Tru-Alarm: Trustworthiness Analysis of Sensor Network in Cyber Physical Systems

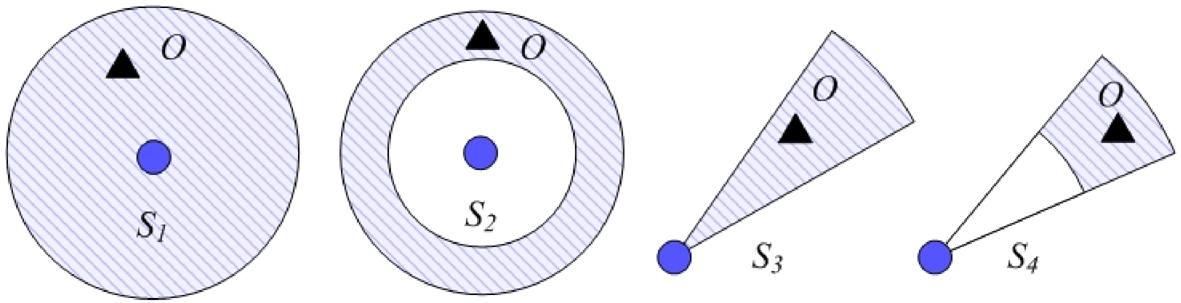
The Database and Info. Systems Lab.
University of Illinois at Urbana-Champaign

Introduction

- A cyber-physical system (CPS) integrates physical devices with cyber components to form a integrated analytical system
- CPS = sensor network + data mining module
  - Traffic monitoring system
  - healthcare system
  - battlefield surveillance, etc
- Major Problem: Data reliability, especially the trustworthiness due to technology limitation and environment influences

CPS Sensors for Motion Detection

- The CPSs are deployed in different scenarios with various types of sensors
- In the scenario of motion detection, several types of sensors are used



(a) Common Sensor

(b) Range Sensor

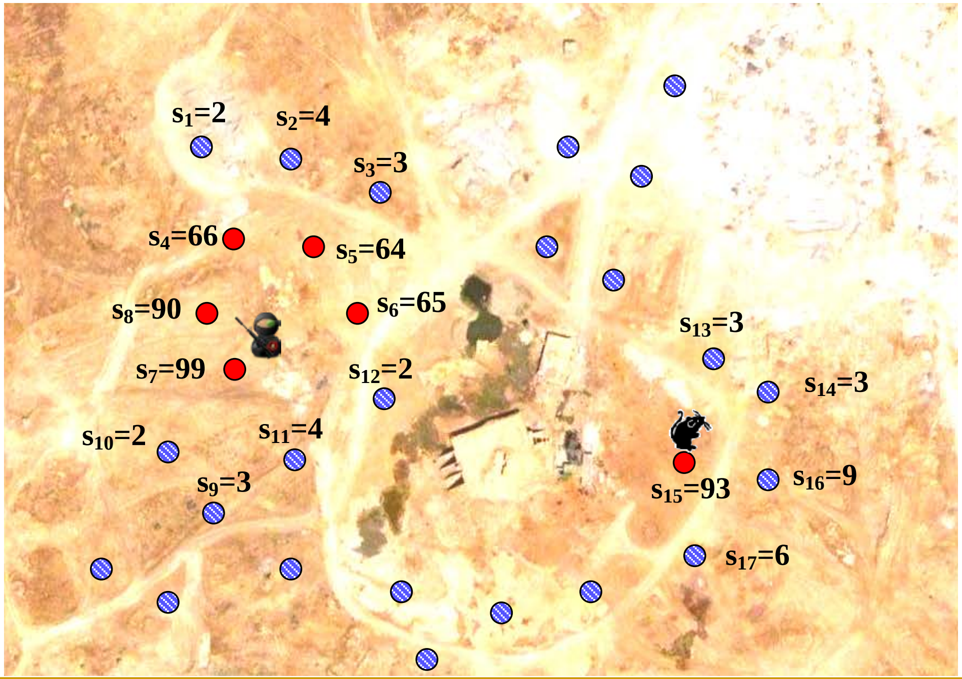
(c) Bearing Sensor

(d) Combination Sensor

- Common sensors used in this paper, however, the method also works for other types of sensors

Motivation Example: Motion Detector

- Battle Network: Deploy sensor network to detect hostile object and actions
- Problem: Sensors are easily damage or influenced by irrelevant activities – generate false alarms



