Mining Lines in the Sand: On Trajectory Discovery From Untrustworthy Data in Cyber-Physical System

Lu-An Tang[1], Xiao Yu[1], Quanquan Gu[1], Jiawei Han[1]

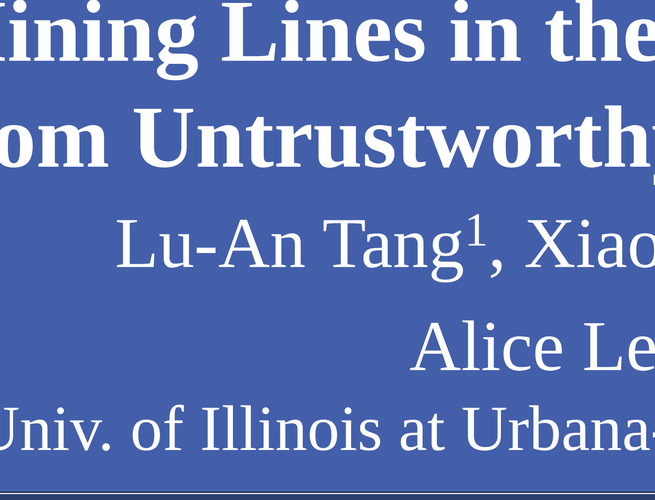
Alice Leung[2], Thomas La Porta[3]

[1] Univ. of Illinois at Urbana-Champaign; [2] BBN Tech.; [3] Penn. State Univ.

Motivation: Mining Lines in the Sand

Cyber Physical System: Integrate physical devices (sensors, cameras) with cyber components to form a situation aware analytical system

1. Deploy larger number of sensors
2. Collect seismic, acoustic and magnetic signals from the field
3. Discover intruder trajectories from the sensor data



Challenges:
- Massive and streaming data
- Untrustworthy data
- Tracking multiple intruders

Problem Statement: Mining Lines in the Sand

Input: sensor network S; sensor data arriving by time, $R = \{R_1, R_2, \ldots, R_m\}$.

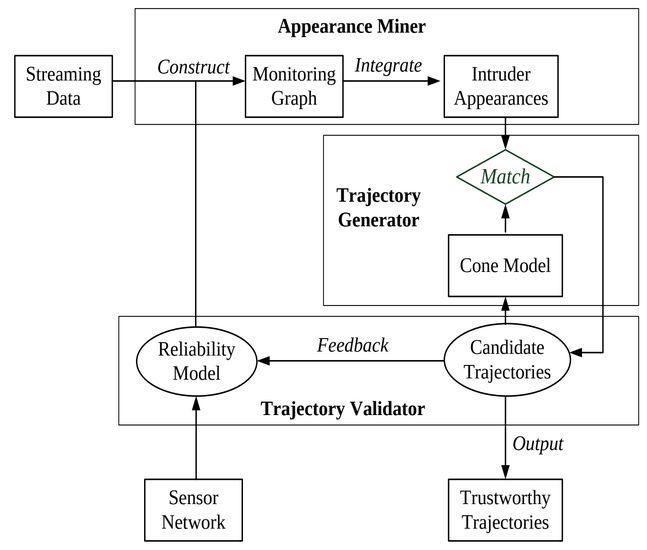
Output: a set of intruder trajectories $\{L_1, L_2, \ldots L_k\}$

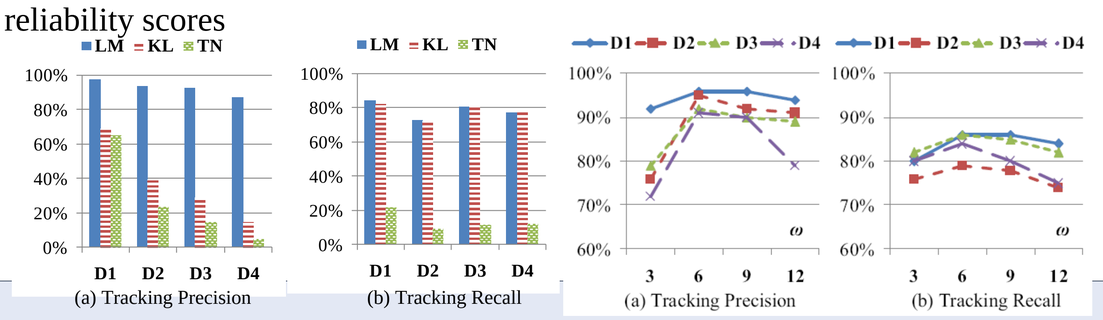
Two sub-problems: (1). Mining dots in the sand; (2). Connecting dots as lines

