UNIVERSITÉ DE MONTRÉAL

MACHINE LEARNING FOR DISEASE OUTBREAK DETECTION USING PROBABILISTIC MODELS

NASTARAN JAFARPOUR KHAMENEH
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
DÉCEMBRE 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

MACHINE LEARNING FOR DISEASE OUTBREAK DETECTION USING PROBABILISTIC MODELS

présentée par : JAFARPOUR KHAMENEH Nastaran
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. GALINIER Philippe, Doctorat, président
M. DESMARAIS Michel C., Ph.D., membre et directeur de recherche
Mme PRECUP Doina, Ph.D., membre et codirectrice de recherche
M. PAL Christopher J., Ph.D., membre
M. PECHENIZKIY Mykola, Ph.D., membre externe

To my beloved family

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Doina Precup who has provided constant guidance and encouragement throughout my research at McGill University. I deeply thank my advisor Michel Desmarais for his time and support during my studies at Ecole Polytechnique de Montreal. I do not know where my research would go without their patience and efforts.

I am sincerely grateful to David L. Buckeridge for his invaluable guidance and also funding this research. Also, I would like to express my special appreciation and thanks to Masoumeh Izadi for her great support and time in progressing the research.

I am using this opportunity to express my gratitude to Anya Okhmatovskaia and Aman Verma for providing the simulated data used in this research and their constant help in addressing related issues. I also thank Rolina van Gaalen for co-working in Cost analysis.

I thank the former and current members of the Reasoning and Learning Lab in the School of Computer Science at McGill. I consider myself fortunate to benefit tremendously from such a creative and smart group of people. I am sincerely thankful to my labmates in Department of Computer Engineering at Ecole Polytechnique de Montreal and members of Surveillance Lab in the Clinical and Health Informatics Research Group at McGill University.

I would like to thank administrative staff and system administrators in the School of Computer Science and Surveillance Lab at McGill, and Department of Computer Engineering at Ecole Polytechnique de Montreal for their incredible helps.

At the end, words cannot express how grateful I am to my family who never stopped supporting me even from distance. A special thanks to my friends who always inspire me to strive towards my goals.

RÉSUMÉ

L'expansion de maladies connues et l'émergence de nouvelles maladies ont affecté la vie de nombreuses personnes et ont eu des conséquences économiques importantes. L'Ébola n'est que le dernier des exemples récents. La détection précoce d'infections épidémiologiques s'avère donc un enjeu de taille. Dans le secteur de la surveillance syndromique, nous avons assisté récemment à une prolifération d'algorithmes de détection d'épidémies. Leur performance peut varier entre eux et selon différents paramètres de configuration, de sorte que l'efficacité d'un système de surveillance épidémiologique s'en trouve d'autant affecté. Pourtant, on ne possède que peu d'évaluations fiables de la performance de ces algorithmes sous différentes conditions et pour différents types d'épidémie. Les évaluations existantes sont basées sur des cas uniques et les données ne sont pas du domaine public. Il est donc difficile de comparer ces algorithmes entre eux et difficile de juger de la généralisation des résultats. Par conséquent, nous ne sommes pas en mesure de déterminer quel d'algorithme devrait être appliqué dans quelles circonstances.

Cette thèse poursuit trois objectifs généraux : (1) établir la relation entre la performance des algorithmes de détection d'épidémies et le type et la sévérité de ces épidémies, (2) améliorer les prédictions d'épidémies par la combinaison d'algorithmes et (3) fournir une méthode d'analyse des épidémies qui englobe une perspective de coûts afin de minimiser l'impact économique des erreurs du type faux positifs et faux négatifs.

L'approche générale de notre étude repose sur l'utilisation de données de simulation d'épidémies dont le vecteur de transmission est un réseau d'aqueducs. Les données sont obtenues de la plateforme de simulation SnAP du Department of Epidemiology and Biostatistics Surveillance Lab de l'université McGill. Cette approche nous permet de créer les différentes conditions de types et d'intensités d'épidémiologie nécessaires à l'analyse de la performance des algorithmes de détection.

Le premier objectif porte sur l'influence des différents types et différentes intensités d'épidémiologie sur la performance des algorithmes. Elle est modélisée à l'aide d'un modèle basé sur un réseau bayésien. Ce modèle prédit avec succès la variation de performance observée dans les données. De plus, l'utilisation d'un réseau bayésien permet de quantifier l'influence de chaque variable et relève aussi le rôle que jouent d'autres paramètres qui étaient jusqu'ici ignorés dans les travaux antérieurs, à savoir le seuil de détection et l'importance de tenir compte de récurrences hebdomadaires.

Le second objectif vise à exploiter les résultats autour du premier objectif et de combiner les algorithmes pour optimiser la performance en fonction des facteurs d'influence. Les résultats des algorithmes sont combinés à l'aide de la méthode de Mixture hiérarchique d'expert (Hierarchical Mixture of Experts—HME). Le modèle HME est entraîné à pondérer la contribution de chaque algorithme en fonction des données. Les résultats de cette combinaison des résultats d'algorithmes sont comparables avec les meilleurs résultats des algorithmes individuels, et s'avèrent plus robustes à travers différentes variations. Le niveau de contamination n'influence pas la performance relative du modèle HME.

Finalement, nous avons tenté d'optimiser des méthodes de détection d'épidémies en fonction des coûts et bénéfices escomptés des prédictions correctes et incorrects. Les résultats des algorithmes de détection sont évalués en fonction des décisions possibles qui en découlent et en tenant compte de données réelles sur les coûts totaux d'utilisation des ressources du système de santé. Dans un premier temps, une régression polynomiale permet d'estimer le coût d'une épidémie selon le délai de détection. Puis, nous avons développé un modèle d'apprentissage d'arbre de décision qui tient compte du coût et qui prédit les détections à partir des algorithmes connus. Les résultats expérimentaux démontrent que ce modèle permet de réduire le coût total des épidémies, tout en maintenant le niveau de détection des épidémies comparables à ceux d'autres méthodes.

ABSTRACT

The past decade has seen the emergence of new diseases or expansion of old ones (such as Ebola) causing high human and financial costs. Hence, early detection of disease outbreaks is crucial. In the field of syndromic surveillance, there has recently been a proliferation of outbreak detection algorithms. The choice of outbreak detection algorithm and its configuration can result in important variations in the performance of public health surveillance systems. But performance evaluations have not kept pace with algorithm development. These evaluations are usually based on a single data set which is not publicly available, so the evaluations are difficult to generalize or replicate. Furthermore, the performance of different algorithms is influenced by the nature of the disease outbreak. As a result of the lack of thorough performance evaluations, one cannot determine which algorithm should be applied under what circumstances.

Briefly, this research has three general objectives: (1) characterize the dependence of the performance of detection algorithms on the type and severity of outbreak, (2) aggregate the predictions of several outbreak detection algorithms, (3) analyze outbreak detection methods from a cost-benefit point of view and develop a detection method which minimizes the total cost of missing outbreaks and false alarms.

To achieve the first objective, we propose a Bayesian network model learned from simulated outbreak data overlaid on real healthcare utilization data which predicts detection performance as a function of outbreak characteristics and surveillance system parameters. This model predicts the performance of outbreak detection methods with high accuracy. The model can also quantify the influence of different outbreak characteristics and detection methods on detection performance in a variety of practically relevant surveillance scenarios. In addition to identifying outbreak characteristics expected to have a strong influence on detection performance, the learned model suggests a role for other algorithm features, such as alerting threshold and taking weekly patterns into account, which was previously not the focus of attention in the literature.