Problem Definition

- Given a CPS dataset including both alarming and normal data records, find out the trustworthy alarms – Focuses on the trustworthiness tasks for alarming records

- Formal Definition:
- *Let* $R = \{r(s_1, t_1), r(s_1, t_2), \ldots r(s_m, t_n)\}$ *be a CPS dataset,* $R_a \subseteq R$ *be the set of alarm records, given a trustworthy threshold* δ_t*, the Tru-Alarm's task is to find out the trustworthy alarms* $r_a(s, t)$ *with* $\tau(r_a) > \delta_t$

Challenges in Trustworthiness Analysis

- Huge size: A typical CPS contains hundreds of sensors and millions of data records
- Unreliable Data: Buonadonna *et.al*: 51% of the data are faulty; Szewzyk *et.al*: 60% of the data are faulty in a deployment in green lake
- No/Rare Training Sets: it is costly and error-prone to manually label the large dataset
- Conflicts of Sensors: Well deployed sensor network has reasonable redundancies.

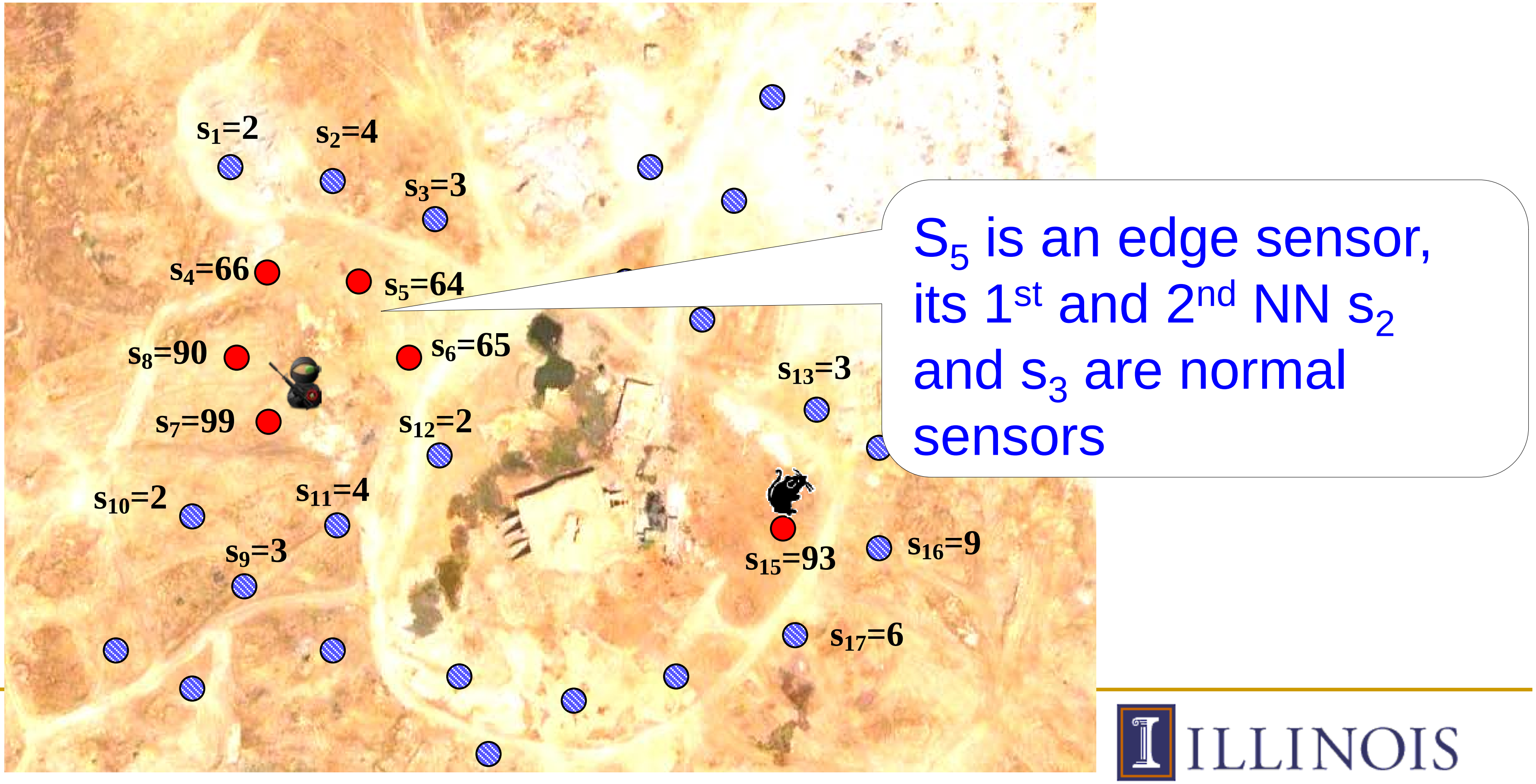
Related Works: Spatial Similarity

- Assumption: The sensors that are spatially close to each other should report the similar readings (Krishnamachari et. al 2004)

