Journal of Urban Health: Bulletin of the New York Academy of Medicine Vol. 80, No. 2, Supplement 1 2003

WSARE: What's Strange About Recent Events?

Weng-Keen Wong, Andrew Moore, Gregory Cooper,
and Michael Wagner

ABSTRACT *This article presents an algorithm for performing early detection of disease outbreaks by searching a database of emergency department cases for anomalous patterns. Traditional techniques for anomaly detection are unsatisfactory for this problem because they identify individual data points that are rare due to particular combinations of features. Thus, these traditional algorithms discover isolated outliers of particularly strange events, such as someone accidentally shooting their ear, that are not indicative of a new outbreak. Instead, we would like to detect groups with specific characteristics that have a recent pattern of illness that is anomalous relative to historical patterns. We propose using an anomaly detection algorithm that would characterize each anomalous pattern with a rule. The significance of each rule would be carefully evaluated using the Fisher exact test and a randomization test. In this study, we compared our algorithm with a standard detection algorithm by measuring the number of false positives and the timeliness of detection. Simulated data, produced by a simulator that creates the effects of an epidemic on a city, were used for evaluation. The results indicate that our algorithm has significantly better detection times for common significance thresholds while having a slightly higher false positive rate.*

KEYWORDS *Anomaly detection, Data mining, Detection algorithm, Multiple hypothesis testing, Syndromic surveillance.*

INTRODUCTION

This article is a shortened version of the Wong et al.[1] article on rule-based anomaly pattern detection for detecting disease outbreaks. Detection systems typically monitor multidimensional temporal data for anomalies and raise an alert on discovery of any deviations from the norm. For example, in the case of an intrusion detection system, an anomaly would indicate a possible breach of security.[2–4] Although early disease outbreak detection appears to be similar to traditional anomaly detection systems, shortcomings in these systems, which we illustrate, limit their usefulness in early disease outbreak detection.

In our database of emergency department (ED) cases from several hospitals in a city, each record contains information about the individual who was admitted to the ED. This information includes age, gender, symptoms exhibited, home location, work location, and time admitted. (To maintain patient confidentiality, personal identifying information such as patient's names, addresses, and identification numbers were not in the data set used in this research.)

Mr. Wong and Dr. Moore are with the Department of Computer Science, Carnegie Mellon University; Drs. Cooper and Wagner are with the Center for Biomedical Informatics, University of Pittsburgh.

Correspondence: Weng-Keen Wong, MSc, Department of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213. (E-mail: wkw@cs.cmu.edu)

Clearly, when an epidemic sweeps through a region, there will be extreme increases in the number of ED visits. While these dramatic upswings are readily apparent during the late stages of an epidemic, the challenge is to detect the outbreak during its early stages and mitigate its effects. Different diseases cause different signals to appear in temporal, spatial, and demographic data.[5] For any anomaly detection algorithm to be successful in early detection of disease outbreaks, it must be able to detect abnormalities in these three aspects of ED data.

A simplistic first approach would be to report an ED case as an anomaly if it has a rare value for some attribute. For example, we would signal an anomaly if we encountered a patient more than 100 years old. While this method detects the outliers for a single attribute, it fails to identify anomalies that occur due to combinations of features that alone might not be abnormal, but together would be unusual. For instance, the first technique would not find anomalies in cases for which the patients were male and under the age of 30 years but exhibited symptoms associated with a disease that affects primarily female senior citizens. Fortunately, there are plenty of anomaly detection algorithms that can identify outliers in multidimensional feature space. Typically, these detection algorithms build a probabilistic model of the "normal" data using a variety of techniques, such as neural nets[6] and a mixture of naive Bayes submodels.[7]

Even that kind of sophisticated outlier detection, however, is insufficient for our purposes. Outlier detection succeeds at finding data points that are rare based on the underlying density, but these data points are treated in isolation from each other. Early epidemic detection, on the other hand, hinges on identifying anomalous groups, which we refer to as *anomalous patterns*. Specifically, we want to know if the recent proportion of a group with specific characteristics is anomalous based on what the proportion is normally. This approach is closely related to the work done by Bay and Pazzani[8] in mining data from contrast sets and is somewhat similar to item set mining.[9] Traditional outlier detection, on the other hand, will likely return isolated irregularities that are insignificant to the early detection system.

