Detecting Time-Related Changes in Wireless Sensor Networks Using Symbol Compression and Probabilistic Suffix Tree

YuanYuan Li, Michael Thomason, and Lynne Parker

Distributed Intelligence Laboratory (DILab)
Department of Electrical Engineering and Computer Science
The University of Tennessee, Knoxville

A presentation for CISML
October 11, 2010
Outline

- Introduction
- Overall approach
- Experimental results
- Conclusion
Research objective

- Design an anomaly detection system using wireless sensor networks (WSNs) that
  - *Detect time-related changes*
  - *Operate in unknown environments*

- WSNs characteristics:
  - *Simple and inexpensive*
  - *Low dependability*
  - *Energy restricted*
Hierarchical networking and learning

- All nodes run the same program

- Sample commands for base station:
  - *Configure cluster setting*
  - *Control learning process*
  - *Control sensor setting*
Intruder detection application
Related works

- Time-series analysis in WSN
  - *Network traffic, e.g.,* [Huang, et al., 2007]
  - *Sensor value forecasting, e.g.,* [Borgne, et al., 2007]

- Regression models
  - *Auto Regression Moving Average (ARMA)*
  - *Least-Square-Error based linear forecasting method*

- Fixed length Markov models

- Hidden Markov Model (HMM)

- Variable Memory Markov (VMM) model [Ron, 1996]
  - *Probabilistic Suffix Tree (PST)*
  - *Probabilistic Suffix Automata (PSA)*
Fixed-length Markov model is expensive

### 1st-order matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>C</td>
<td>0.2</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td>E</td>
<td>0.7</td>
<td>0.3</td>
<td>0</td>
</tr>
</tbody>
</table>

### 2nd-order matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>E</th>
<th>AA</th>
<th>AC</th>
<th>AE</th>
<th>CA</th>
<th>CC</th>
<th>CE</th>
<th>EA</th>
<th>EC</th>
<th>EE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0.1</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AA</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AC</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AE</td>
<td>0.1</td>
<td>0.7</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CC</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CA</td>
<td>0.2</td>
<td>0</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CE</td>
<td>0.1</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>EA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>EC</td>
<td>0</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>EE</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The size of order L Markov models is exponential in L
**Time-series application: volcano monitoring**

Long sequence of inactivity makes Markov model expensive

*Time (hours) [Werner-Allen et al., 2006]*
Key contributions

- Developed sensor network system that:
  - *Detects anomalies without prior knowledge*
  - *Models time efficiently*
  - *Saves communication costs*
  - *Supports learning that is:*
    - Distributed
    - Scalable
    - Autonomous
    - Modular
    - Robust
    - Online
Approach

- Design a system to detect anomalies in an unknown environment

- The system:
  - *Uses an initial learning period*
  - *Learns a normal model during the training period*
  - *Treats variations of the normal model as anomalies*
Proposed sensor network system architecture

Root node

Clusterhead node

- Missing data estimator
- Classifier
- Symbol compressor
- Time analyzer

Cluster member node

- Classifier
- Symbol compressor
- Time analyzer

Raw sensor readings from the environment
Data modeling on sensor node

1. Sensors’ signals over time, $O$
2. Sequence of classes, $C$
3. Symbol compressor
   - Compressed sequence, $S$
4. Time model
   - Temporal model, $\lambda$

Two-step temporal modeling process:

- Classifier
- Two-step temporal modeling process
  - Symbol compressor
  - Time model

Introduction
Approach
Experiments
Conclusion
Fuzzy ART detects anomalous sensor data

Sensors’ signals over time, $O$

Sequence of classes, $C$

Symbol compressor

Compressed sequence, $S$

Time model

Temporal model, $\lambda$
Example: cluster member learning

Sensor reading

Category

C1: Light on
Buzzer on

C2: Light on
Buzzer off

C3: Light off
Buzzer on
Symbol compressor extracts semantics

Sensors’ signals over time, $O$

Sequence of classes, $C$

Classifier

Two-step temporal modeling process

Symbol compressor

Compressed sequence, $S$

Time model

Temporal model, $\lambda$
Data compression techniques

- **Lossy compression**
  - I.e., JPEG image compression rounds off “less important” information
  - May achieve higher compression
  - E.g., 25.888888888 = 26