- kNN Approach
  - Setup a neighbor threshold *k*
  - Judge the alarm trustworthiness by neighboring information
  - Suppose an alarm sensor *s* has *l* alarming neighbors in its kNN, if $l/k > \delta_t$, the alarm is trustworthiness, else it is not

Problem of Spatial Similarity based Approach

- The edge sensor's alarms may be ignored
- Hard to determine *k*



Related Works: Temporal Similarity

- Assumption: The sensors that reports alarms in the same time are likely to report together in the future (Xiao et. al 2007)
- Train a correlation model from historical data, test the alarms by such model
- Problem:
  - The noisy data are in a large portion (30% -- 50%)
  - The damaged sensors are likely to report false alarms for a long time
  - Some unreliable sensors and false alarms may have such strong correlations with real alarms

TruAlarm : Philosphy

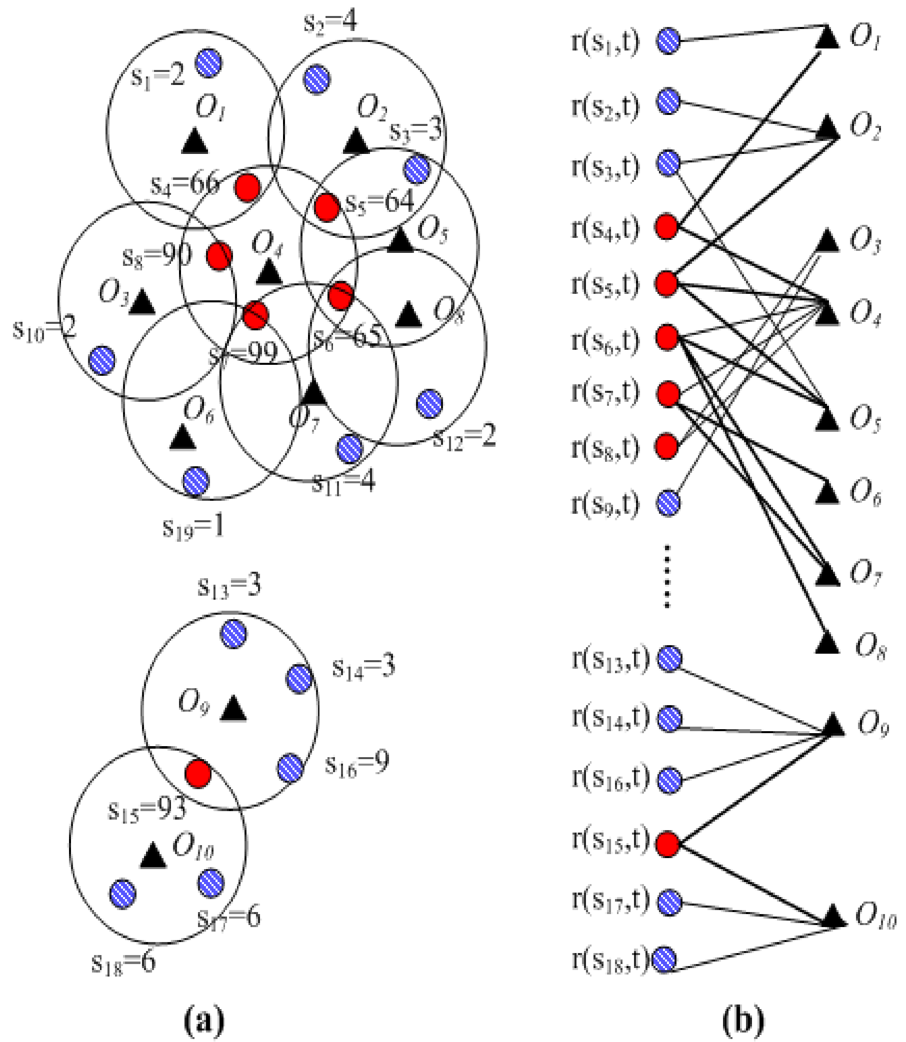
- More trustworthy the alarms are, more accurately we can estimate the object locations
- More accurate the object positions are, more trustworthy the alarms are