To achieve the second objective, we use Hierarchical Mixture of Experts (HME) to combine the responses of multiple experts (i.e., predictors) which are outbreak detection methods. The contribution of each predictor in forming the final output is learned and depends on the input data. The developed HME algorithm is competitive with the best detection algorithm in the experimental evaluation, and is more robust under different circumstances. The level of contamination of the surveillance time series does not influence the relative performance of the HME.

The optimization of outbreak detection methods also relies on the estimation of future benefits of true alarms and the cost of false alarms. In the third part of the thesis, we analyze some commonly used outbreak detection methods in terms of the cost of missing outbreaks and false alarms, using simulated outbreak data overlaid on real healthcare utilization data. We estimate the total cost of missing outbreaks and false alarms, in addition to the accuracy of outbreak detection and we fit a polynomial regression function to estimate the cost of an outbreak based on the delay until it is detected. Then, we develop a cost-sensitive decision tree learner, which predicts outbreaks by looking at the prediction of commonly used detection methods. Experimental results show that using the developed cost-sensitive decision tree decreases the total cost of the outbreak, while the accuracy of outbreak detection remains competitive with commonly used methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUC	Area Under Curve
BN	Bayesian Networks
CPD	Conditional Probability Distribution
CUSUM	Cumulative Sum Control Chart
ED	Emergency Department
EM	Expectation-Maximization
EWMA	Exponentially Weighted Moving Average
GLRT	Generalized Likelihood Ratio Test
HME	Hierarchical Mixture of Experts
REP Tree	Reduced Error Pruning Tree
ROC	Receiver Operating Characteristic
SnAP	Simulation Analysis Platform

CHAPTER 1
INTRODUCTION

1.1 Problem Definition and Challenges

Infectious disease outbreaks results in high human and financial costs. The past decade has seen the emergence of diseases caused by previously unrecognized threats or the sudden appearance of known diseases in the environment. For instance, the Centre of Disease Control and Prevention in USA (CDC) estimated that between 43 million and 89 million cases of the new influenza A (H1N1) epidemic occurred between April 2009 and April 2010. Also between 8,880 and 18,300 H1N1-related deaths occurred in the same time frame [1]. Since the probability of major infectious disease outbreaks is very high, their early detection is a crucial task in order to prevent or reduce the large spread of diseases. Preparing against the risk of future outbreaks may help in managing the delivery of medical care and reduce the huge costs associated with it.

One of the most challenging aspects of public health surveillance is the early recognition of infectious disease outbreaks which have the potential of high morbidity and mortality. Due to the volume of data collected in public health agencies, all data cannot be reviewed manually by surveillance analysts. Modern electronic data collection methods make it necessary to automate outbreak detection processes (Lombardo and Buckeridge, 2007).

Automated surveillance systems can monitor clinical data drawn from multiple sources, with the goal of detecting potential disease outbreaks rapidly and accurately. The role of outbreak detection processing is to screen large volumes of data and issue alerts to draw an epidemiologist's attention to anomalies (Buckeridge et al., 2008a). Automated health surveillance systems apply statistical algorithms to detect anomalies. When an outbreak occurs, the care-seeking infected population adds a signal to the background data. Detecting these changes in the number of people infected is the basis of calling an outbreak.

Examples of surveillance time series are the daily number of visits to a clinic or hospital, the daily number of over-the-counter sales of specific medicines, school and work absenteeism rates, and others. Some epidemiological investigations use spatial location in surveillance data because a group of patients in a same location who become ill at the same time may indicate a cluster of cases caused by a common exposure. These methods apply different algorithms to detect cases separately for different spatial regions (Lombardo and Buckeridge, 2007). Our work is focused on

1. http://www.cdc.gov/h1n1flu/cdcresponse.htm

time series without special treatment of space.

The detection process of a possible outbreak consists of two phases: the prediction of expected data values, and the determination of anomalies. Using the available data history, also called baseline time series, the prediction phase estimates the quantity of interest for a time interval. An anomaly is determined by calculating the difference between the estimated value and the actual observed value and applying statistical methods to decide if the difference is unusual. In addition to prediction-based methods, other anomaly detection approaches, such as searching for unexplained patterns are also possible (Tokars et al., 2009a). However,we will focus on prediction-based algorithms, as they are more wide-spread in practice.

An important aspect of a detection algorithm is the computation of the baseline data. The simplest approach is to average the number of observed cases over some period of time. However, this approach cannot account for systematic data patterns and may generate irrelevant alarms. For example, visit counts of a clinic which is closed on Sundays and holidays may have a sharp rise on Mondays and after holidays. Frequent statistical alerts will lead to the system being ignored, which in turn will result in a loss of sensitivity to true events and undermine the utility and credibility of the system. Modelling the day-of-week effects, long-term trends, and other known systematic patterns will increase the sensitivity of the detection algorithm over repeated cycles of data (Lombardo and Buckeridge, 2007).

Some practical outbreak detection algorithms were derived from probability-based process control charts which are widely used in monitoring industrial processes. These methods help the early recognition of unusual changes in the data under control. For example, outbreak detection methods like the C-family algorithms and adaptive Exponentially Weighted Moving Average (EWMA) were inspired by Cumulative Sum Control Chart (CUSUM) and EWMA control charts. Some detector algorithms adapt regression modelling for syndromic surveillance data. Also, several modifications of these algorithms, like stratification of the data by weekdays versus weekend days and using data ratios instead of absolute data values, have been proposed.

In the field of syndromic surveillance, there has recently been a proliferation of outbreak detection algorithms. The choice of outbreak detection algorithm and its configuration can result in important variations in the performance of public health surveillance systems. The performance of these algorithms is evaluated in terms of sensitivity and specificity of detection and time to detection. They can also be evaluated in terms of outcomes, such as infections averted. Sensitivity is the probability that a public health event of interest will be detected in the data given that the event really occurred. Specificity is the probability that no health event will be detected when no

such event has in fact occurred (Lombardo and Buckeridge, 2007). Time to detection is the time duration from the beginning of the outbreak to the first day on which the outbreak is detected.

However, evaluations of these new algorithms remain highly circumscribed. These evaluations are usually based on a single data set which is not publicly available, so the evaluations are difficult to generalize or replicate. Furthermore, the performance of different algorithms is influenced by the nature of the disease outbreak. As a result of the lack of thorough performance evaluations, one cannot determine which algorithm should be applied under what circumstances.

The work in this thesis used data provided through simulations from the Surveillance Lab of the Department of Epidemiology and Biostatistics at McGill University. The Simulation Analysis Platform (SnAP), developed by that group is a software infrastructure for the automatic deployment and analysis of multiple runs of a simulation model. Simulation models in this context consist of simulated syndromic data and different types of outbreak scenarios. This platform can efficiently explore the influence of parameter adjustments (Buckeridge et al., 2011). The simulated data will be described in section 2.3.1.

