Identification of Intruder Paths in Sensor Networks

James A Shine[1], James E Gentle[2]
[1]US Army ERDC Topographic Engineering Center, 7701 Telegraph Road, Alexandria, VA 22315
[2]SGeorge Mason University, 4401 University Drive, Fairfax, VA 22030

Abstract
Sensor networks are designed to discover as much information about anomalous activities (such as intrusion) as possible. Some important knowledge for such networks are normal sensor activity levels, sensor alarms (when sensors are activated more than normal), and patterns over multiple sensors. Our work focuses on methods for discovering possible paths that intruders might be taking based on a record of sensor activations. We use the distance and time between successive sensor activations. At any activation, the primary question is whether the activity is caused by a new intruder, or by one who has already activated an earlier sensor (or sensors). Our approach looks in both temporal directions to examine and weight all possible intruder paths. We are developing a tool that outputs possible paths given input from sensors of time and location. Initial results from the tool are presented.

Key Words: network analysis, sensors, intrusion detection, probabilistic network

1. Introduction

1.1 Problem Statement

The authors have done previous work with sensor data analysis to model normal activity and discover anomalous patterns (Shine and Gentle, 2010, 2011). The work described in this paper differs from that work in that we are looking at sensor data events that are highly unusual and indicative of an unexpected intruder, as opposed to searching for anomalous periods of peak activity in sensors where individual activations are not considered to be an anomaly.

Our work examines a contiguous geographic region with a number of fixed sensors, and we assume data coming in a sequence of sensor activations and the time that they are activated. The spatial location of each sensor is obviously implicit information for the problem as well.

Based on the sequential time and location, we look to build and iteratively update a probabilistic network that describes the status of possible intruders as events occur or are registered. We start with the assumption that each new event could be either a new intruder or (for all events but the first) an existing intruder that has moved from their initial intrusion location to the location of the new intrusion event. At any given location or time of a new event, we assign probabilities to each possible intruder. At any given time, we also assign probabilities for each location for each possible intruder.

Figure 1 shows a sensor layout on a geographic grid; the data are synthetic.

.

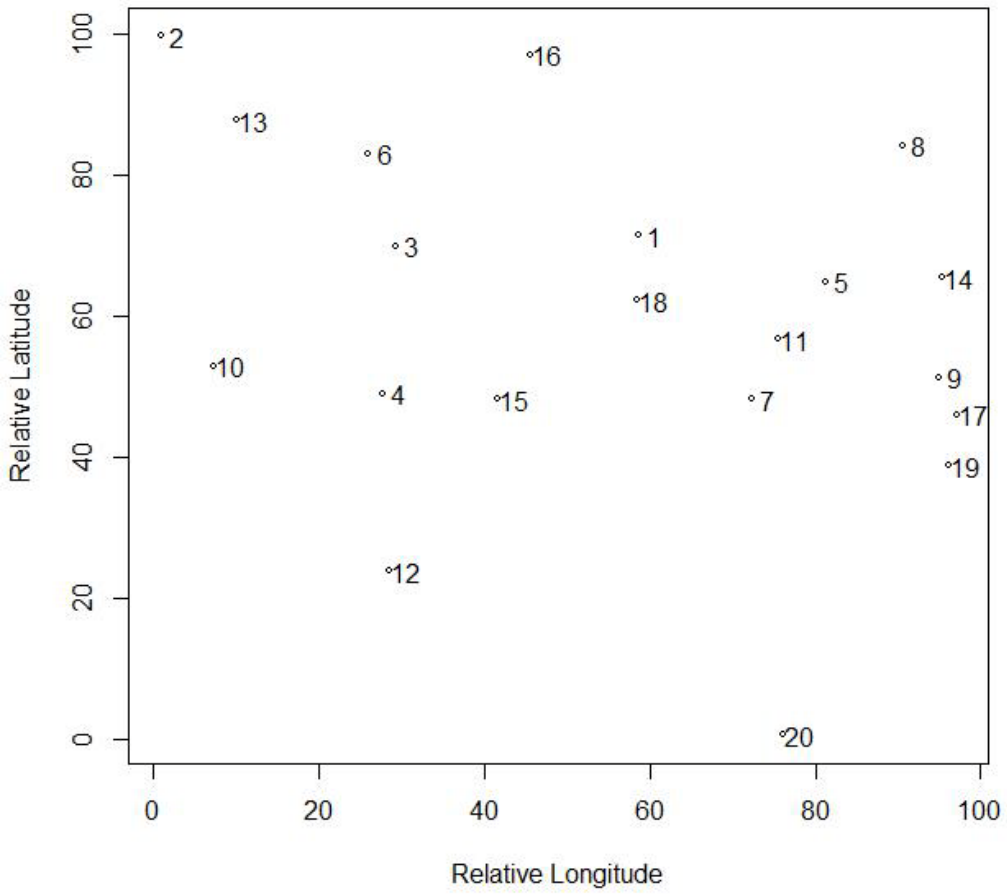

Figure 1: A sensor network.