We might, then, argue that aggregate daily counts of a single attribute or combination of attributes should be monitored to detect an anomalous group. For instance, we could monitor the daily number of people appearing in the ED with respiratory problems. A naive detector would determine the mean and variance of the monitored signal over a training set that is assumed to capture the normal behavior of the system. Then, a threshold would be established based on these values. Whenever the daily count exceeds this threshold, an alert would be raised.

This technique works well if the monitored features are known. The spatial, temporal, and demographic signatures of diseases are simply too extensive, however, for us to know a priori which features to monitor. We could well miss some combination of features that would indicate an outbreak of a particular disease. Thus, we need an algorithm that is able to detect anomalous patterns rather than predefined anomalies.

The algorithm that we propose is designed to be a general safety net for new kinds of emerging problems that have not been anticipated. Instead of operating by itself, this piece of software is intended to be one component in a suite of detectors. Within the suite, the other detectors will be tuned carefully for spotting specific diseases or classes of diseases.

Our approach to this problem uses a rule-based anomaly pattern detector. Each anomalous pattern is summarized by a rule, which in our current implementation consists of one or two components. Each component takes the form $X_i = V_i^j$, where

X_i is the *i*th feature, and V_i^j is the *j*th value of that feature. Multiple components are joined by a logical AND. For example, a two-component rule would be Gender = Male AND Age Decile = 4. One benefit to a rule-based system is that the rules are easily understood by a nonstatistician.

We need to be wary, however, of the pitfalls of rule-based anomaly pattern detection. Since we are finding anomalous patterns rather than isolated anomalies, we will be performing multiple hypothesis tests. When multiple hypothesis tests are performed, the probability of a false positive becomes inflated unless a correction is made.[10] In addition, as we add more components to a rule, overfitting becomes a serious concern. Thus, a careful evaluation of significance is clearly needed. Furthermore, temporal health care data used for disease outbreak detection are frequently subject to "seasonal" variations. As an example, the number of influenza cases is typically higher during winter than summer. In addition, the number of ED visits varies between weekends and weekdays. The definition of what is normal will change depending on these variations.

RULE-BASED ANOMALY PATTERN DETECTION

The basic question asked by all detection systems is whether anything strange has occurred in recent events. This question requires definitions of what it means to be recent and what it means to be strange. Our algorithm considers all patient records that fall on the current day under evaluation to be recent events. Note that this definition of recent is not restrictive—our approach is fully general, and *recent* can be defined to include all events within some other period.

To define an anomaly, we need to establish what is normal. Our algorithm is intended to be applied to a database of ED cases, and we need to account for environmental factors such as seasonal and weekend versus weekday differences in the number of cases. Consequently, normal behavior is assumed to be captured by the events occurring on the days that are exactly 5, 6, 7, and 8 weeks prior to the day under consideration. The definition of what is normal can be easily modified to another time period without major changes to our algorithm. We refer to the number of events that fit a certain rule for the current day as C_{today}. Similarly, the number of cases matching the same rule from 5 to 8 weeks ago is called C_{other}.

There is clearly a tradeoff when defining the normal period. At one extreme, we could only use data from the previous day. This approach, however, makes the algorithm susceptible to outliers that may only occur in a short, but recent, period. On the other hand, we could determine the baseline from all data collected over the past few years. This choice would smooth out trends in the data, and we might falsely raise an alarm for something that is a seasonal variation. One area of future work for this project would be to determine the baseline distribution automatically given a history of data.

From this point, we refer to our algorithm as WSARE, which is an abbreviation for "what's strange about recent events." WSARE operates on discrete data sets with the aim of finding rules that characterize significant patterns of anomalies. Due to computational issues, the number of components for these rules is two or fewer. Our description of the rule-finding algorithm begins with an overview, followed by a more detailed example.