Although using real disease outbreak data seems to be the best option for the evaluation of algorithms, it has several drawbacks. First, it is often difficult to precisely determine the starting time and the size of the outbreak. Second, few historical data containing verified real disease outbreaks are available. For this reason, the detection performance is usually evaluated by simulating disease outbreaks. Performance evaluation via simulation can be repeated with variations of outbreak characteristics as much as desired and provides ground truth that is not usually available in real data (Lombardo and Buckeridge, 2007).

1.2 Research Questions

In this thesis, we used machine learning methods to explore three questions of great practical importance.

- In the field of syndromic surveillance, how does the performance of outbreak detection algorithms depend on the type and severity of outbreaks?
- Can the performance of outbreak detection algorithms, in terms of time to detection, the number of false alerts and true alerts, be improved by combining the prediction of several detection algorithms?
- Which approach is optimal for outbreak detection from a cost-benefit point of view?

1.3 General Objectives

This research has three general objectives. The first objective is to characterize the dependence of the performance of detection algorithms on the type and severity of outbreak, to develop and evaluate a probabilistic model for discovering determinants of outbreak detection and quantifying the effect of determinants on detection performance, and to predict the performance of outbreak detection algorithms under different circumstances. Having good predictors will guide the method selection and algorithm configuration in surveillance systems.

The second research objective is to study how one can aggregate the predictions of several outbreak detection algorithms and investigate whether this can enhance performance, compared to using single methods.

The third objective is to analyze existing outbreak detection methods from a cost-benefit point of view and to develop a detection method whose goal is not only predicting the outbreaks accurately, but also minimizing the total cost of missing outbreaks and false alarms.

1.4 Hypotheses

The research in this thesis tests the following hypotheses:

Hypothesis 1: The performance of detection algorithms depends on the type and severity of outbreaks. Specifically, different algorithms are best for different types of outbreaks.

Hypothesis 2: Aggregating the predictions of different detectors based on performance characteristics will improve the performance and the robustness of outbreak detection in syndromic surveillance.

Hypothesis 3: Different detection algorithms are optimal depending on the relative cost of false alarms vs. delays in detection.

1.5 Main Contributions

The main contributions of this thesis can be summarized as follow:

- Our work aimed to characterize the dependence of the performance of detection algorithms on the type and severity of outbreak. For example, if the magnitude of an outbreak is very large, like the SARS exposure, most of the detection methods can pick it up. So, the capability of early recognition of non-obvious outbreaks is the discriminative feature of detectors. We learned a Bayesian network model from simulated outbreak data overlaid on real healthcare utilization

data to predict detection performances as a function of outbreak characteristics and surveillance system parameters. This model can predict the performance metrics of commonly used outbreak detection methods with high accuracy. The model can also quantify the influence of different outbreak characteristics and detection methods on detection performance in a variety of practically relevant surveillance scenarios. In addition to identifying characteristics expected to have a strong influence on detection performance, the learned model suggests a role for other algorithm features which was previously not the focus of attention in the literature. This contribution is discussed in detail in Chapter 3.

- We investigated how outbreak detection methods can be combined in order to improve the overall detection performance. We used Hierarchical Mixture of Experts (HME), a probabilistic model for combining classification methods which has been well-studied in computer science and statistics. This algorithm uses the divide-and-conquer strategy to combine the responses of multiple experts (i.e., predictors) and form a single response. The contribution of each predictor in forming the final output is adjustable based on the input data. We used simulated surveillance data to train an HME in order to aggregate predictions from several outbreak detection methods. The developed HME algorithm was competitive with the best detection algorithm in the experimental evaluation. The developed detection algorithm based on HME was more robust under different circumstances and the level of contamination of surveillance time series did not influence the relative performance of the HME. This contribution is presented in Chapter 4.
- The optimization of outbreak detection methods also relies on the estimation of future benefits of true alarms and the cost of false alarms. For example, in the case of an anthrax attack, delays of hours in detection and taking the decision to intervene can lead to hundreds of lives lost and millions of additional expenses (Izadi and Buckeridge, 2007). On the contrary, if the system generates many false alarms, the alarms will be ignored by public health personnel and the credibility of the system decreases. So cost is an important determinant in selecting optimal outbreak detection methods. We analyzed some commonly used outbreak detection methods in terms of the cost of missing outbreaks and false alarms, using simulated outbreak data overlaid on real healthcare utilization data. We estimated the total cost of missing outbreaks and false alarms, in addition to the accuracy of outbreak detection. We fitted a polynomial regression function to estimate the cost of an outbreak based on the delay until it is detected. Then, we developed a cost-sensitive decision tree learner, which predicts outbreaks by looking at the prediction of commonly used detection methods. Experimental results showed that using the developed cost-sensitive decision tree decreased the total cost of outbreak, while the accuracy of outbreak detection remained competitive with commonly used methods. The cost- benefit analysis and proposed algorithm are discussed in Chapter 5.

This research mainly focuses on temporal outbreak detection algorithms; however many of these concepts and techniques will be applicable in the spatio-temporal setting too.

1.6 Publications

1. **N. Jafarpour**, D. Precup, M. Izadi, D. L. Buckeridge, "Cost Analysis of Outbreak Detection Methods", Submitted in *Journal of Artificial Intelligence in Medicine*.
2. **N. Jafarpour**, D. Precup, "Cost Analysis of Outbreak Detection Methods", Poster in *Women in Machine Learning Workshop (WiML 2014), Co-located with NIPS*, December 2014, Montreal, CANADA.
3. **N. Jafarpour**, M. Izadi, D. Precup, D. L. Buckeridge, "Quantifying the Determinants of Outbreak Detection Performance through Simulation and Machine Learning", *Journal of Biomedical Informatics*, November 2014.
4. **N. Jafarpour**, D. Precup, M. Izadi, D. L. Buckeridge, "Using Hierarchical Mixture of Experts Model for Fusion of Outbreak Detection Methods", *American Medical Informatics Association Annual Symposium (AMIA 2013)*, November 2013, Washington D.C., US.
5. **N. Jafarpour**, D. Precup, D. L. Buckeridge, "Determinants of Outbreak Detection Performance", *International Society for Disease Surveillance (ISDS 2012)*, December 2012, San Diego, US.

1.7 Organization Of the Thesis

We review some of related literature on outbreak detection algorithms and performance measurements in Chapter 2. The main contributions of the research are explained in details in Chapter 3, 4, and 5 as summarized above. Finally, we conclude and outline future work in Chapter 6.

CHAPTER 2
LITERATURE REVIEW

According to Fricker (2011, p. 403), "Biosurveillance is the process of actively gathering and analyzing data related to human health and disease in order to obtain early warning of human disease events, and overall awareness of disease activities in the population." Outbreak detection algorithms play a central role in automated biosurveillance, analyzing large volumes of clinical data in real-time and detecting the potential of a disease outbreak.

In this chapter, we review the basics of surveillance data and its behaviour and some of the existing outbreak detection algorithms.We discuss the evaluation process of outbreak detection algorithms and introduce the main evaluation measures. Next, we review basic concepts of evaluating learned hypotheses. Details of the learning algorithms that we build on will be reviewed as needed in the later chapters.

2.1 Basic Description of Surveillance Data

The most common sources of data for syndromic surveillance are patient visits in hospital emergency departments and clinics, over the counter medication sales in pharmacies, calls to emergency centres, and school absenteeism. These data are called *surveillance data*. Surveillance data are gathered in the form of time series. In the absence of an outbreak event, surveillance data is called background data. Whenever an outbreak event occurs, the care-seeking infected population adds an outbreak signal to the surveillance data. The automated biosurveillance system analyzes the surveillance data to find outbreak patterns.