Figure 2 illustrates visually the problem we are seeking to analyze. The sensor numbers and locations are identical to those in Figure 1. We have drawn a convex hull around the sensors; the location of a sensor with respect to the hull (and the geographic perimeter) will affect the probabilities of old and new intruders (see Section 1.2 for details). Activations occur in sequence at sensors 4, 13 and 1. Each activation could be caused by a new intruder entering the grid from the outside (represented by the blue arrows) or (in the cases of the events at sensors 13 and 1) could be caused by previous intruders moving from one sensor to another in the time interval between the activation events.

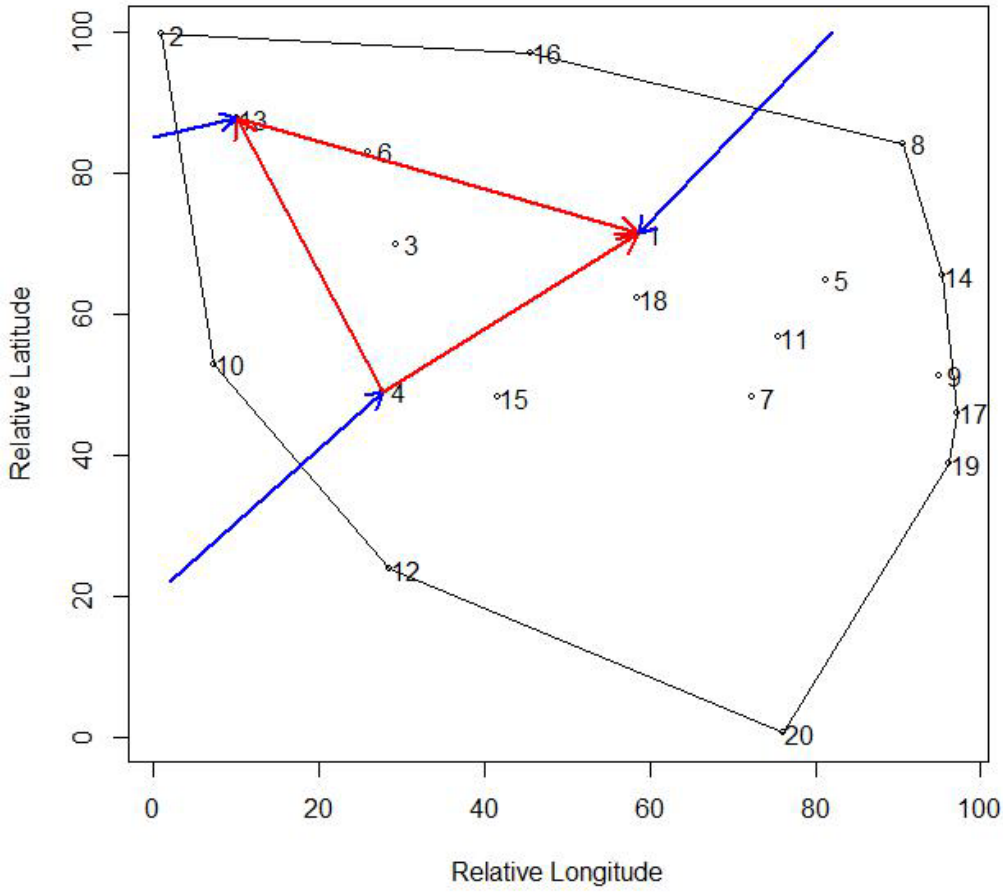

Figure 2: The the outcome of 3 sequential events (at sensors 4, 13 and 1) and all the intruder movement possibilities.

1.2 Sensor location

The location of a particular sensor will affect the probabilities of old and new intruders on that sensor. Each alarm is given a weight based on its location with respect to the perimeter of the grid. It would make sense that new intruders are more probable for sensors near the edge of the grid, and less probable for sensors in the interior of the grid. Figure 3 shows the importance of the sensor location on the overall grid in terms of the probability of a new intruder. We use a convex hull algorithm to compute the sensors that are closest to the geographic perimeter; the hull is shown in Figure 3 as well. All sensors on the hull are given an equal weight for new intruder probability; interior points get a lesser weight based on a function of their distance from the hull. This is discussed further in Section 1.4.