- Observation: Mutual Enhancement

  - Estimate object locations from noisy data.
  - Use such objects to verify the alarms – find out the false ones and trustworthy ones
  - Refine the object locations with trustable alarms

Build up links of objects and alarms

- Construct a bipartite graph of object (positions) and sensor (records)
- For each sensor s: the monitored objects O_s
- For each object o: the monitoring sensors S_o



ILLINOIS

Task 1: Compute object trustworthiness

- For each object o: the monitoring sensors S_o

- Conditional trustworthiness: $\tau(r_a(s_i,t)|o)$
  - How likely the alarm $r_a(s_i,t)$ is caused by an object o

- o's trustworthiness $\tau(o)$ is the average of all its conditional alarm trustworthiness of alarms in

$$\tau(o) = \frac{\sum_{r_a \in R_a} \tau(r_a(s,t)|o)}{|S_o|}$$

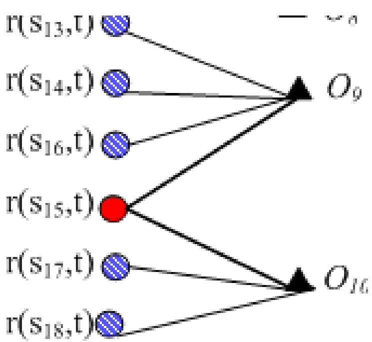
- So we need to compute $\tau(r_a(s_i,t)|o)$?

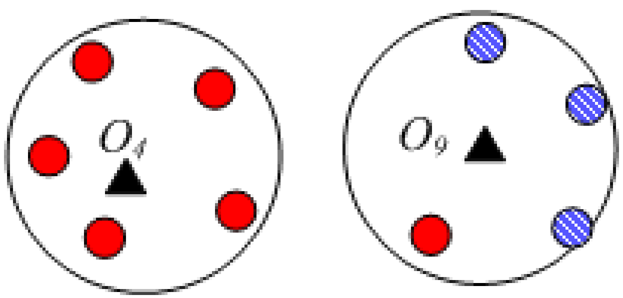
ILLINOIS

Estimate $\tau(r_a(s_i,t)\,|\,o)$:

- It is determined by the coherence of other sensors' readings in the same monitoring sensor set of S_o

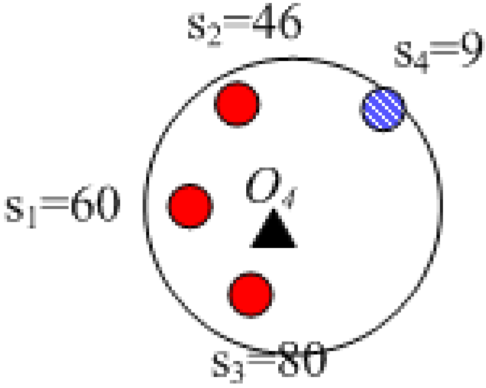
$$\tau(r_a(s_i,t)|o) = \frac{\sum\limits_{s_j \in S_o, s_j \neq s_i} coh(r(s_j,t), r_a(s_i,t))}{|S_o| - 1}$$





Estimate coherence of two sensor records

- $coh(r_a(s_i, t), r(s_j, t))$?
- The system should take count in both their reading differences and positions
- $r_i = f(dist(s_i, o), \Omega(o))$,
- Estimate $\Omega_i(o)$ by r_i : $\Omega_i(o) = f^{-1}(dist(s_j, o), r_j)$
- $r_j' = f(dist(s_j, o), \Omega_i(o))$, -- the expect value of r_j from r_i



Estimate coherence of two sensor records

- Coherence $coh(r_a(s_i, t), r(s_j, t))$ is judged by the difference of the expected reading and real value

$$diff(r', r) = |r'(s_j, t) - r(s_j, t)|$$

$$coh(r_a(s_i, t), r(s_j, t)) = \begin{cases} 1 - \frac{diff(r', r)}{\sigma} & \text{if } diff(r', r) < \sigma \\ 0 & \text{otherwise} \end{cases}$$

  - σ is the standard deviation of monitoring sensor set S_o
  - If s_i' reading is the same as expected value, the coherence score reaches the maximum of 1; if the difference is larger than σ, the score is set to 0