Background data collected over several years of monitoring is useful to observe *long-term trends*, i.e., general changes that are not repeated over the range of data during analysis. The most common long-term trend in surveillance data is due to the increase or decrease in the size of the population under monitoring. Long-term trends usually add a linear component over the time series. For example, every year the population may increase by 1000 people, or by 15% [1].

In addition to long-term trends, there exist cyclic patterns in surveillance data which are repeated in time intervals. These patterns are called *seasonal trends*. An example of seasonal pattern is the increase in the number of clinical visits for influenza-like illnesses during autumn and winter. This

1. Time Series Analysis (http://www.statsoft.com/textbook/time-series-analysis)

change is not a long-term trend, because after the cold season of the year, the number of visits decreases, and it is cyclic because every year the pattern is repeated.

Another pattern that might be observed in surveillance data is called *day-of-week effect*. For example, if the visit counts are monitored in a clinic that is closed on holidays and weekends, there is a sharp increase in the number of visits on Monday and after holidays, but this increase should not be considered as an outbreak (Lombardo and Buckeridge, 2007). As an example, Figure 2.1 shows seasonal and day-of-week effects in two biosurveillance time series. This time series consists of daily counts of respiratory visits in a large area over 3 years of surveillance. While seasonal trends are obvious in the upper plot, the lower plot magnifies the reduced visits on holidays and weekends (Lombardo and Buckeridge, 2007).

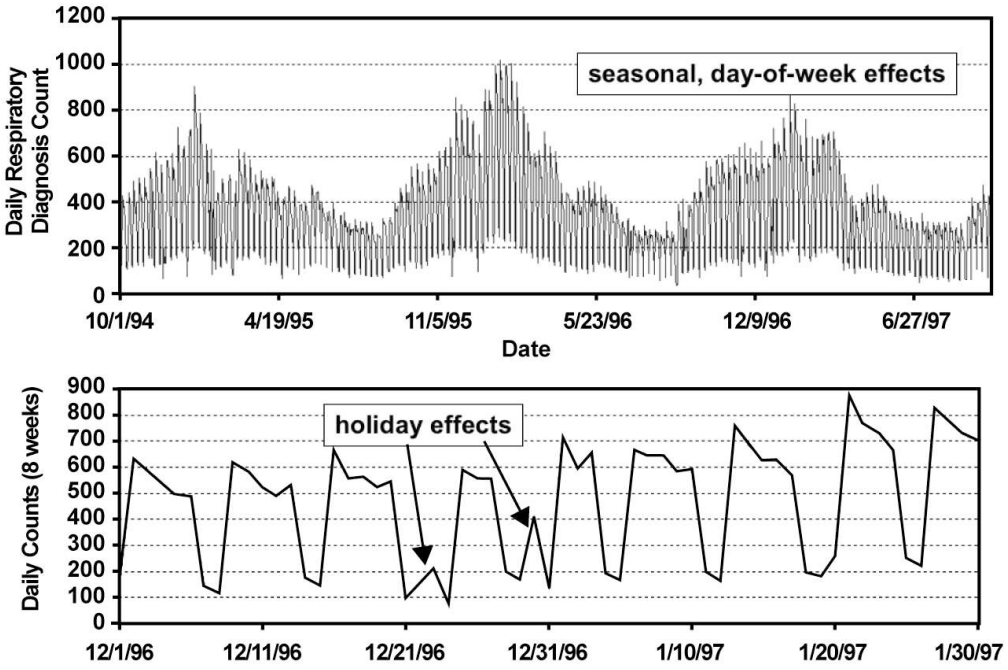

Figure 2.1 Seasonal trends and day-of-week effects in surveillance time series (Lombardo and Buckeridge, 2007)

The day-of-week effects, seasonal trends, and long-term trends caused by natural variations can be interpreted as expected background patterns. They are expected because the required information for modelling these patterns is available. However, the interpretation as background noise is variable: an event that is significant in one context may be irrelevant in another. For example, the onset of seasonal flu is an outbreak signal in a system with the goal of general public health surveillance, but it is noise to a system concerned with the detection of anthrax attacks (Lombardo and Buckeridge, 2007).

One external influence on surveillance data is the variation in the care-seeking population. In analyzing surveillance data, it is best to estimate the proportion of infected people rather than using the number of infected people, as the latter is influenced by the size of the population. For example, a sudden increase in the number of flu cases in a military treatment facility after a large recruitment is not a sign of a flu outbreak. Conversely, a sudden drop in the size of the population might mask a real outbreak, if it is simultaneous with a real outbreak. Hence, epidemiologists usually estimate *rates* across time periods, as the number of infected people in a population at a given time divided by population size at that time (Lombardo and Buckeridge, 2007).

Ignoring the influence of the systematic patterns and variations of the population in modelling background data reduces the performance of outbreak detection algorithms, so the analysis of background data is necessary before applying outbreak detection algorithms. We will now discuss outbreak detection algorithms.

2.2 Outbreak Detection Algorithms

Early detection of disease outbreaks can reduce the disease impact and save lives, but this type of investigation is costly and excessive false alarms weaken the credibility of the alerting system. So, biosurveillance system developers look for outbreak detection algorithms with high sensitivity and few false alarms.

An outbreak detection algorithm looks for significant differences between the expected value and the observed value of the surveillance time series at a time t. If this difference exceeds a predefined threshold, the algorithm triggers an alert and warns of an outbreak event. The time interval over which the difference is computed is called *test interval*. Most detection algorithms work based on daily test intervals. The portion of the surveillance time series used to calculate the expected value is called *baseline interval*. For example, in an outbreak detection algorithm that uses the average number of visits to a clinic for a specific disease in last seven days, the length of the baseline interval is 7.

If the parameters and thresholds of outbreak detection algorithms are adjusted according to recent data, the algorithms are called *adaptive*. In non-adaptive algorithms, the entire historical surveillance data is used as a baseline interval, while the baseline interval of adaptive algorithms is a sliding window in the time series.

2.2.1 C-algorithms

The C-family of outbreak detection algorithms was developed in the Early Aberration Reporting System (EARS) and is widely used by the Centre of Disease Control and Prevention, USA (CDC) and other public health departments. The C-family algorithms, C1, C2, and C3, are adaptive algorithms developed based on the process control chart concept used for controlling industrial processes (Hutwagner et al., 2003). These algorithms take advantage of the Central Limit Theorem, which states that a series of means approaches a Gaussian distribution as the size of series increases (Lombardo and Buckeridge, 2007). So, the expected value of the time series for time t is simply the mean of the values observed during the baseline interval. Generally, C-algorithms measure the variation of the observed value from the mean of the baseline interval. If this variation is much bigger than the standard deviation of the data from the baseline interval (e.g., three times above standard deviation), an unusual event is flagged and the possibility of a disease outbreak is signalled.

The C1 and C2 algorithms use Equation (2.1) for deriving the test statistic from the baseline interval.

$$testStatistic_t = \frac{observed_t - mean_{baseline}}{standardDeviation_{baseline}} \tag{2.1}$$

An alert is issued if $testStatistic_t > 3$. The threshold 3 used in the EARS system was derived empirically. Since the baseline interval is a sliding window and moves forward each day, the parameters of the C-algorithms (i.e. the mean and standard deviation of the baseline) change every day, so the C-algorithms are adaptive.