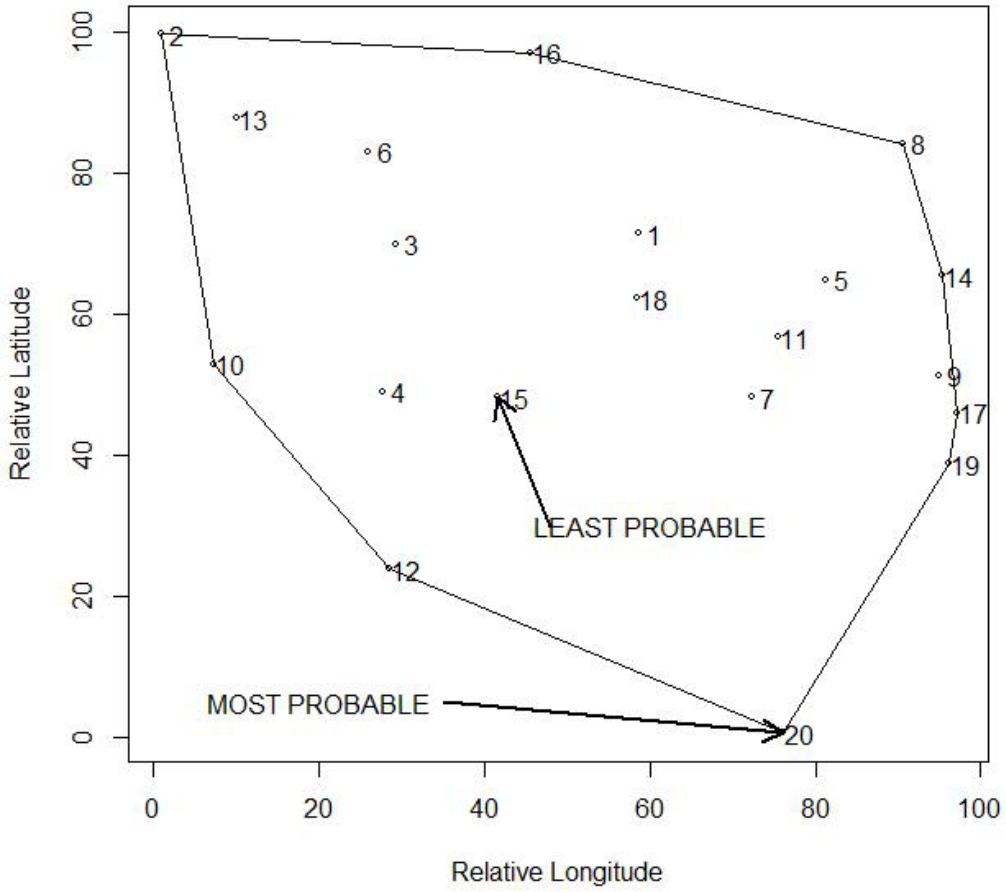

Figure 3: Visualization of the convex hull and the relative probabilities of inner and outer sensors for being activated by a new intruder.

1.3 Building a probabilistic network

At the first activation, there is definitely a new intruder, and the probability of that new intruder at the first activation is 1. This is the beginning of the evolving list of intruders and probabilistic network. In Figure 2, the blue arrow coming into sensor 4 indicates that an intruder has entered the perimeter from the outside and has triggered an alarm at that sensor. The probability of course is 1, and we store that value as P(1,1), the probability that the first intruder is at the first alarm site.

At the second activation (sensor 13), there may be a new intruder (shown by the blue arrow coming into sensor 13 in Figure 2) or the intruder at sensor 4 may move from sensor 4 to sensor 13 (the red arrow connecting these two sensors). The probability of a new intruder will be P(2,2); the probability that the initial intruder has moved to sensor

13 is P(1,2); and the probability that the initial intruder is still at sensor 1 is P(1,1), which is now less than 1.
At the third activation (sensor 1), there may be a new intruder (blue arrow in Figure 2), or a new intruder from sensor 13 that has moved to sensor 1, or the initial intruder moving from sensor 13 to sensor 1, or the initial intruder moving from sensor 4 to sensor 1. The first two probabilities will be P(3,3) and P(2,3) respectively; the sum of the last two probabilities will be P(1,3). The probabilities P(2,2), P(1,2) and P(1,1) will be adjusted as well to keep the probabilities for any intruder from exceeding 1. Our approach thus adjusts the probabilistic network in the reverse temporal direction as well as the forward one.

For this paper, we are assuming that an intruder is only at one of the activated sensors, which works well for simplification but may not be realistic. An intruder may be between sensors or have left the geographic grid. This is an issue we will consider in future work.

The specifics of probability computation will be discussed further in Section 2. Obviously, complexity grows rapidly with the number of events.

1.4 Setting probabilities: edginess, distance and time between activations

For any sensor activation event k (except the first), our model assigns a probability of a new intruder P(k,k) (vs an existing one, P(1:(k-1),k) based on the sensor location on the geographic grid. We compute these for all sensors, and call it "edginess". Sensors on the convex hull of the area (closest to the perimeter) are assigned equal, relatively high probabilities. Sensors in the interior of the area are assigned lesser probabilities proportional to their distance from the convex hull, with sensors in the center of the grid getting the lowest probabilities for new intruders. Other factors (such as distance to the nearest neighbor sensor or distance to the outer perimeter rather than just being on the convex hull) could also affect the weight or probability assigned for a new intruder, but are beyond the scope of this paper. In cases where three dimensions rather than two must be considered (intruders from the air, under ground or under water), the edginess probability would also require modification to include that factor.