ILLINOIS

Compute object trustworthiness

- A low $\tau(r_a|o)$ indicates two possibilities:
  - r_a is a false alarm
  - r_a is a true alarm, but it is not caused by object o
- In either case, object o is not likely to be a real one; a real object should cause alarms for all its monitoring sensors
- o's trustworthiness $\tau(o)$ is the average of all its conditional alarm trustworthiness

$$\tau(o) = \frac{\sum_{r_a \in R_a} \tau(r_a(s,t)|o)}{|S_o|}$$

Task II: Compute alarm trustworthiness

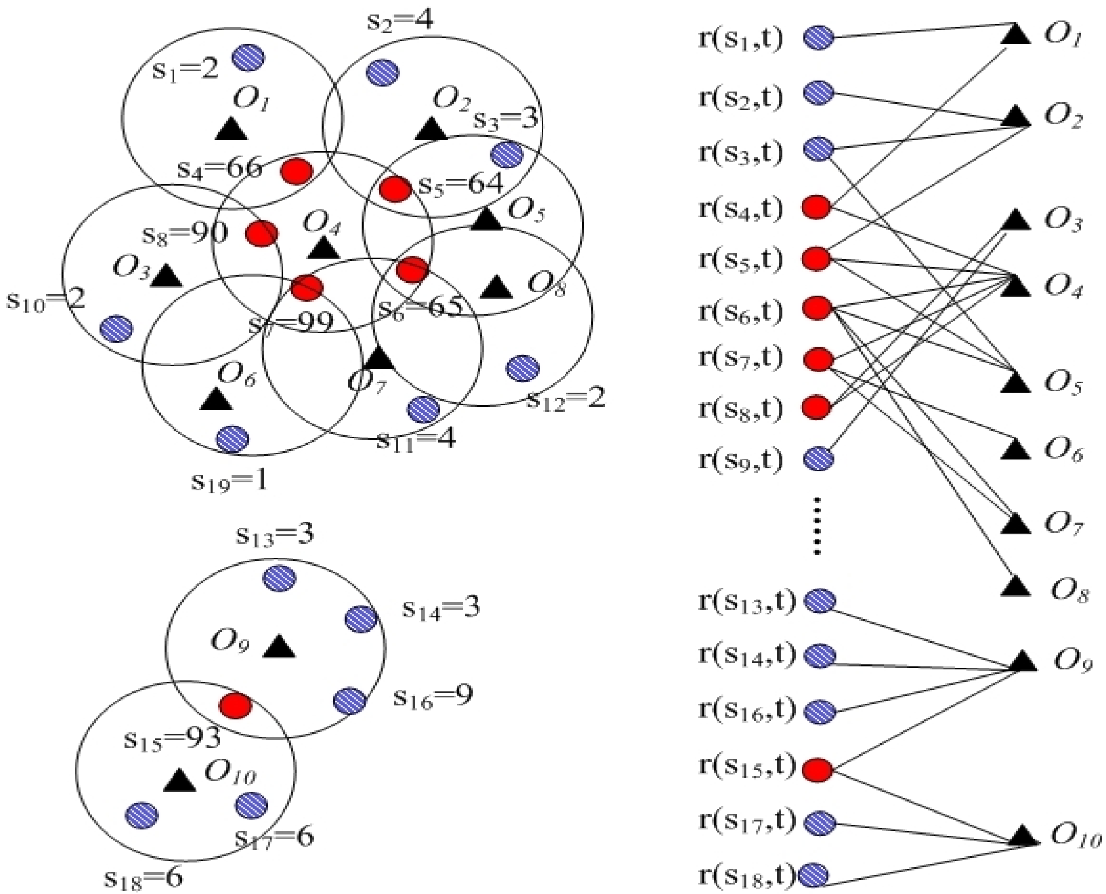
- Even there is only one real object that causes the alarm, such alarm is still meaningful
- If an alarm has different conditional trustworthiness with different objects, we will take the maximum one as $\tau(r_a)$

$$\tau(r_a(s,t)) = \max(\tau(r_a(s,t)|o)), o \in O_s$$

Tru-Alarm Algorithm

- For each object *o,* first retrieves its related data records from the object-alarm graph, and computes the conditional alarm trustworthiness
- The object's trustworthiness is then computed as the average of its conditional alarm trustworthiness
- The system groups the conditional alarm trustworthiness by alarm and select the max one as $\tau(r_a)$

