BANBAD – A Bayesian-Networks-Based Anomaly Detection Algorithm for Mobile Networks

Chaoli Cai

Wireless Sensornet (WiSe) Laboratory
Department of Computer Science
Western Michigan University
Kalamazoo, MI, 49008, USA

Outline – Anomaly Detection

• Motivation
• Problem Description
• Previous Work
• Assumptions
• Proposed Algorithm
• Simulation Results
• Contributions
• Future Work

Motivation

• Characteristics of mobile networks
  – Mobile nodes have no fixed infrastructure
  – Arbitrary node movement
  – Lack of centralized control
• Prevention vs detection
• Misuse detection vs anomaly detection

Anomaly Detection
Problem Description

- Crucial features in mobile networks
  - Average velocity,
  - Power consumption,
  - Local computation,
  - Communication,
  - Response time,
  - ...
- Any of the above feature might be anomalous at any time

Problem Description (Cont’d)

<table>
<thead>
<tr>
<th>1st feature</th>
<th>2nd feature</th>
<th>...</th>
<th>Nth feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>7</td>
<td>...</td>
<td>CV</td>
</tr>
<tr>
<td>CV</td>
<td>CV</td>
<td>...</td>
<td>7</td>
</tr>
<tr>
<td>CV</td>
<td>7</td>
<td>...</td>
<td>CV</td>
</tr>
</tbody>
</table>

- Can we detect anomaly in a certain target feature? Can we detect anomaly in any feature? If the feature has some missing values, can we still detect anomaly?
- Evidence is an event that can be observed. Intuitively, we might be able to derive the behavior for a certain target feature from the evidence. But how?
- Evidences are divided into the true positive and the true negative cases. How to distinguish them?
- In order to improve the detection performance in terms of false alarm rate and detection rate, can we determine the design parameter, threshold?
- Do we need to combine all the evidences? Or is there any way to improve the overhead?

Previous Work

- Cai et al, GlobeCom 2006, Mahalanobis distance is used for similarity of mobility patterns.
**Assumptions**

- Centralized scenario – profile is stored in a secret, hard to be compromised node (base station)
- Nodes can be compromised – all secret information associated with the compromised nodes is open to attackers
- All nodes have fairly regular behaviors within certain time period – i.e., exclude total random behavior
- Complete raw dataset for profile generation during training process

**Proposed Algorithm**

- Based on Bayesian Networks (BN)
  - Joint probability distribution modeled as a directed acyclic graph (DAG)
  - Knowledge structure
  - Computational architecture

![Bayesian Network Diagram]

**Bayesian Networks**

- Bayes rule
  \[
  P(A_i|E) = \frac{P(E|A_i)P(A_i)}{\sum_i P(E|A_i)P(A_i)} = Bel(A_i)
  \]
- Based on definition of conditional probability, we have:
  - \( P(A_i|E) \) is posterior probability given evidence \( E \);
  - \( P(A_i) \) is the prior probability of range \( i \) of feature \( A \);
  - \( P(E|A_i) \) is the likelihood of the evidence given \( A_i \);
  - \( P(E) \) is the marginal probability of evidence \( E \);
Bayesian Networks (cont’d)

- Propagation process – one of the exact inference algorithm of BN
  - Bel(B) = p(B | e+) = α π(B) λ(B) where
  - π(B) = π(A) M(B | A), prior evidence;
  - λ(B) = M(C | B) λ(C), likelihood evidence;
  - α - normalized constant

- •: term by term product of two vectors
- •*: dot product of two vectors
- Bel(x) = p(x | e), the posterior

BN Propagation Example

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B2</td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
</tr>
</tbody>
</table>

M(A-B) = [0.8 0.2] [0.7 0.3] [0.5 0.4 0.1]

M(A-C) = [0.9 0.1 0.9 0.1 0.7 0.3 0.5 0.4 0.1]

Profile Generation

Sample Chain – Setup (Training)

1. Set all lambdas to be a vector of 1’s; Bel(A) = α λ(A) π(A)

<table>
<thead>
<tr>
<th>A</th>
<th>Bel(A)</th>
<th>λ(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>A2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