- **Lossless compression**
  - Exploits statistical redundancy
  - Reversible
  - E.g., 25.888888888 = 25.[9]8
Extracting semantics from temporal sequences

- Lempel-Ziv-Welch (LZW) data compression [Welch, 1984]
  - Lossless
  - Table based
  - Online

- Given: \( C = \{c_1, \ldots, c_T\} \), LZW produces temporal sequence \( S \)
  - Scanning through the input string for successively longer substrings until it finds one that is not in the dictionary
  - Sent the longest substring that is in the dictionary to output
  - Add the current scanned substring to dictionary with next available index
LZW compression example

\[ C = \{1,2,2,2,3,3,1,2,3,2,2,2,1,3\} \]

<table>
<thead>
<tr>
<th>Index</th>
<th>Entry</th>
<th>Index</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>(code)</td>
<td>(categories)</td>
<td>(code)</td>
<td>(categories)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>7</td>
<td>33</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>9</td>
<td>123</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>10</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>11</td>
<td>2222</td>
</tr>
<tr>
<td>5</td>
<td>222</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>

- **Entries** = Original sub-sequences
- **Indices** = New semantic classes

Post LZW dictionary pruning eliminates:
- Non-meaningful temporal features (manually)
- Entries that are smaller than the Markov order (redundant)
Modeling semantic class sequences

Introduction

Approach

Experiments

Conclusion
Probabilistic Suffix Tree (PST)

- PST uses a suffix tree as the index structure [Ron et al., 1996]
- PST is based on the memory (i.e., order) $M$ of natural sequences
  - The root node gives the empirical probability of each symbol in the alphabet
  - Each node at subsequent levels is associated with a vector that gives each symbol probability given the label of the node as the preceding segment
    - $P(s_{i+1} | s_0...s_i) = P(s_{i+1} | s_{i-M}...s_i), i > M$
      - $S$ is the semantic symbol
      - $M$ is the memory length
- PST parameters are estimated based on the Maximum-Likelihood criteria
An example: PST model construction

Class labels from time 1 to time 9:
S={1, 2, 3, 1, 2, 3, 2, 1, 3}

Order-2 PST (M=2)
Using Universal Background Model (UBM) likelihood-ratio detector to detect anomalies

- **Formulation:**
  - $H_0 : \hat{S}$ is normal
  - $H_1 : \hat{S}$ is abnormal
  - **Likelihood-ratio test**
    - $p(\hat{S}|H_0) / p(\hat{S}|H_1) \geq \Theta$ accept $H_0$
    - $p(\hat{S}|H_0) / p(\hat{S}|H_1) < \Theta$ reject $H_0$
  - **Log-likelihood-ratio test**
    - $\Lambda(\hat{S}) = \log p(\hat{S}|H_0; \lambda_0) - \log p(\hat{S}|H_1; \lambda_1)$ where $\lambda$ denotes PST models

What are these models?

- **Ideal:**
  - ratio = normal / abnormal
- **Problem:**
  - Unknown environments
- **UBM solution:**
  - ratio = normal / universal

[Reynolds, 1997]
Evaluation method

- Use volcano dataset [Werner-Allen et al., 2006]
  - Seismic data collected over 24 hours from Volcano Reventador

- Compare the fixed order Markov models vs. the PST models

- Compare PST models with different memory length that are built from compressed sequences

- Collect statistics on
  - Model size
  - Compression ratio
    - compressed size/uncompressed size
  - Miss rate
  - False alarm rate
**PST model uses less space**

