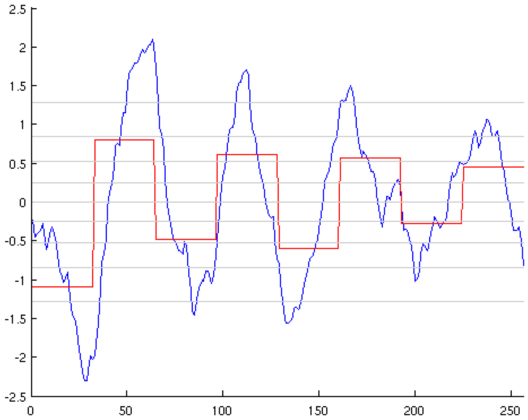
Why?

- Vast collection of algorithms.
- Rich Pattern Detection Capabilities.
- Scalable & Resource-efficient solution.

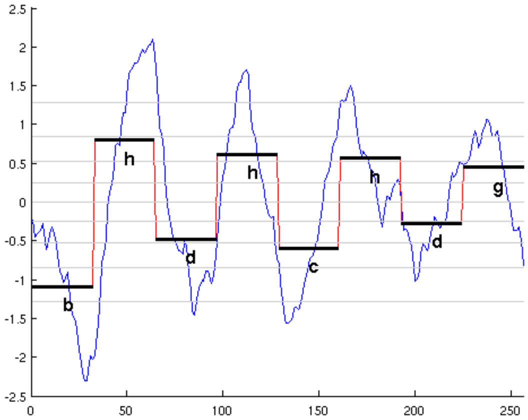
Symbolic Aggregate approXimation (SAX)



Symbolic Aggregate approXimation (SAX)



Symbolic Aggregate approXimation (SAX)



Contributions

- Developed temporal domain in-network Pattern Matching and Detection algorithms that reduce network communication.
- Demonstrated that the algorithms are suitable for extremely resource-constrained WNSs by integer arithmetic refactoring and dynamic sampling frequency adjustments.
- Introduced an iterative algorithm, based on a geometric computation and a Kalman filter that detects spatial patterns inside the network in a collaborative manner.

Exact and Approximate or Non-parametric Detection

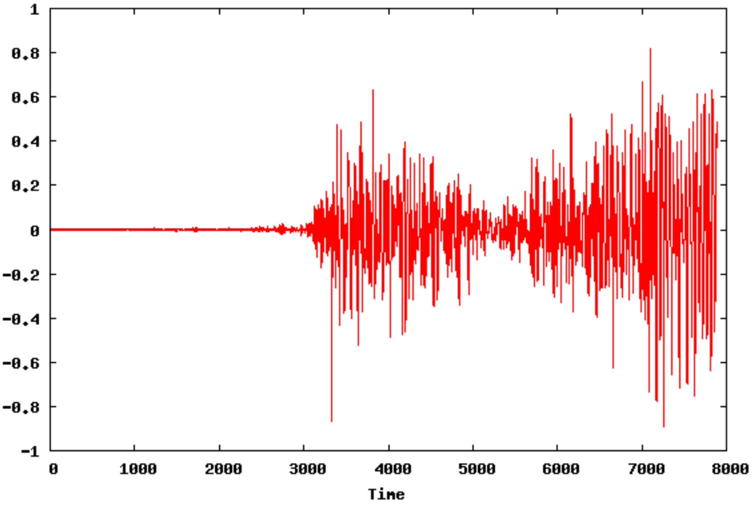
- Exact matching is performed when a user is interested in a specific pattern.
- Approximate matching is performed when a user is interested in a pattern and other patterns similar to it.
- Non-parametric matching is performed when a user does not provide a pattern.