Running Example I



(a) (b)

ILLINOIS

Running Example II

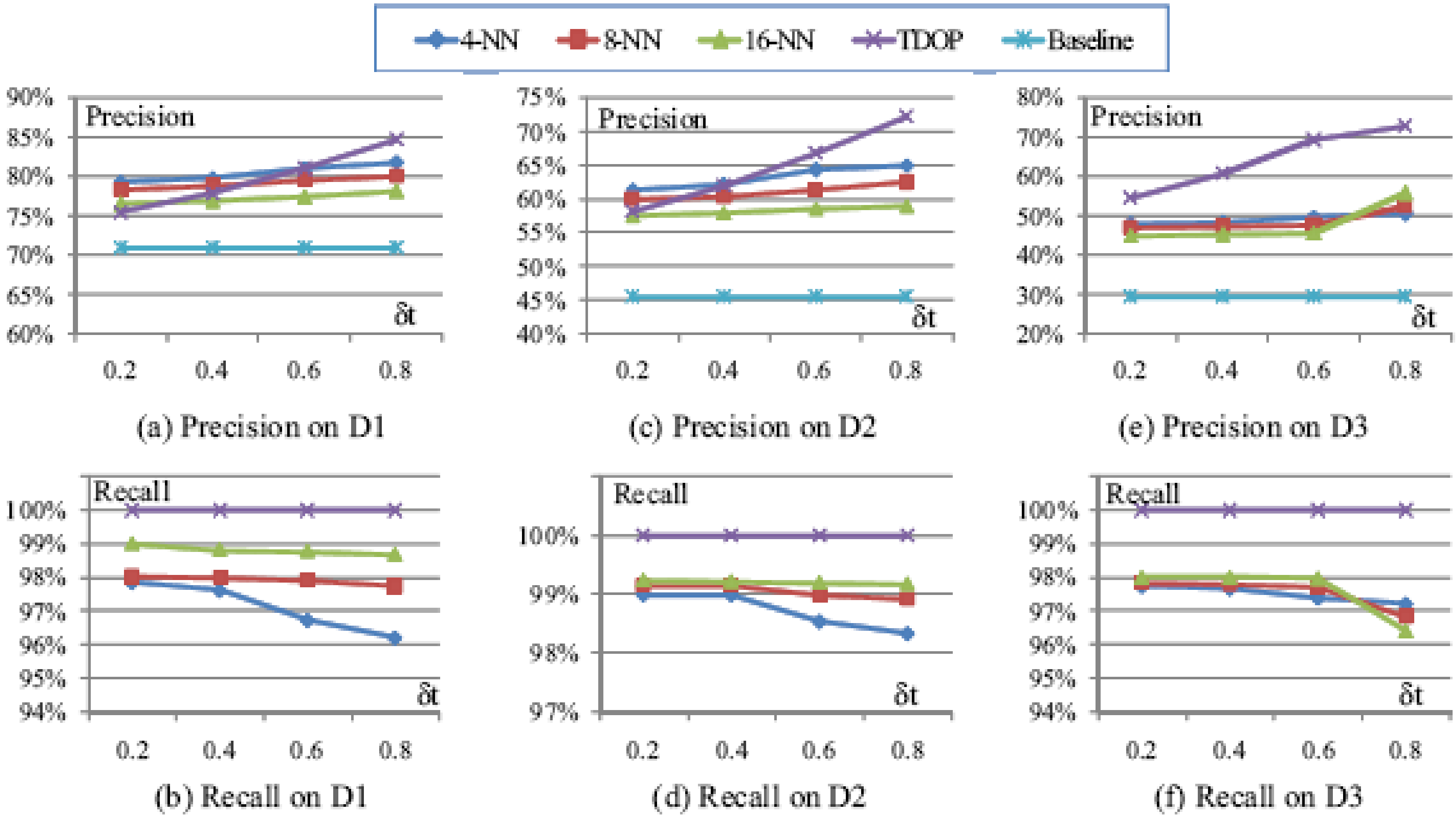
Conditional Alarm Trustworthiness (Group by Object)	Object Trustworthiness	Conditional Alarm Trustworthiness (Group by Sensor)	Alarm Trustworthiness
$\tau(r(s_4,t)\|o_1)=0.3$	$\tau(o_1)=0.15$	$\tau(r(s_4,t)\|o_1)=0.3$ $\boldsymbol{\tau(r(s_4,t)\|o_4)=0.92}$	$\tau(r(s_4,t))=0.92$
$\tau(r(s_5,t)\|o_2)=0.27$	$\tau(o_2)=0.09$	$\tau(r(s_5,t)\|o_2)=0.27$ $\boldsymbol{\tau(r(s_5,t)\|o_4)=0.89}$ $\tau(r(s_5,t)\|o_5)=0.43$	$\tau(r(s_5,t))=0.89$
$\tau(r(s_8,t)\|o_3)=0.10$	$\tau(o_3)=0.05$	$\boldsymbol{\tau(r(s_7,t)\|o_4)=0.91}$ $\tau(r(s_7,t)\|o_5)=0.66$ $\tau(r(s_7,t)\|o_7)=0.59$ $\tau(r(s_7,t)\|o_8)=0.44$	$\tau(r(s_7,t))=0.91$
$\tau(r(s_4,t)\|o_4)=0.92$ $\tau(r(s_5,t)\|o_4)=0.89$ $\tau(r(s_7,t)\|o_4)=0.91$ $\tau(r(s_8,t)\|o_4)=0.82$ $\tau(r(s_9,t)\|o_4)=0.76$	$\tau(o_4)=0.86$	$\boldsymbol{\tau(r(s_8,t)\|o_4)=0.82}$ $\tau(r(s_8,t)\|o_3)=0.10$	$\tau(r(s_8,t))=0.82$
……	……	……	……
$\tau(r(s_{15},t)\|o_9)=0.03$	$\tau(o_9)=0.01$	$\tau(r(s_{15},t)\|o_9)=0.03$ $\boldsymbol{\tau(r(s_{15},t)\|o_{10})=0.04}$	$\tau(r(s_{15},t))=0.04$
$\tau(r(s_{15},t)\|o_{10})=0.04$	$\tau(o_{10})=0.02$		