Overview of WSARE

The best rule for a day is found by considering all possible one- and two-component rules governing events occurring on that day and returning the one with the best "score." The score is determined by comparing the events on the current day with events in the past. The best scoring rule then has its *P* value estimated by a randomization test. This *P* value is the likelihood of finding a rule with as good a score under the hypothesis that the features of the case and the date are independent. The randomization-based *P* value takes into account the effect of the multiple testing that went on during the rule search. If we were running the algorithm on a day-by-day basis, we would end at this step. If we were looking at a history of several days, however, we would need the additional step of using the false discovery rate (FDR) method[10] to determine which *P* values are significant. The days with significant *P* values are returned as anomalies.

One-Component Rules

To illustrate this algorithm, suppose we have a large database of 1 million ED records collected during a 2-year span. This database contains approximately 1,000 records a day, thereby yielding approximately 5,000 records if we consider the cases for today plus those from 5 to 8 weeks ago. We refer to this record subset as DB_i, which corresponds to the recent event data set for day *i*. The algorithm proceeds as follows. For each day *i*, retrieve the records belonging to DB_i. We first consider all possible one-component rules. For every possible feature-value combination, obtain the counts C_{today} and C_{other} from the data set DB_i. As an example, suppose the feature under consideration is the age decile for the ED case. There are nine possible age decile values, ranging from 0 to 8. We start with the rule Age Decile = 3 and count the number of cases for the current day *i* that have Age Decile = 3 and those that have Age Decile ≠ 3. The cases from 5 to 8 weeks ago are subsequently examined to obtain the counts for the cases matching the rule and those not matching the rule. The four values form a 2 × 2 contingency table such as the one shown in the Table.

Scoring Each One-Component Rule

The next step is to evaluate the score of the rule using a test in which the null hypothesis is the independence of the row and column attributes of the 2 × 2 contingency table. In effect, this hypothesis test measures how different the distribution for C_{today} is compared with that of C_{other}. This test will generate a *P* value, which we will call the score to differentiate it from the *P* value obtained from the randomization test. We use the Fisher exact test[11] to find the score for each rule. Running the Fisher exact test on the Table yields a score of 0.00005058, which indicates that the count C_{today} for cases matching the rule Age Decile = 3 is significantly different from the count C_{other}.

Two-Component Rules

At this point, the best one-component rule for a particular day has been found. We refer to the best one-component rule for day *i* as BR_i^1. The algorithm then attempts

TABLE. Sample 2 × 2 contingency table

	C_{today}	C_{other}
Age decile = 3	48	45
Age decile ≠ 3	86	220

to find the best two-component rule for the day by adding on the component to BR_i^1 that yields the best score for the resulting two-component rule, which we refer to as BR_i^2. Note that the best two-component rule may not necessarily be found in this fashion. This greedy approach was taken to reduce the computational cost of the algorithm. In addition, BR_i^2 may not be an improvement over BR_i^1. We need to perform further hypothesis tests to determine if the presence of either component has a significant effect. For further details on the creation of two-component rules, consult Ref. 1.

Finding the *P* Value for a Rule

The algorithm for determining scores shown above is extremely prone to overfitting. Even if data were generated randomly, most single rules would have insignificant *P* values, but the best rule would be significant if we searched more than 1,000 possible rules. To illustrate this point, suppose we follow the standard practice of rejecting the null hypothesis when the *P* value is less than α, where $\alpha = .05$. In the case of a single-hypothesis test, the probability of making a false discovery under the null hypothesis would be α, which equals .05. On the other hand, if we perform 1,000 hypothesis tests, one for each possible rule under consideration, then the probability of making a false discovery could be as bad[12] as $1 - (1 - .05)^{1000} \approx 1$, which is much greater than .05. Thus, if our algorithm returns a significant *P* value, we cannot accept it at face value without adding an adjustment for the multiple hypothesis tests we performed. We address this problem by using a randomization test in which the date and each ED case features are assumed to be independent. In this test, the case features in the data set DB_i remain the same for each record, but the date field is shuffled among records from the current day and records from 5 to 8 weeks ago. The best rule is obtained on this randomized data set. We repeat this procedure 1,000 times and obtain a compensated *P* value for BR_i, which we refer to as CPV_i. The full description for the randomization test is given in Ref. 1.