Overview

Technical Contribution:

1. Propose a watching network to model the relationship among the sensors, records and intruders
2. Detect the intruder appearances based on the link information of watching network
3. Design a cone model to track multiple intruders
4. Validate the mining results and feed back to update sensors' reliability scores





Detecting Intruder Appearances in Watching Network

Mining dots in the sand:

- Input: sensor network S; sensor data in time t_j, $R_j = \{r_{1,j}, r_{2,j}, \ldots, r_{n,j}\}$, sensor's reliability scores
- Output: intruder appearances in time t_j

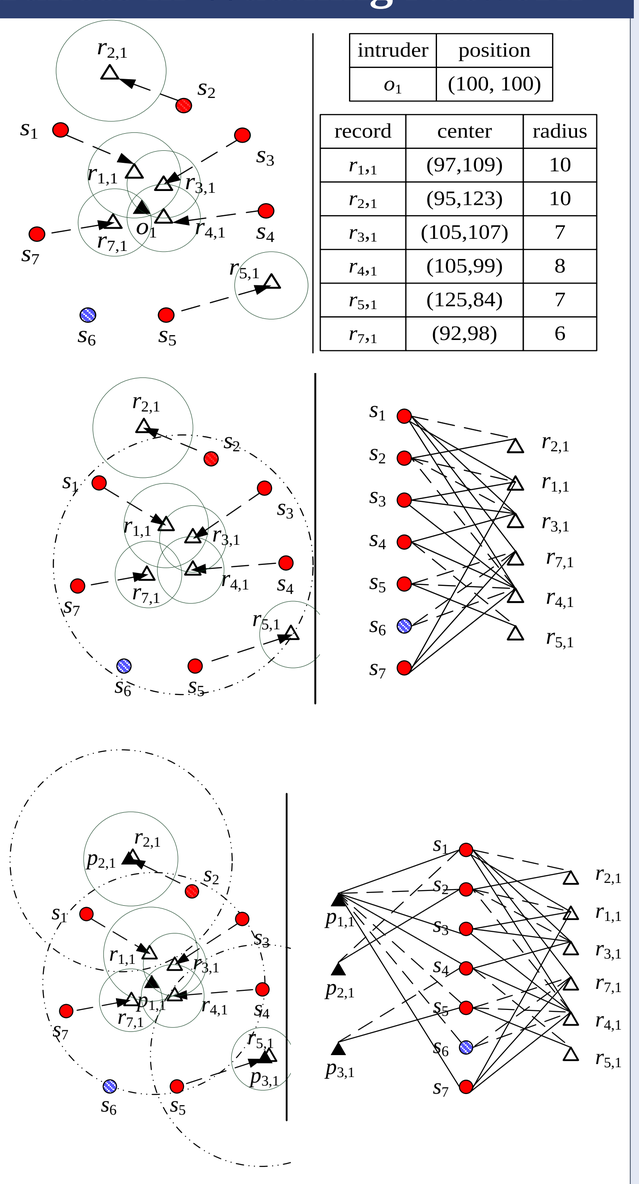
1. Constructing the watching network:
- Select the watching sensors for each record
- Responding sensor: also has a detection near-by
- Non-responding sensor: no detection or the detection is far away

2. Detecting the intruder appearances
- Cluster the detection records
- Calculate the position of the intruder appearances for each cluster

The position is a weighted average of the detection centers, the weights are determined by the detection radius

$$p_{k,j} = \sum_{r_{i,j} \in H_{k,j}} \lambda_{i,j} \cdot cen(r_{i,j})$$

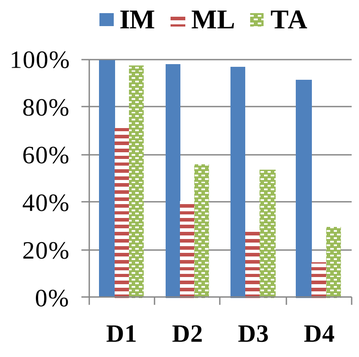
$$\lambda_{i,j} = 1 - \frac{rad(r_{i,j})}{\sum_{r_{l,j} \in H_{k,j}} rad(r_{l,j})}$$