2. π(B) = π(A) M(B | A); Bel(B) = α λ(B) π(B)

<table>
<thead>
<tr>
<th>B1</th>
<th>Bel(B)</th>
<th>λ(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>B2</td>
<td>0.27</td>
<td>0.27</td>
</tr>
</tbody>
</table>

3. π(C) = π(B) M(B | C); Bel(C) = α λ(C) π(C)

<table>
<thead>
<tr>
<th>C1</th>
<th>Bel(C)</th>
<th>λ(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>C2</td>
<td>0.35</td>
<td>0.36</td>
</tr>
<tr>
<td>C3</td>
<td>0.24</td>
<td>0.24</td>
</tr>
</tbody>
</table>
1st Propagation (Testing)

\[
\pi(A) \quad \text{Bel}(A) \quad \lambda(A)
\]
\[
\begin{array}{c|cc}
A1 & 0.8 & 0.8 & 1.0 \\
A2 & 0.2 & 0.2 & 1.0 \\
\end{array}
\]

\[
\pi(B) \quad \text{Bel}(B) \quad \lambda(B)
\]
\[
\begin{array}{c|cc}
B1 & 0.73 & 0.73 & 1.0 \\
B2 & 0.27 & 0.27 & 1.0 \\
\end{array}
\]

\[
\pi(C) \quad \text{Bel}(C) \quad \lambda(C)
\]
\[
\begin{array}{c|cc}
C1 & 0.39 & 0.3 & 0.5 \\
C2 & 0.35 & 0.5 & 1.0 \\
C3 & 0.24 & 0.2 & 0.6 \\
\end{array}
\]

2nd Propagation – Update beliefs of B

\[
\pi(B) = \pi(A) M_{B|A}
\]

\[
\pi(B) \quad \text{Bel}(B) \quad \lambda(B)
\]
\[
\begin{array}{c|cc}
B1 & 0.66 & 0.66 & 0.71 \\
B2 & 0.34 & 0.34 & 0.71 \\
\end{array}
\]

3rd Propagation – update beliefs of A & C

\[
\pi(A) = M_{B|A} \pi(A)
\]

\[
\pi(A) \quad \text{Bel}(A) \quad \lambda(A)
\]
\[
\begin{array}{c|cc}
A1 & 0.8 & 0.8 & 0.71 \\
A2 & 0.2 & 0.2 & 0.71 \\
\end{array}
\]

\[
\pi(C) = \pi(B) M_{C|B}
\]

\[
\pi(C) \quad \text{Bel}(C) \quad \lambda(C)
\]
\[
\begin{array}{c|cc}
C1 & 0.36 & 0.25 & 0.5 \\
C2 & 0.37 & 0.52 & 1.0 \\
C3 & 0.27 & 0.23 & 0.6 \\
\end{array}
\]

Compare computed belief of A & C after propagation with the profile.
Proposed Algorithm - BANBAD

- Structure learning techniques
  - K2[3] - uses a greedy search algorithm over the space of network topologies, is constrained by a given initial node ordering pattern to reduce the search space
  - PC[4] - uses local conditional independence tests between a set of models to determine the network topology

Note - We do not attempt to assess the actual learning structures, our interest lies in the predictive power of the models which is measured by their ability to perform accurate assessment of behaviors.

Proposed Algorithm (cont’d)

- DAG application model

- Conditional probability representation

\[ M(D | V) = \begin{bmatrix} P(D_2 | V_{i2}) & P(D_3 | V_{i3}) \\ P(D_2 | V_{i2}) & P(D_3 | V_{i3}) \end{bmatrix} \]

Proposed Algorithm (cont’d)

Training Process

Start

Traffic & Data Collection

Structure Learning

Profile Creation

Stop

Testing Process

Start

Feature & Data Collection

Evidence Extraction

Inference Algorithm

Model Updating

Belief Difference (Diff)

Diff > Threshold ?