Using False Discovery Rate to Determine Which *P* Values Are Significant

WSARE can be used on a day-to-day basis similar to an on-line algorithm, or it can be used to review a history of several days to report all significantly anomalous patterns. When using our algorithm on a day-to-day basis, the compensated *P* value CPV_i obtained for the current day through the randomization tests can be interpreted at face value. When analyzing historical data, however, we need to compare the CPV_i values for each day in the history, thereby creating a second overfitting opportunity due to yet another multiple hypothesis testing problem. We address this problem through the use of the FDR method.[10,12] For an in-depth discussion of the use of FDR in WSARE, see the related section in Ref. 1.

RESULTS

We evaluated our algorithm using data from a simulator that simulated (to a first approximation) the effects of an epidemic on a grid world populated by people of varying characteristics. Thus, whenever an infected person exhibited the monitored symptom, an entry would be added to the log file to simulate an ED record.

Our results were obtained by running the simulator for 180 simulated days with the epidemic, named Epidemic0, introduced to the environment on the 90th day. Epidemic0 had a target demographic group of males 50–59 years old. In addition, there were nine nonepidemic background diseases that spontaneously appeared at random points in the simulation. At certain stages, these background diseases caused infected people to display the monitored symptom. These background diseases had low infection probabilities as they were intended to provide a baseline for the number of ED cases. Our initial publication on this subject[1] contains a detailed description of the simulator and the experimental settings.

Evaluation of Performance

We treated our algorithm as if it ran on a day-by-day basis. Thus, for each day in the simulation, WSARE was asked to determine if the events on the current day were anomalous. We evaluated the performance of WSARE against a standard anomaly detection algorithm that treated a day as anomalous when the daily count of ED cases for the monitored symptom exceeded a threshold. The standard detector was applied to the ED case data from day 30 to day 89 in the simulation to obtain the mean μ and variance σ^2. The threshold was calculated by the formula below, in which Φ^{-1} is the inverse to the cumulative distribution function of a standard normal.

$$\text{Threshold} = \mu + \sigma * \Phi^{-1}\left(1 - \frac{P}{2}\right)$$

Both the standard algorithm and WSARE were tested using five levels of P values (.1, .05, .01, .005, and .001). To evaluate the performance of the algorithms, we measured the number of false positives and the number of days until the epidemic was detected. The specific rules for determining detection time and counting false positives are stated in the full version of this article.[1]

Figures 1 and 2 plot the detection time in days versus the number of false positives for five different P-value thresholds used in both the standard algorithm and WSARE. In Fig. 1, the error bars for detection time and false positives are shown. Figure 2 fills in the lines to illustrate the asymptotic behavior of the curves. To extend the graph of the standard model, we performed additional experiments beyond the range of the P values indicated. The values for Figs. 1 and 2 were generated by taking the average of 100 runs of the simulation.

Results From Simulated Data

The results from simulated data indicate that, for P-value thresholds above .01, the detection time for WSARE is significantly smaller than that of the standard algorithm. On the other hand, as the P-value threshold decreases, the detection time for WSARE is somewhat worse than that of the standard algorithm. Choosing an extremely low threshold would be unprofitable, however, since all anomalies except those at an unusually high significance level would be ignored. For example, using a threshold of .01 corresponds to a 99% significance level.

The results also demonstrate that WSARE signals more false positives than the standard algorithm for higher P-value thresholds. Although this behavior is not desirable, it is tolerable since the number of false positives produced by WSARE differs by a small amount from the count generated by the standard algorithm. In Fig. 1, there are at most three more false positives identified by WSARE that were not identified by the standard algorithm.

We now show some of the rules learned by WSARE. The rules below were obtained from one of the result-generating simulations.