Multiple Pattern Matching (using Suffix Arrays)

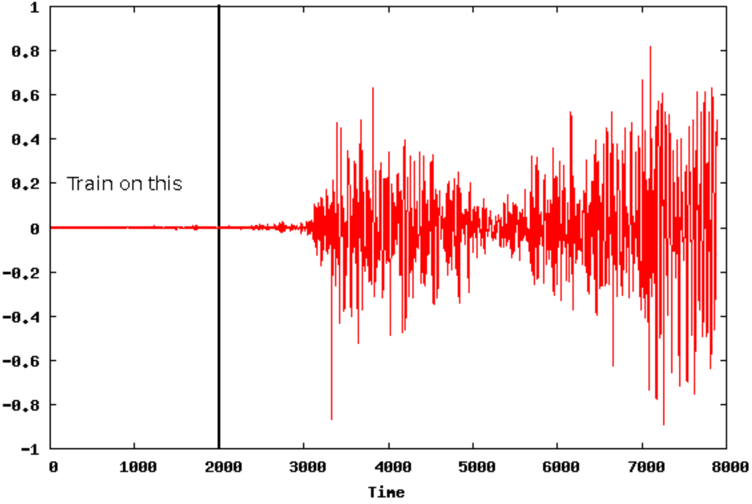
Store all interests in a suffix array: speeds up searches & makes the solution scalable.

- ...
- efcghaaagdbeabc
- fcghaaagdbeabc
- gdbeabc
- ghaaagdbeabc
- haaagdbeabc
- ...

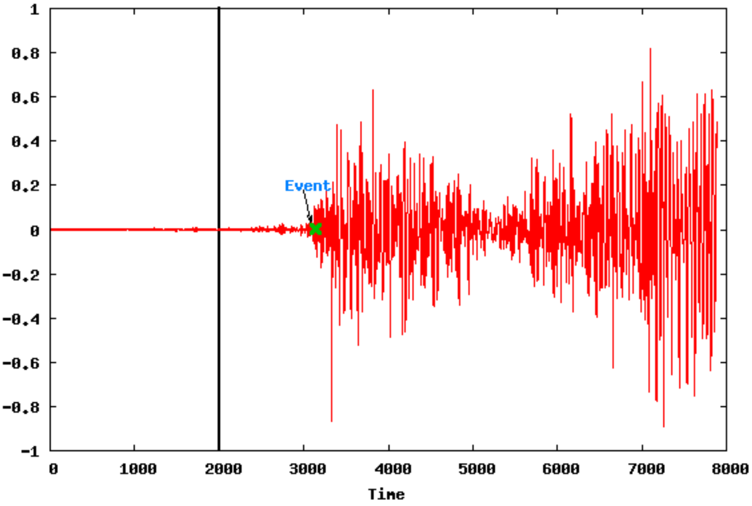
Non-parametric Detection



Non-parametric Detection



Non-parametric Detection



Dynamic Sampling Frequency Adjustments

- Many WSN nodes exhibit the *boredom punctuated by panic* property.
- Sampling at a constant rate when very little happens is wasteful.
- Instead, adjust sampling frequency by continuously monitoring string distances.

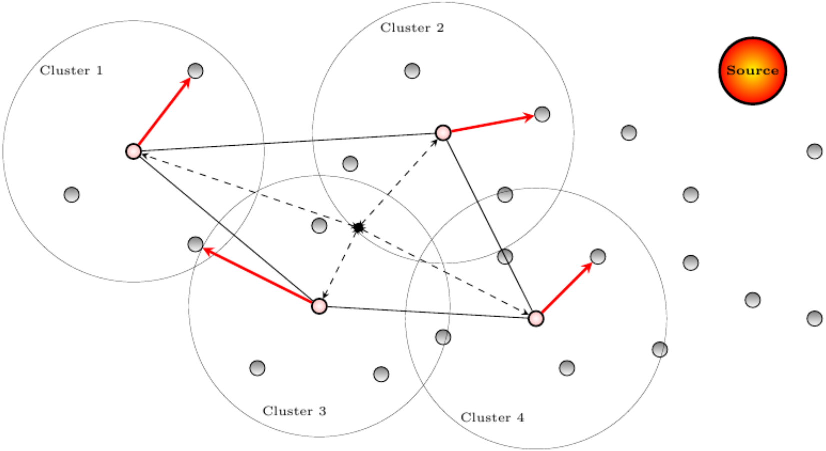
Probabilistic Detection

- The process that generates characters can be viewed as a Markov Chain of desirable order (depending on the app requirements).
- Use a portion of the time series that is known to be normal to generate transition probabilities.
- Allows to compute probabilities for various sequences.