Alert Generation

No Alert

Stop
Proposed Algorithm (cont'd)

- Prior evidence:
  \[ P(A) \pm \epsilon = [P(A) - \epsilon, P(A) + \epsilon] \]
  Where \( 0 < \epsilon < 1 \)

  - \( [P(A) - \epsilon, P(A) + \epsilon] \) if \( P(A) - \epsilon \leq 0 \)
  - \( [0, P(A) + \epsilon] \) if \( P(A) - \epsilon > 0 \)

Special Cases:

- Likelihood evidence: (derived from prior)
  - True – normal \( [1 - \frac{1}{2}, \frac{1}{2}] \) & \( [1 - \frac{1}{2}, \frac{1}{2}] \)
  - False – anomaly \( [\frac{1}{2}, 1] \) & \( [\frac{1}{2}, 1] \)

- Without losing generality, \(|P_1 - P| > |P_2 - P|\)
- If \(|\text{threshold}| < |P_2 - P|\), the # of false alarm raised
- If \(|\text{threshold}| > |P_1 - P|\), the # of detection missed raised

Hence, we need "fine tuning" threshold

Simulation Results

- Performance metric
  - False alarm rate – measured over normal itineraries. Suppose \( N_\text{n} \) normal itineraries are measured, and \( N_\text{a} \) of them are identified as abnormal, false alarm rate is defined as \( N_\text{a}/N_\text{n} \)
  - Detection rate – measured over abnormal itineraries. Suppose \( N_\text{a} \) abnormal itineraries are measured, and \( N_\text{d} \) of them are detected, detection rate is defined as \( N_\text{d}/N_\text{a} \)
By fine tuning the design parameter, threshold, we can achieve good performance for all states.

Threshold can be determined efficiently, e.g.,

- \([0.5, 0.4, 0.9, 0.35]\) (4 states of 10000 records)
- \([0.9, 0.9]\) (2 states of 10000 records)

Number of states affect the accuracy.

- True negative case
  - \([1 \ 4 \ 1 \ 2 \ 3 \ 3]\) (4 states) --- detected
  - \([1 \ 2 \ 1 \ 2 \ 2]\) (2 states) --- missing detected

- True positive case
  - \([1 \ 4 \ 1 \ 2 \ 3 \ 3]\) (4 states) --- false alarm
  - \([1 \ 2 \ 1 \ 2 \ 2]\) (2 states) --- no false alarm

The more the states, the better the accuracy.

Fine tuning threshold

- \([0.33, 0.1, 0.9, 0.1]\) – false alarm rate of state 1, 2 and 4 is too high
- \([0.8, 0.9, 0.9, 0.9]\) – detection rate of state 2 and 4 is too low

Thus, \([0.5, 0.4, 0.9, 0.35]\) is chosen to be the reasonable threshold value.
Simulation Results (cont’d)

Simulation results of identifying neighbor nodes for determining threshold

General Case

In summary, given a specific BN topology, identifying crucial nodes and/or states are critical for improving the efficiency of BANBAD algorithm.

Infer from True and False evidences

- Use majority rule to infer True (i.e., normal) or False (i.e., abnormal) from all the evidences of all the neighbor features
- Easily extendable to distributed scenario
Contributions

- Easily generate and maintain the profile by applying the structure learning algorithm
- Detect anomaly for any feature given any evidence even if some values are missing
- No need to rely on just one or two features for anomaly detection
- Achieve fairly good performance for all the states (value-ranges)
- Reduce the overhead of the BANBAD algorithm by identifying neighbor nodes
- Design parameter – threshold, can be efficiently determined

Future Work

- Distributed BANBAD
  - Intrinsic ability of BN is distributed
  - Majority agreement (minority node can be treated as faulty)
  - Profile can be stored in many locations (clusters) to make it more secure
  - Automatically recognize and compensate for faulty nodes (draft idea, more assumptions needed)
- Multimodal techniques
  - To represent anomalies using multimedia methods (in audio, video, or image formats)

Partial References

Thank you
Questions?
http://www.cs.wmich.edu/wise