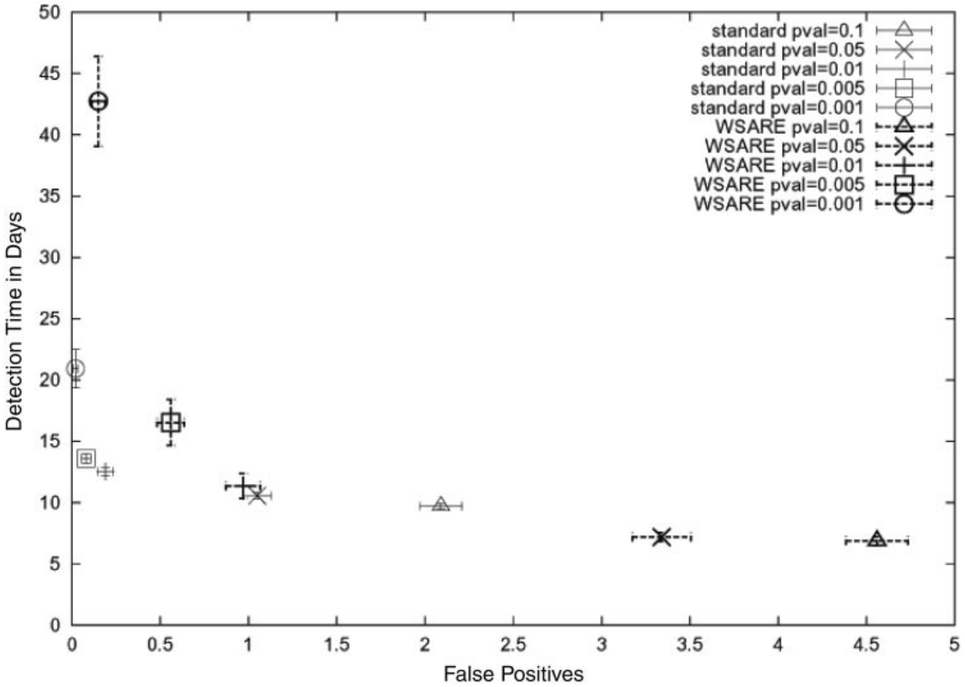

FIGURE 1. Scatterplot of detection time versus false positives with error bars for detection time and false positives.

```
### Rule 1: Sat Day97 (daynum 97, dayindex 97)
SCORE = -0.00000011 PVALUE = 0.00249875
33.33% (16/48) of today's cases have Age Decile = 5 and Gender = Male
3.85% (7/182) of other cases have Age Decile = 5 and Gender = Male
### Rule 2: Tue Day100 (daynum 100, dayindex 100)
SCORE = -0.00001093 PVALUE = 0.02698651
30.19% (16/53) of today's cases have Age Decile = 5 and Column less than 25
6.19% (12/194) of other cases have Age Decile = 5 and Column less than 25
```

In rule 1, WSARE demonstrates that it is capable of finding the target demographic group that Epidemic0 infects. This rule proves to be significant above the 99% level. On the other hand, rule 2 discovers something that was not deliberately hard coded into Epidemic0. Rule 2 states that, on day 100, an unusually large number of cases involves people in their 50s in the left half of the grid. Since we designed the people in the simulation to interact with places that are in geographic proximity to their homes, we suspected that the locality of interaction of infected individuals would form some spatial clusters of ED cases. On further inspection of the log files, we discovered that 12 of the 16 cases from the current day that satisfied this rule were, in fact, caused by Epidemic0. This example illustrates the capability of WSARE to detect significant anomalous patterns that are completely unexpected.

Results From Real Emergenct Department Data

We also ran WSARE on actual ED data collected from hospitals in a major US city. This database contained approximately 70,000 records collected during a period of