<table>
<thead>
<tr>
<th>Model Order</th>
<th>Number of Nodes</th>
<th>Negative Log-Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Length Markov</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>-0.0141</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>-0.0112</td>
</tr>
<tr>
<td>3</td>
<td>104</td>
<td>-0.0092</td>
</tr>
<tr>
<td>4</td>
<td>190</td>
<td>-0.0078</td>
</tr>
<tr>
<td>5</td>
<td>296</td>
<td>-0.0069</td>
</tr>
<tr>
<td>10</td>
<td>981</td>
<td>-0.0046</td>
</tr>
<tr>
<td>PST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>-0.0141</td>
</tr>
<tr>
<td>2</td>
<td>41</td>
<td>-0.0112</td>
</tr>
<tr>
<td>3</td>
<td>76</td>
<td>-0.0092</td>
</tr>
<tr>
<td>4</td>
<td>116</td>
<td>-0.0078</td>
</tr>
<tr>
<td>5</td>
<td>146</td>
<td>-0.0069</td>
</tr>
<tr>
<td>10</td>
<td>205</td>
<td>-0.0046</td>
</tr>
</tbody>
</table>

*PST models are preferred, since it takes less space*
The average compression rate is 1:33

<table>
<thead>
<tr>
<th>Entry (categories)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>155</td>
</tr>
<tr>
<td>6115</td>
</tr>
<tr>
<td>5122</td>
</tr>
<tr>
<td>21211</td>
</tr>
<tr>
<td>1111111122</td>
</tr>
<tr>
<td>211111115</td>
</tr>
<tr>
<td>51511</td>
</tr>
<tr>
<td>111112111111</td>
</tr>
<tr>
<td>11111111111111111111111111111111...1111111115</td>
</tr>
<tr>
<td>51111111</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Utilize ROC curves to decide performance trade-offs

**Order-5 PST model (M=5)**

- Compression + PST = good performance
- Order-10 PST better than order-5 PST

**Order-10 PST model (M=10)**
**UBM likelihood-ratio detectors have good performances**

<table>
<thead>
<tr>
<th>PST order</th>
<th>Threshold $\Theta$</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=5</td>
<td>0.5248</td>
<td>87.69%</td>
<td>77.19%</td>
<td>88.83%</td>
</tr>
<tr>
<td></td>
<td>0.5375</td>
<td>90.03%</td>
<td>74.85%</td>
<td>91.67%</td>
</tr>
<tr>
<td></td>
<td>0.5502</td>
<td>91.40%</td>
<td>70.76%</td>
<td>93.62%</td>
</tr>
<tr>
<td></td>
<td>0.5629</td>
<td>92.42%</td>
<td>64.91%</td>
<td>95.39%</td>
</tr>
<tr>
<td></td>
<td><strong>0.5756</strong></td>
<td>93.85%</td>
<td>56.14%</td>
<td>97.92%</td>
</tr>
<tr>
<td>M=10</td>
<td>0.4911</td>
<td>90.83%</td>
<td>100.00%</td>
<td>89.84%</td>
</tr>
<tr>
<td></td>
<td>0.5039</td>
<td>94.13%</td>
<td>98.25%</td>
<td>93.69%</td>
</tr>
<tr>
<td></td>
<td>0.5167</td>
<td>95.50%</td>
<td>98.25%</td>
<td>95.20%</td>
</tr>
<tr>
<td></td>
<td><strong>0.5295</strong></td>
<td>97.04%</td>
<td>96.49%</td>
<td>97.10%</td>
</tr>
<tr>
<td></td>
<td><strong>0.5424</strong></td>
<td>97.49%</td>
<td><strong>88.89%</strong></td>
<td><strong>98.42%</strong></td>
</tr>
</tbody>
</table>

- Based on the ROC curve:
  - $\Theta=0.5756$ is chosen for order-5 PST model
  - $\Theta=0.5424$ is chosen for order-10 PST model

- ROC curve is a good indicator for threshold parameter tuning

*PSTs built from compressed sequence has high accuracies*
Conclusion

The system uses:

- Lempel-Ziv-Welsh (LZW) symbol compression technique to extract high level temporal semantics
- Probabilistic Suffix Tree (PST) to model time sequences
- A Universal Background Model (UBM) likelihood-ratio detector to detect time related anomalies

Our approach:

- Detects anomalies without prior knowledge
- Models time efficiently
- Extracts the semantics in temporal sequences
- Enables WSNs to save transmission power
- Saves communication costs
- Supports learning that is: distributed, scalable, autonomous, modular, robust, and online