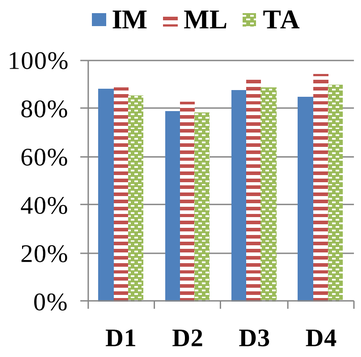
intruder	position
o_1	(100, 100)

record	center	radius
$r_{1,1}$	(97,109)	10
$r_{2,1}$	(95,123)	10
$r_{3,1}$	(105,107)	7
$r_{4,1}$	(105,99)	8
$r_{5,1}$	(125,84)	7
$r_{7,1}$	(92,98)	6

Compare Intruder Mining (IM) to Maximum Likelihood detecting (ML) and TruAlarm (TA)



(a) Detecting Precision



(b) Detecting Recall

ground truth: real traj. pts
result: detection result
valid detection: $dist(p_g, p_d) < \delta_d$
precision = |valid |/|result|
recall = |valid |/|ground|

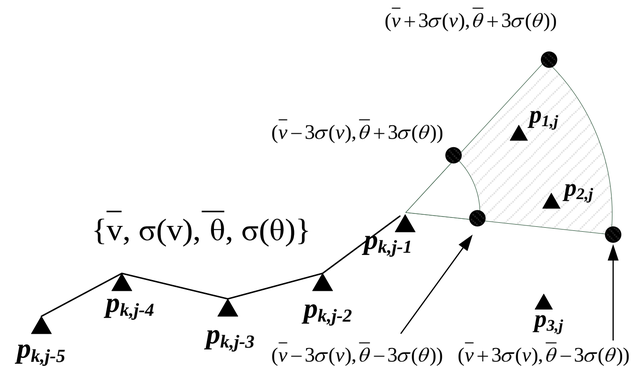
Tracking Intruders with Cone Model

The Candidate-generation-and-refinement framework

- Maintain a set of trajectory candidates
- Match the new intruder appearances with existing candidates, the best matched ones are added to the candidates
- Initialize new candidates from remaining appearances
- Validate the trajectory candidates and update sensor reliability scores

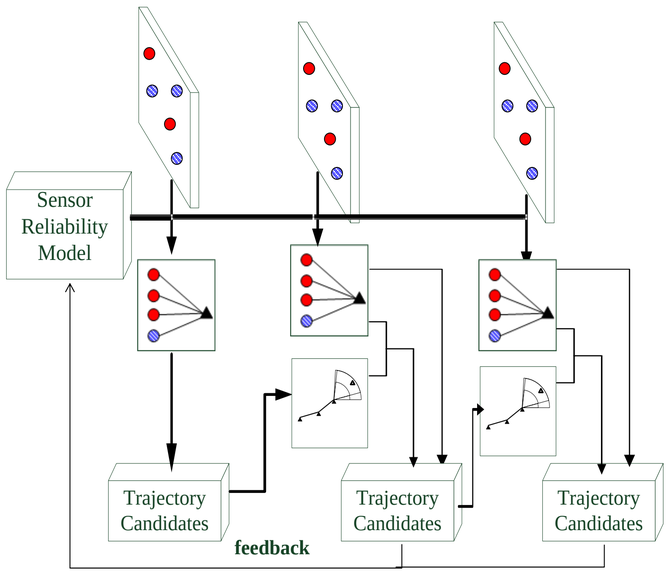
The Cone Model

- Retrieve the ω-most recent points from the candidate trajectory
- Calculate the mean and deviation values of the intruder's moving speed and direction
- Assumption: the moving speed and direction are in the normal distribution
- Construct a cone area to predict next move of L_k



Validation and Feedback

- Observation: The real intruder moves in a continuous manner and the trajectory grows longer over time, the false positive appearances are unlikely to be connected
- Filtering out the false positives by time
- Update sensor reliability scores when a candidate is removed as false positive or selected as the mining results



Acknowledgement

RESEARCH POSTER PRESENTATION 2012
Yahoo-DAIS! Award Competition