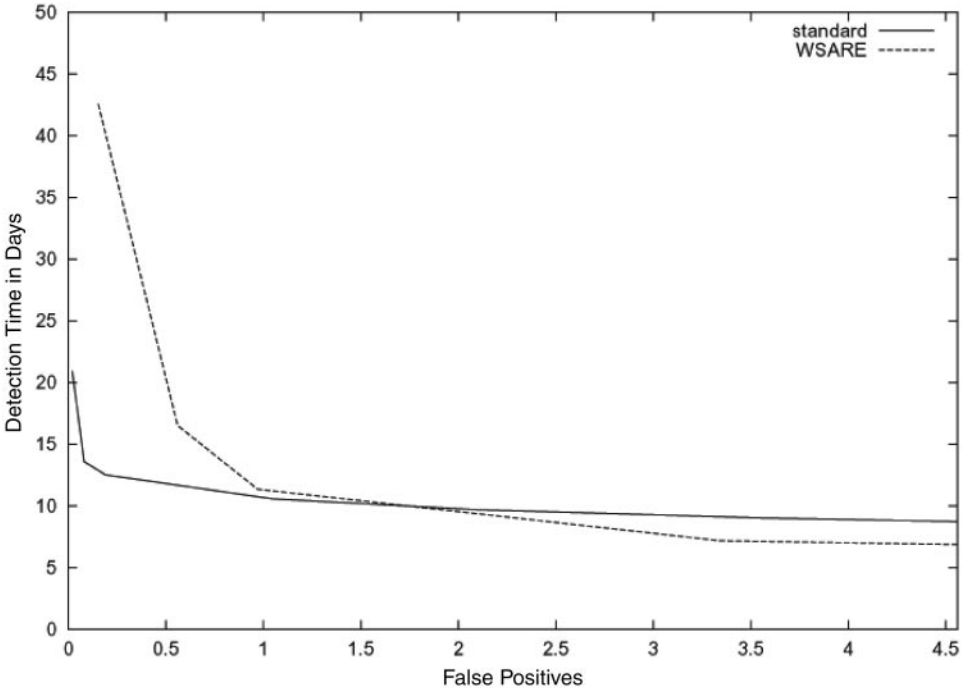

FIGURE 2. Plot of detection time versus false positives.

505 days. Since we are looking at historical data, we need to use FDR to determine which of the P values is significant. The results are shown below with $\alpha = .1$ for FDR.

```
### Rule 1: Tue 05-16-2000 (daynum 36661, dayindex 18)
SCORE = -0.00000000 PVALUE = 0.00000000
32.84% (44/134) of today's cases have Time Of Day after 6:00 pm
90.00% (27/30) of other cases have Time Of Day after 6:00 pm
### Rule 2: Fri 06-30-2000 (daynum 36706, dayindex 63)
SCORE = -0.00000000 PVALUE = 0.00000000
19.40% (26/134) of today's cases have Place = NE and Lat = d
5.71% (16/280) of other cases have Place = NE and Lat = d
### Rule 3: Wed 09-06-2000 (daynum 36774, dayindex 131)
SCORE = -0.00000000 PVALUE = 0.00000000
17.16% ( 23/134) of today's cases have Prodrome = Respiratory and Age less
than 40
4.53% ( 12/265) of other cases have Prodrome = Respiratory and Age less than
40
### Rule 4: Fri 12-01-2000 (daynum 36860, dayindex 217)
SCORE = -0.00000000 PVALUE = 0.00000000
22.88% ( 27/118) of today's cases have Time Of Day after 6:00 pm and Lat = s
8.10% ( 20/247) of other cases have Time Of Day after 6:00 pm and Lat = s
### Rule 5: Sat 12-23-2000 (daynum 36882, dayindex 239)
SCORE = -0.00000000 PVALUE = 0.00000000
```

```
18.25% ( 25/137) of today's cases have ICD9 = shortness of breath and Time Of Day before 3:00 pm
5.12% ( 15/293) of other cases have ICD9 = shortness of breath and Time Of Day before 3:00 pm
### Rule 6: Fri 09-14-2001 (daynum 37147, dayindex 504)
SCORE = -0.00000000 PVALUE = 0.00000000
66.67% ( 30/ 45) of today's cases have Time Of Day before 10:00 am
18.42% ( 42/228) of other cases have Time Of Day before 10:00 am
```

Rule 1 notices that there are fewer cases after 6:00 PM, possibly due to a lack of reporting by some hospitals. Rule 6 correctly identifies a larger volume of data being collected before 10:00 AM on day 504. Since day 504 was the last day of this database, this irregularity was the result of the database being given to us in the morning.