The difference between the different C algorithms is in the configuration of two parameters, as we will now describe. The length of the baseline interval for all C algorithms is 7 days. The C1 algorithm uses the baseline interval from day $t-7$ to $t-1$ for computing the test statistic used on day t. Generally, gradually increasing outbreaks can bias the test statistic upward, so the detection algorithm will fail to flag the outbreak. To avoid this situation, the solution is to use a gap between the baseline interval and the test interval. This gap is called *guardband* or *buffer interval*. The C2 algorithm uses a 2-day guardband interval, so its baseline interval is from day $t-9$ to $t-2$. The use of the guardband in C2 improves the sensitivity of outbreak detection compared with C1, because outbreaks spread over several days are not missed.

The C3 algorithm also uses a 2-day guardband and the baseline interval is from day $t-9$ to day $t-2$. But, C3 evaluates the variation of the test statistic over 3 recent days, t, $t-1$, and $t-2$ instead of just day t. An alert is issued if $testStatistic_t + testStatistic_{t-1} + testStatistic_{t-2} > 3$.

C3 has better sensitivity because small changes in the time series are cumulated, enabling earlier detection. If the C2 algorithm produces an alert, C3 will alert too. However, C3 is also more prone to generate false alarms. To summarize, C1 is the least sensitive because of the lack of a guardband, C2 has moderate sensitivity, and C3 is the most sensitive algorithm, using guardband and the average of the three recent observations. Recalling the systematic patterns of surveillance time series introduced in section 2.1, the C-algorithms take seasonal trends into account, because the mean and standard deviation are calculated in generally speaking the same season (cycle) as the test interval (Hutwagner et al., 2005).

The simplicity of the C-algorithms has made them popular and widely used among public health departments, and several modifications of these algorithms have been adopted by local users. However, the C-algorithms have several drawbacks. First, the short baseline of 7 days makes them volatile, since the threshold ($3 \times standardDeviation_{baseline}$) fluctuates noticeably day-to-day. Second, the assumption of the Central Limit Theorem is a Gaussian distribution for the average of input time series. Most surveillance time series have a negative binomial or Poisson distribution. These distributions have longer tails than the Gaussian distribution, so the number of false alarms increases. In practice, statistical and heuristic measures are applied to reduce the number of false alarms (Lombardo and Buckeridge, 2007). These led to a lot of work on modifications of the C-algorithms, especially C2, because of its moderate sensitivity. Here, we describe the most important modifications.

As we discussed earlier, there might be weekly patterns in the surveillance time series, which the C-algorithms do not take into account. Tokars et al. (2009b) proposed a simple method to adjust the C-algorithms for time series with weekly patterns. This method stratifies the baseline data to two baselines, one for weekdays and one for weekend days. The mean and standard deviation of each baseline are calculated separately. This variation of the C-algorithms is called the W2 algorithm if two days immediately before the test day are excluded from the baseline. The W3 algorithm is similar to W2 but also uses two days of guardband and pools three test-statistics, in the spirit of C3. Experiments showed that stratifying baseline days into weekdays versus weekends increases the sensitivity of detection in the data set which has a strong day-of-week effect, but mostly decreases sensitivity in data sets without weekly patterns (Tokars et al., 2009b).

The short 7-day baseline interval of C2 produces volatile parameters, so that the triggering alert threshold might vary widely over a short period of time. To decrease the volatility of the algorithm, Tokars et al. (2009a) proposed to lengthen the baseline interval because a 7-day baseline interval does not provide sufficient data for an accurate and stable calculation of mean and standard deviation. They tested baselines intervals of 7, 14, and 28 days. The result was that the longer baselines

improved the sensitivity of the detection algorithm, by producing more accurate and stable standard deviation estimates, but had no effect on the accuracy of the expected value calculation.

The C2 algorithm also does not account for the population at risk, which may vary during a crisis situation. Tokars et al. (2009a) adjusted C2 to consider the rate of daily visits to a medical centre. In the rate-based C2 algorithm, the observed value at time t is replaced by:

$$Observed_t \times \frac{\sum_{i=1}^{L} syndromicVisits_i}{\sum_{i=1}^{L} totalVisits_i}, \tag{2.2}$$

where L is the length of the baseline interval, $syndromicVisits_t$ is the number of visits for a specific syndrome on day t, and $totalVisits_t$ is the number of total visits on day t. It means that for calculating the expected number of clinical visits for a given syndrome on day t, the number of visits for that syndrome is summed over 7 recent days (or a longer baseline); the total visits to the medical centre is also summed over the same period of time; the proportion of these two values is calculated and multiplied by the number of visits for that syndrome on day t (i.e. the observed value). Then, the variation of expected value from the observed value is evaluated, in number of standard deviations (Xing et al., 2011). The standard deviation is computed as:

$$standardDeviation_t = \frac{\sum_{i=1}^{L} |syndromicVisits_i - \frac{totalVisits_i \times \sum_{j=1}^{L} syndromicVisits_j}{\sum_{j=1}^{L} totalVisits_j}|}{L} \tag{2.3}$$

Although the number of total visits is not an ideal denominator for calculating rates, in general it is better than not having denominator (Tokars et al., 2009a). Accounting for the total visits of medical centre improved the accuracy of calculation of expected value and the sensitivity of the rate-based C2 was better, compared to simple count-based C2, at the same alert rate (Tokars et al., 2009a).

2.2.2 Non-adaptive Log-linear Regression Algorithm

The global non-adaptive regression model of surveillance data assumes that the chosen covariates, such as long-term trends, seasonal trends, and day-of-week, are sufficient to capture the systematic background behaviour of the time series. The expected value at t is computed as a linear combination of these covariates. If the observed value at time t is outside a certain range, the algorithm flags an outbreak event (Burkom et al., 2007).

This algorithm uses several years of historical data as the baseline interval, in order to extract the systematic patterns and estimate the algorithm parameters. Afterwards, parameters will not be

adjusted to changes, so the model is not adaptive to recent data.

In the non-adaptive log-linear regression algorithm proposed by Brillman et al. (2005), the original value at time t is transformed into log scale, because the time series values differ on a percentage scale rather than by fixed amounts. This model assumes that the difference between the expected value and the observed value has a Gaussian distribution. Ordinary least square is used for parameter estimation. The expected value at time t is estimated as:

$$\log(Expected_t) = [\sum_{i=1}^{7} c_i * dow_i(t)] + [c_8 + c_9 * t] + [c_{10} * \cos(2\pi t/365.25) + c_{11} * \sin(2\pi t/365.25)] \tag{2.4}$$

where t is the day of the year. The term $[\sum_{i=1}^{7} c_i * dow_i(t)]$ captures the day-of-week effect. The coefficients, $c_1 - c_7$ are trained from the baseline data and $\sum_{i=1}^{7} c_i = 0$. The indicator of the day of week, $dow_i(t)$ is 1 for the *i*-th day of week and zero for the others.

The term $[c_8 + c_9 * t]$ captures the long-term trends, like changes in the size of population, with the assumption that the long-term trend is a linear component over the surveillance time series. This term might mask gradually increasing outbreaks (Lombardo and Buckeridge, 2007).