Experiment Setup

- Synthetic a battle field with hundreds of sensors
- Objects (i.e., tanks and soliders) move across the battlefield
- Random false alarms added in

Dataset	Sensor#	Alarm#	True Alarm Rate
D1	625	5247	71%
D2	900	12390	46%
D3	2500	39415	29%
Parameter Settings			
Dataset: default D3 Sampling Ratio l%: default 4% k in kNN: 4 to 16			

Precision and Recall with kNN methods



ILLINOIS

Thank You Very Much!

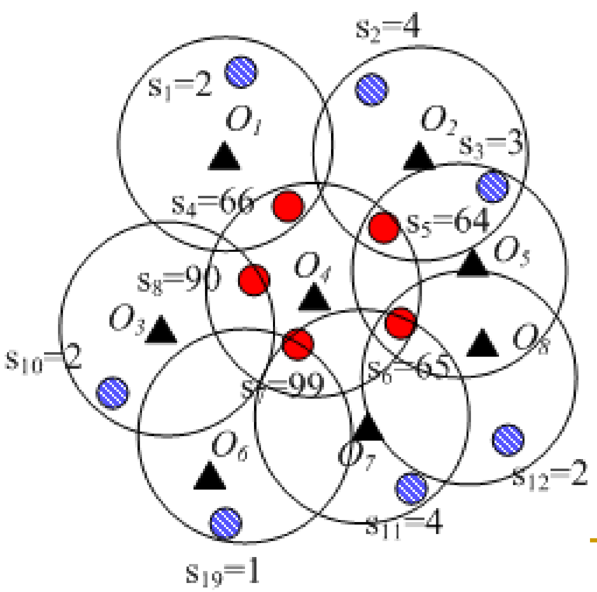
Efficient Trustworthiness Analysis

- The time complexity of *Tru-Alarm* is linear in the number of objects
- The efficiency will be a problem when there are a large number of objects generated by the sampling algorithm
- Most objects turn out to be low trustworthy: In the running example, there are 10 objects but only one is trustworthy
- Can we prune the untrustworthy objects in advance?