We are beginning the process of using input from public health officials of the target city to help us validate and measure the performance of WSARE.

DISCUSSION

WSARE successfully identified anomalous patterns in the data. Our simulation results indicate that WSARE has significantly lower detection times than a standard detection algorithm, provided the *P*-value threshold is not extremely low. This should not be a problem since most anomalies are reported at a significance level of 95% or 99%, corresponding, respectively, to *P* values of .05 and .01. WSARE also has a slightly higher false positive rate than the standard algorithm. This difference, however, was approximately three more false positives in the worst case for our particular simulation.

WSARE is designed to be a general safety net for emerging problems that have not been anticipated. It is intended to operate with a suite of other detectors in which each detector is tuned to look for a specific disease. WSARE has been experimentally deployed in Utah and Pennsylvania.

We believe the three main innovations in this article are

1. Turning the problem of detecting the emergence of new patterns in recent data into the question, Is it possible to learn a propositional rule that can significantly distinguish whether records are most likely to have come from the recent past or from the more distant past?
2. Incorporating several levels of significance tests into rule learning to avoid several levels of overfitting caused by intensive multiple testing.
3. Examining the domain of early outbreak detection by means of machine learning tools.

ACKNOWLEDGEMENT

This work was sponsored by a grant from the DARPA IAO Biosurveillance Program and by the State of Pennsylvania.

We thank Howard Burkom, Rich Tsui, Bob Olszewski, Jeremy Espino, and Jeff Schneider for their help and suggestions. We would also like to thank the reviewers for the *Journal of Urban Health* for their insightful comments and suggestions.

REFERENCES

1. Wong W, Moore AW, Cooper G, Wagner M. Rule-based anomaly pattern detection for detecting disease outbreaks. In: Ford K, ed. *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02)*. Cambridge, MA: MIT Press; 2002:217–223. Available at: www.autonlab.org/.
2. Lane T, Brodley CE. Temporal sequence learning and data reduction for anomaly detection. *ACM Trans Inform Syst Security*. 1999;2:295–331.
3. Eskin E. Anomaly detection over noisy data using learned probability distributions. In: Langley P, ed. *Proceedings of the 2000 International Conference on Machine Learning (ICML-2000)*. San Francisco: Morgan Kaufmann; 2000:255–262.
4. Maxion RA, Tan KMC. *Anomaly Detection in Embedded Systems*. Pittsburgh, PA: Carnegie Mellon University; 2001. Technical Report CMU-CS-01-157.
5. Wagner MM, Tsui FC, Espino JU, et al. The emerging science of very early detection of disease outbreaks. *J Public Health Manage Pract*. 2001;7(6):51–59.
6. Bishop CM. Novelty detection and neural network validation. *IEEE ProcVision, Image Signal Proc*. 1994;141:217–222.
7. Hamerly G, Elkan C. Bayesian approaches to failure prediction for disk drives. In: Brodley CE, Danyluk AP, eds. *Proceedings of the 18th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann; 2001:202–209.
8. Bay SD, Pazzani MJ. Detecting change in categorical data: Mining contrast sets. In: Chaudhuri S, Madigan D, eds. *Knowledge Discovery and Data Mining*. New York: Association for Computing Machinery; 1999:302–306.
9. Brin S, Motwani R, Ullman JD, Tsur S. Dynamic itemset counting and implication rules for market basket data. In: Peckham J, ed. *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13–15, 1997, Tucson, Arizona, USA*. New York: ACM Press; 1997:255–264.
10. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J Roy Stat Soc, Series B*. 1995;57:289–300.
11. Good P. *Permutation Tests—a Practical Guide to Resampling Methods for Testing Hypotheses*. 2nd ed. New York, NY: Springer-Verlag; 2000.
12. Miller CJ, Genovese C, Nichol RC, et al. *Controlling the False Discovery Rate in Astrophysical Data Analysis*. Pittsburgh, PA: Carnegie Mellon University, 2001. Technical report #747.