The last term $[c_{10} * \cos(2\pi t/365.25) + c_{11} * \sin(2\pi t/365.25)]$ captures the seasonal trend, where 365.25 is the average number of days in a year and $2\pi/365.25$ gives a 1-year sinusoidal cyclic behaviour. The coefficients of each trend can be set to zero if the corresponding pattern is not observed in the data.

All parameters are estimated using a long baseline, and are not adapted to recent changes in data. This algorithm is suitable only if training data over several years, without large year-to-year changes, is available.

2.2.3 Adaptive Log-linear Regression Algorithm

A modified version of log-linear regression uses a shorter sliding baseline interval of 56 days (Burkom, 2003). Hence, the logarithm of the expected value at time t is computed similarly to the non-adaptive algorithm as:

$$\log(Expected_t) = [\sum_{i=1}^{7} c_i * dow_i(t)] + [c_8 + c_9 * t] + [c_{10} * holiday_t] \tag{2.5}$$

As in the non-adaptive regression model, the term $[\sum_{i=1}^{7} c_i * dow_i(t)]$ models the day-of-week effect. When a holiday occurs during the baseline interval, the value of the surveillance time

series (e.g., the number of clinical visits) in post-holiday day may increase after it. The term $[c_{10} * holiday_t]$ captures this effect. The holiday indicator $holiday_t$ is 1 if day t is a holiday. Note that this effect is not noticeable for a very long baseline, but could be significant for a 56-day baseline.

Comparing with non-adaptive regression algorithm, the sinusoidal term for capturing seasonal trends over a year was removed because the baseline is short. Instead, the term $[c_8 + c_9 * t]$ captures recent seasonal and long-term trends. A 2-day guardband interval is used to avoid the contamination of the baseline interval and test interval.

Lombardo and Buckeridge (2007) showed that the adaptive log-linear regression algorithm predicted more accurately than non-adaptive algorithm, because the model of time series in the adaptive algorithm reflects the recent behaviour of data.

Regression algorithms developed for one time series cannot be transferred easily to other time series. The performance comparison between C-algorithms and the regression algorithm will be discussed in section 2.2.4.

2.2.4 Adaptive Poisson Regression Algorithm

To adopt a broader perspective on adaptive log-linear regression algorithm discussed in section 2.2.3, we can see the computation of the expected value as a linear combination of the data values:

$$Expected_t = f([\sum_{i=1}^{n} c_i * x_i(t)] + \varepsilon) \tag{2.6}$$

where $x_i(t)$ are values based on the time series and ε is the error term. The class of models described by Equation (2.6) is called *generalized linear models (GLM)* (Nelder and Wedderburn, 1972). The GLMs are general category of models for a random variable from an exponential family distribution. This family is very rich and includes Gaussian, binomial, multinomial, Poisson, and other distributions. A GLM assumes that the input variable comes into the model via a linear combination $C^T x$ and the output is represented as a function of this linear combination $f(C^T x)$. The function *f* is known as an *activation function* in the machine learning literature, whereas its inverse, f^{-1}, is called a *link function* in the statistics literature. The activation function provides the relationship between the linear combination and the mean of the distribution. The choice of exponential family distribution and the choice of the activation function provide the specification of a GLM. The choice of exponential family distribution is constrained by the nature of the data. The constraints reflecting the conditional expectation of the model are imposed on the activation

function. For example, for a Bernoulli distribution the conditional expectation must lie between 0 and 1, so an activation function with the range of $(0, 1)$ should be selected.

From this perspective, the link function of adaptive log-linear regression is a logarithmic function of the surveillance data and the error term is assumed to have a Gaussian distribution.

As discussed in section 2.2.1, the assumption of a Poisson distribution is more realistic for surveillance time series than the Gaussian assumption. For this reason, the Poisson regression algorithm uses a generalized linear model for outbreak detection and assumes that the distribution of the error is Poisson. Poisson regression does not require constant variance and can be more robust than other generalized linear models. Xing et al. (2011) developed a Poisson regression algorithm with a baseline interval of 56 days. Empirical studies did not show any improvement by using longer baseline intervals. Also, this algorithm captures the seasonal trends in cycles of 14 days. A 2-day guardband was used to avoid the contamination of the baseline interval and test interval. The Poisson regression algorithm with logarithmic link function estimates the expected value at time t as:

$$\log(Expected_t) = c_0 + [c_1 * dow_{baseline}(t)] + [c_2 * 14day_{baseline}(t)] \tag{2.7}$$

where c_0 is a constant intercept, the term $[c_1 * dow_{baseline}]$ captures the day-of-week effect, and the term $[c_2 * 14day_{baseline}]$ represents the current seasonal behaviour. The indicator variable $14day_{baseline}$ has a fixed value for each day during the in-day interval. The Poisson regression algorithm is adaptive to recent changes in the data and the algorithm parameters and threshold for alerting are estimated based on a sliding window of 56 days.

The Poisson regression algorithm described above can be modified in order to consider the proportion of infected population. For this purpose, two variations of Poisson regression were developed by Xing et al. (2011). In the first modified Poisson regression, the number of total clinical visits during the most recent 56 days was presented as an additive covariate, so the expected value of time t is estimated as:

$$\log(Expected_t) = c_0 + [c_1 * dow_{baseline}(t)] + [c_2 * 14day_{baseline}(t)] + [c_3 * totalVisits_{baseline}] \tag{2.8}$$

where $totalVisits_{baseline}$ is the total number of clinical visits during 56 days of baseline and the day t.

In the second modified Poisson regression algorithm, the log of total clinical visits was used as an offset $\log(totalVisits_{baseline})$ in the computation of expected value at time t. This algorithm

uses the ratio of syndromic and total visits rather than modelling syndromic counts with total visits as a covariate, as in the former version.

Now, we briefly describe the threshold calculation method used by Xing et al. (2011). Generally, the surveillance time series are not time-independent and do not fit a single probability distribution. Therefore, using a single probability distribution function to derive an alerting threshold is not appropriate. Xing et al. (2011) used historical baseline data and the concept of *recurrence interval* to derive the threshold. The recurrence interval is the number of the days for which the expected number of threshold crossing events is 1, so the empirical daily probability of crossing the threshold is 1/N. Since an alert rate of 1 in almost 3 months is practical for public health surveillance, they chose a recurrence interval of 100 days. They calculated the z-score of the time series as:

$$z - score_t = \frac{Observed_t - Expected_t}{standrardDeviation_t} \tag{2.9}$$

Then, they selected the 99th percentile cutoff value of all z-scores from the time series as the alerting threshold. Thus, the daily alert rate for each time series is the same, 1% (Xing et al., 2011).

Xing et al. (2011) used real data from US CDC BioSense surveillance system of emergency department with artificially-added increases in syndrome counts for evaluating the performance of detection algorithms. The performance evaluation of the Poisson regression algorithm and its modifications showed that including the total number of visits improved the accuracy of prediction and the sensitivity of the detection algorithm, because considering the total number of clinical visits helps in modelling the unexpected and hard-to-model effects (e.g., unscheduled closure of clinics). Specifically, the second approach (i.e. using the log of total visits as an offset) outperformed the first modified version (Xing et al., 2011).