Upperbound of $\tau(o)$

- Let o be an object, S_o be its monitoring set and Ra_o be the set of related alarms. $\tau(o)$'s upper-bound $\tau(o) = |Ra_o|/|S_o|$

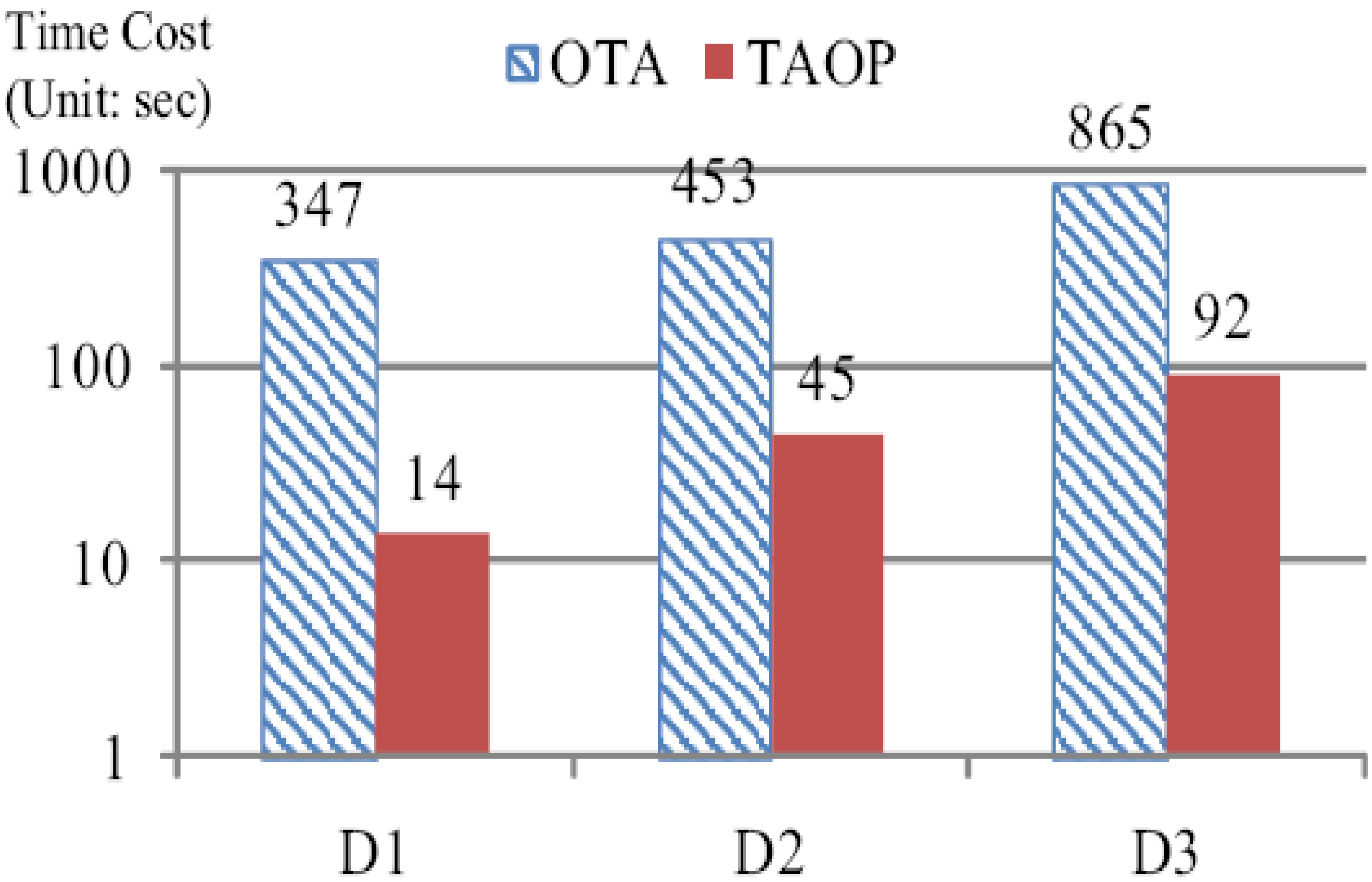
$$\tau(o) = \frac{\sum_{r_a \in R_a} \tau(r_a(s,t)|o)}{|S_o|} < \frac{|Ra_o|}{|S_o|}$$



Improved Tru-alarm Algorithm

- Initialize the trustworthiness for each object and alarm
- For each object *o*, first compute its upper-bound, if it is less than δ_t, then prune it
- Retrieves *o*'s related data records from the object-alarm graph, and computes the conditional alarm trustworthiness
- The object's trustworthiness is then computed as the average of its conditional alarm trustworthiness
- Groups the conditional alarm trustworthiness by alarm and select the max one as $\tau(r_a)$

Time Cost



Time Cost
(Unit: sec)
OTA
TAOP
1000
100
10
1
347
14
453
45
865
92
D1
D2
D3