The probabilities given to the existing intruders will depend on the distance between their previous possible sensor location (or locations) and the new sensor, as well as the time elapsed between the sensor events. More specifics on this process are given in Section 2.1.

2. Algorithm and Applications

2.1 Algorithm

We have developed an algorithm assigning probabilities for new and existing intruders based on times, locations, and the geographic constraints of the sensor locations. We have designed the algorithm for iterative update and minimal computational burden.

We create four arrays. The first has the geographic sensor information (x and y coordinates). The second contains the distance between any two sensors. These two arrays are fixed at the beginning of the analysis. The third array contains the time and

sensor number for each event. As we develop the model, this array is usually computed and changed before analysis, but it is designed to eventually respond to sensor information coming in from an actual surveillance system. The fourth array is the intruder probability matrix, or probabilistic network. This is designed to be the output to any agent monitoring the area for intrusion.

First, we initialize the first two arrays (sensor locations and distances), and then compute a complex hull and "edginess" probabilities for new intruders for each sensor. Then we fill in the intruders probability matrix event by event. The values sent to output are the upper portion of a matrix, with each column P(1:k,k) giving the probability of each possible intruder at that event, and each row P(k,1:k) giving the probability that any given intruder is at that particular sensor/event.

We populate the probability matrix for each event k as follows. First, a probability P(k,k) is assigned based on the position of the sensor on the geographic grid, as discussed in Section 1.4. Then we assign values for P(1:(k-1),k) for the probability of each previous possible intruder getting to the sensor at event k from any of its previous possible locations. We use distance between sensors and time between events to determine the magnitude of this probability. Sensors that are close together, and longer times between events, will increase the probability; sensors that are farther apart, and shorter times between events, will decrease it. For earlier intruders, all possible paths from their first intrusion and the current event must be considered. For example, on the diagram in Figure 2, an intruder that comes in at the first event (sensor 4) may get to the third event (sensor 1) directly from sensor 4, or he may get there by traveling from sensor 4 to sensor 13 (the second event) to sensor 4. We sum all possible pathways to compute each total probability, and then normalize for all possible intruders to make the probabilities add to 1.

After the new column is computed, we adjust previous values for each row so that those probabilities also sum to 1. Table 1 shows a 3 event example of this adjustment. For the first event, P(1,1) was 1.0. For the second event, P(2,2) and P(1,2) were .923 and .077 respectively. We then adjust P(1,1) so that P(1,1) + P(1,2) = 1 (maximum; for future work we may allow this probability to be less than 1). Similarly, after the third event in Table 1, the rows for possible intruders 1 and 2 are adjusted in a similar fashion.

Table 1: A sample probabilistic network, as the probabilities change and new possible intruders are added, for the first three sensor events.

	Probabilistic Network Evolving		
Intruder	*Event1*	*Event2*	*Event3*
1	1.00		
1	.923	.077	
2	0	.923	
1	.878	.032	.090
2	0	.744	.256
3	0	0	.654

2.2 Applications

We are designing our algorithm so that it can process data in real or near-real time, updating the probability network each time a new event occurs. Such an algorithm can be used as a surveillance tool of some sort, complementing existing perimeter alert systems. Some applications include security at sensitive installations (military, nuclear, chemical), night surveillance at banks and museums, and containment of people in institutions such as prisons. Unlike much of the work in sensor network analysis, we are tracking intruders rather than just processing alarms.

3. Conclusions and Future Work

We have constructed a demonstration algorithm to predict the probabilities of different intruders at sequential sensor activation events. This approach can be adapted to serve as a surveillance tool that iteratively updates probabilities of different intruders for any given event and also probabilities for the location of any possible intruder at any given time.

For future work, we want to incorporate meta-data or "ground truth" into the probability assessment process. Often there are facts and extra information about the problem which can be used to adjust probabilities and create a more accurate model. We also want to create some sort of visualization, perhaps a map of some sort which changes over time, which will provide better and timelier information to a user than just a matrix of probabilistic network values. Other issues we wish to examine include reactivation of the same sensor at different times and the possibility of multiple intruders at one or more events.

References

Shine, J.A. and J.E. Gentle, "Statistical Surveillance and Alarm Activation in Sensor Networks", JSM 2010, Vancouver, Canada, August 2010.

Shine, J.A. and J.E. Gentle, "Pattern Discovery and Anomaly Detection in Sensor Networks", JSM 2011, Miami Beach, FL, August 2011.