Furthermore, Xing et al. (2011) also compared the performance of outbreak detection of Poisson regression algorithms, linear regression, and modified versions of the C-algorithms (specifically, rate-based C2 and C2 with longer baseline intervals). Generally, the Poisson regression performed better than the linear regression algorithm, because Poisson regression does not require constant variance and is more robust than linear regression. The regression algorithms outperformed modified C-algorithms in both accuracy of prediction and sensitivity of outbreak detection. However, for sparse time series in which the number of clinical visits is low, regression algorithms can overfit the data. Finally, accounting for the total size of population always resulted in more accurate predictions and higher sensitivity of detection for all algorithms.

2.2.5 Generalized Likelihood Ratio Test

The Generalized Likelihood Ratio Test (GLRT) is based on the maximum of a function of the observed and expected number of cases within a fixed-width window. GLRT is used for retrospective evaluations of disease incidence over time, however Wallenstein and Naus (2004) simulated GLRT for prospective temporal surveillance. This method has the advantage of using a window with constant width over the test period without cancelling seasonal and day-of-week patterns. This method sounds an alarm if the following $testStatistic_t$ is larger than a threshold.

$$testStatistic_t(w) = Observed_t(w) \log(\frac{Observed_t(w)}{Expected_t(w)}) - (Observed_t(w) - Expected_t(w)) \tag{2.10}$$

where $Expected_t(w)$ is the expected number of cases in a window of w days ending at time t. $Expected_t(w)$ is calculated based on historical average seasonality. It is the average of observations in the same window of time over last year(s) of surveillance. Different approaches are used to implement this procedure, depending on the type of available baseline data. An alternative surveillance method, called P-scan, scans p-values. It is based on approximations thus can overstate or understate the probabilities. Wallenstein and Naus (2004) shows the P-scan and GLRT statistics with fixed-width windows are highly correlated and can be applied where quick detection of large peaks is vital. However, methods that monitor CUSUM statistics are superior for gradual increases.

2.3 Evaluation of Outbreak Detection Algorithms

In this section, first we examine the data which is used for evaluation of outbreak detection algorithms. Then, we review the formal definitions of performance measures of outbreak detection algorithms.

2.3.1 Evaluation Data

One important challenge of evaluating outbreak detection algorithms is obtaining surveillance data with sufficiently large number of outbreaks. Ideally, the evaluation data must contain outbreaks with known characteristics. Important outbreak characteristics are the magnitude of the outbreak signal, the shape of the signal, and the timing of the outbreak. The magnitude of the outbreak signal is the increment above the background signal due to the incidence of an outbreak. The shape of the signal expresses how this increase occurs; quickly over time or slowly over several days. The timing of the outbreak states the overall duration of the outbreak and the time between the onset of the outbreak and the peak day. Outbreak characteristics may influence the performance of outbreak detection algorithms (Buckeridge, 2007).

There are three common approaches to provide data for evaluation. Although using surveillance data collected from real outbreaks may initially seem to have the greatest validity for performance evaluation, it has several drawbacks. First, the characteristics of the outbreak signal (e.g the onset of the signal) cannot be derived without error. The reliance on an external review or imperfect standard to define those characteristics limit the accuracy and usefulness of real data. Moreover, there are few available time series containing verified outbreaks, even fewer associated with emerging infectious diseases or bio-terrorism (Lombardo and Buckeridge, 2007).

In the semisynthetic approach, simulated disease outbreaks are injected into real background data. The signal characteristics (e.g., the onset, magnitude, and shape of the signal) are known with certainty and random variations in outbreak signals and the onset time can be simulated. The main drawback of this approach is that the background data may contain undocumented outbreaks. Also, the simulated outbreaks are noise-free and the expected noise of real data is underestimated. This issue may result in overestimation of an algorithm's sensitivity and timeliness (Wagner and Wallstrom, 2006).

In the third approach of providing evaluation data, simulated outbreaks are injected into simulated background data. In this approach, the characteristics of outbreaks and background data are known with certainty and can be completely controlled to produce a wide variety of evaluation data, however, it is difficult to model background data.

While fully authentic outbreaks tend to be used in qualitative evaluation, the simulated outbreaks are useful in quantitative evaluation of outbreak detection algorithms. The semisyntethic and fully syntethic data are usually used for comparing a detection algorithm against other algorithms, test whether the algorithm is working, and check if its performance can be improved (Wagner and Wallstrom, 2006).

There are several methods for simulating the outbreaks. One is using mathematical functions (like step or linear) or probability functions (like exponential or lognormal) to generate different outbreak signals over time. Another method is using empirical density functions that are computed from observed real outbreaks. In the mechanistic simulation model, the mechanism underlying an outbreak, including disease, infection, and health care utilization are described. People are modelled in either network-based framework to extend communicable diseases or as independent stochastic processes in noncommunicable diseases. Mechanistic simulation models have many parameters and the parameter value selection influences on the result (Lombardo and Buckeridge, 2007).

The evaluation data that we used in this study is semisynthetic data generated by a mechanistic approach. The simulated surveillance data were generated using the Simulation Analysis Platform (SnAP) (Buckeridge et al., 2011) with a validated model for simulating waterborne outbreaks of cryptosporidiosis (Okhmatovskaia et al., 2010). In this simulation scenario, waterborne outbreaks were due to the failure of a water treatment plant. The simulation model includes components to represent water distribution, human mobility, exposure to drinking water, infection, disease progression, healthcare utilization, laboratory testing, and reporting to public health. We used this model with the SnAP to conduct many simulations of surveillance data that would result from a waterborne outbreak due to the failure of a water treatment plant in an urban area. This model creates a synthetic population from census data, and then uses 30 parameters to define the progression of individuals through the model. In the simulation scenarios for generating our data, two parameters were varied systematically: the duration of water contamination, which was varied over 6 values (72, 120, 168, 240, 360 and 480 hours), and the cryptosporidium concentration, which was varied over 3 levels ($10^{-6}, 10^{-5}, 10^{-4}$). The possible combinations of these values define 18 different scenarios. Each of these 18 scenarios was run 1000 times using Latin Hypercube Sampling to randomly select values from hyper-distributions for the other parameters in the model. The outbreak signals were superimposed on baseline data, which were daily counts of people visiting emergency departments in Montreal for gastro-intestinal diseases, over 6 years. The onset of the outbreak was selected randomly, relative to the baseline.

2.3.2 Evaluation Metrics

The goal of outbreak detection algorithms is rapid detection of outbreaks with few false alerts. To assess the accuracy of detection algorithms three quantities are measured: specificity, sensitivity, and timeliness. Specificity is the probability of no alarm when there is no outbreak:

$$specificity = P(alarm = 0|outbreak = 0) = \frac{n(alarm = 0, outbreak = 0)}{n(outbreak = 0)} \tag{2.11}$$

where $n(alarm = 0, outbreak = 0)$ is the number of non-outbreak days in which the algorithm does not raise an alarm and $n(outbreak = 0)$ is the number of non-outbreak days in an analysis interval.