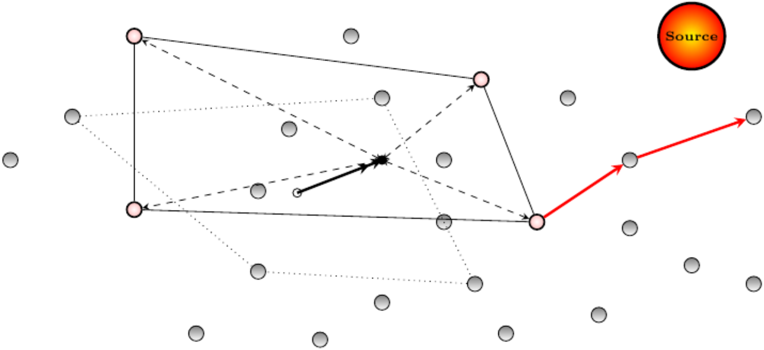
The Spatial Pattern Detection Problem

We considered the problem of estimating the location and intensity of "dirty bomb" dispersion attacks inside the network.
Also known in literature as the *Inverse Problem* where dispersion parameters must be estimated from sensor-observed data.

An example



An example



Evaluation

The spatial detection algorithm was evaluated through simulations and detection accuracy was satisfactory but heavily dependent on node placement and topology.

The work on spatial detection opened a number of future work directions:

- Alter the gas dispersion model and consider multiple sources.
- Evaluate against competitive techniques such as TDOA.
- Implement the algorithm for WSNs.

Selected Results from Emulation

- We compared our temporal detection algorithm to two competitive approaches — Exponential Weighted Moving Average (EWMA) and Real-time Seismic Amplitude Measurement (RSAM).
- We found improved accuracy (true positive ratio of 92.7%) and a reduction of false positives (a ratio of 3.8%).
- We found detection accuracy degrading gracefully in relationship to decreasing SNR.
- For some time-critical applications we established that the average detection latency is low (0.086 secs in a physiological data set).

Selected Results from Deployment

- An integer only implementation showed a factor of ten runtime improvement in comparison with a FP implementation.
- The Dynamic Sampling Frequency Adjustment algorithm resulted in 64% fewer timer ticks in comparison with a fixed sampling scheme.
- The execution profile was sufficiently low to support use in a new class of energy harvesting nodes powered from RFID readers.

Software Timing Model

	Arithmetic (Integer)		
Operation	*16-bit*	*32-bit*	*64-bit*
Increment	6,734	12,824	67,271
Addition	7,955	15,319	88,301
Subtraction	7,969	15,310	87,985
Multiplication	145,020	159,060	316,755
Division	226,265	709,720	3,519,055
Remainder	225,375	706,875	3,540,080
	Bitwise		
AND	7,965	15,285	88,095
OR	7,985	15,325	88,360
XOR	7,955	15,315	88,310
SHIFT	56,735	62,880	573,665
	Floating Point Arithmetic		
Assignment & Cast	2,687,535		
Addition	3,438,530		
Subtraction	3,499,920		
Multiplication	4,841,490		
Division	4,041,785		
	Array Comparisons & Swaps		
Straight Comp	104,095		
Comp C Fcn	83,315		
Comp TOS Fcn	83,335		
Comp TOS task	159,055		
Swap C Macro	93,085		
Swap C Function	83,320		

Thank you!

Questions?