Sensitivity is the probability of raising an alarm given that an outbreak really occurred:

$$sensitivity = P(alarm = 1|outbreak = 1) = \frac{n(alarm = 1, outbreak = 1)}{n(outbreak = 1)} \tag{2.12}$$

where $n(alarm = 1, outbreak = 1)$ is the number of alerts in outbreak days and $n(outbreak = 1)$ is the number of outbreak days. There exists two interpretation of the sensitivity of a detection algorithm. One might count the outbreak days in which the algorithm raises an alarm in the numerator of Equation (2.12) and the total number of outbreak days in the denominator. This is the probability of correctly classifying outbreak days and is called *sensitivity per day*. Alternatively, one can only measure the number of detected outbreaks in the numerator, no matter how many alarms have been raised within an outbreak, and count the number of outbreak intervals in denominator which differs from the number of outbreak days if an outbreak lasts for several days. In epidemiological context, the second interpretation of sensitivity, called *sensitivity per outbreak*, makes more sense because excessive alerts within a single outbreak are trivial. Table 2.1 illustrates a simple example of measuring the accuracy of detection algorithms.

The quantity of false alarm rate is the probability that an algorithm classifies a non outbreak day incorrectly and raises the alarm in the absence of outbreak:

$$falseAlarmRate = P(alarm = 1|outbreak = 0) = \frac{n(alarm = 1, outbreak = 0)}{n(outbreak = 0)} \qquad (2.13)$$

False alarm rate and specificity are sum to one and can be used in evaluations interchangeably (Wagner et al., 2006).

When sensitivity and specificity are calculated over a range of parameter settings for an algorithm, their values can be plotted to determine the *receiver operating characteristic (ROC)* curve. The ROC curve is a means of comparing sensitivity and specificity visually over a range of algorithm thresholds. In practice, the ROC curve is plotted based on sensitivity and false alarm rate (i.e. *1-specificity*) and points near to the left with higher sensitivity and specificity are more desirable. An ideal detection algorithm that classifies all outbreak and non-outbreak days correctly corresponds to point (0,1) in an ROC curve. A classifier that guesses at random corresponds to a diagonal ROC curve. However, when the sensitivity is calculated per outbreak and specificity is calculated per analysis interval, the random line is not necessarily diagonal because they are not

Table 2.1 Example of accuracy of detection algorithms

	outbreak	non-outbreak
alarm	7	5
no alarm	3	95
total	10	100
sensitivity=0.7; specificity=0.95		

calculated in the same scale (Lombardo and Buckeridge, 2007).

The area under an ROC curve (AUC) is a summary of measure of the classification accuracy. Since the AUC is a portion of the area of the unit square, its value is always between 0 and 1.0. Assume that a detection algorithm is a classifier of outbreak and non-outbreak days and raises an alarm if the test value of a questioned day is higher than the threshold. The AUC has a statistical interpretation: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen outbreak day higher than a randomly chosen non-outbreak day (Fawcett, 2004). An ideal classifier has an AUC of 1.0 and classifier that guesses at random (i.e. diagonal ROC curve) has an AUC of 0.5 (Wagner et al., 2006).

While sensitivity and specificity summarize the overall ability of an algorithm to detect outbreaks, they do not evaluate the timeliness of detection. Timeliness is the time between the start of an outbreak and the detection of the outbreak by an algorithm if it is detected. Timeliness is defined as the proportion of time saved by detection relative to the onset of an outbreak. If an outbreak is detected, timeliness is:

$$timeliness = 1 - \frac{t_{detection} - t_{onset}}{outbreakDuration} \tag{2.14}$$

where $outbreakDuration$ is the number of days for which outbreak cases occur. The $t_{detection}$ is the index of the day within the time series when the outbreak is detected and t_{onset} is the index of the day on which the outbreak starts. The proportion of delay is subtracted from 1, so higher values of the timeliness denote an earlier detection of the outbreak. Timeliness is 1 if the outbreak is detected in the first day of occurrence and 0 when the outbreak is not detected (Lombardo and Buckeridge, 2007). Surveillance systems require outbreak detection algorithms that avoid false alarms as much as possible while improving the timeliness of detection.

2.4 Evaluation of Learned Hypotheses

In this section, we shortly discuss how to estimate the quality of learned hypotheses. In machine learning, the learning process and the evaluation of algorithms are only based on a limited amount of data. However, it is important to understand the accuracy of algorithms on new data that the algorithms have not seen before. A learning algorithm is usually trained using a *training set*. If data is plentiful, then one can use some of the available data to train a range of hypotheses and then compare them on independent data, called a *validation set*, and select the hypothesis which has the best predictive performance. During the learning process, as the model is developed using a limited size of data and the complexity of the model grows, the prediction error on training set decreases, but the prediction error on validation set first decreases and then increases. This occurs because the model is being tuned to fit the training set that is not representative of the general distribution

of data. This is called the problem of *over-fitting* to data. So this predictive performance is not a good indicator for choosing the best learned hypothesis. It is necessary to keep aside a third *test set* which is completely untouched during training and validating a model and finally evaluate the performance of a selected model on it (Bishop, 2006).

However, in practice, the supply of data for training and testing is limited. One solution to estimate the predictive performance is to use *cross-validation*. In cross-validation, the n available instances of data are partitioned into k disjoint subsets of size n/k. Then the cross-validation procedure is run k times, each time using a different one of these subsets as the validation set. Other subsets are combined and used to train a set of models which will be evaluated on the subset that was left out (Bishop, 2006).

CHAPTER 3
BAYESIAN NETWORK FOR PREDICTING DETECTION PERFORMANCE

3.1 Background

The main role of automated syndromic surveillance systems is to screen large volumes of data, mainly data generated through healthcare utilization (e.g. counts of Emergency Department visits) in order to detect anomalies and issue alerts to draw the attention of human experts. Statistical algorithms are used to detect anomalies by comparing observed values to expected ones at regular intervals.

Many outbreak detection algorithms have been proposed for use in the syndromic surveillance. While it is clear that these algorithms perform differently when applied to different data sources or in various surveillance situations, there is little empirical evidence of their effectiveness under different conditions and it is not clear how public health practitioners should configure surveillance systems for efficient outbreak detection. The few studies evaluating detection performance are usually based on a single real or simulated data set, which is not publicly available, making evaluations difficult to generalize or replicate Jackson et al. (2007). Moreover, the performance of detection algorithms is influenced by many other factors, including the nature of the disease, characteristics of the outbreak signal (such as peak size and intensity), baseline data (such as weekly mean and standard deviation), and parameters of the detection method used (such as alerting threshold). Some researchers Watkins et al. (2006) argue that the lack of a standardized framework for assessment of outbreak detection methods and the diversity of factors that influence detection performance decrease the comparability of detection methods.

The objective of this research is to develop and evaluate a model for quantitatively characterizing the determinants of outbreak detection performance in terms of characteristics of outbreaks and detection methods and predicting the performance of detection methods. An earlier work Buckeridge et al. (2008b), showed it is possible to predict outbreak detection performance quantitatively with acceptable accuracy. They developed a prediction model based on logistic regression which assumes multiplicative relation between variables. While the logistic regression model could predict the detection performance of algorithms, it cannot model complex dependencies of variables and relations. In previous work, Izadi et al. (2009) addressed those limitations of logistic regression model by utilizing a Bayesian network model. They used data generated by a simple simulation to demonstrate the feasibility of developing a Bayesian network framework to analyze the detection performance and predict the ability of some of algorithms in detecting outbreaks. In this

work, we combine outbreak data generated by a realistic simulation model with real healthcare utilization data and then evaluate the performance of a wider range of statistical methods for detecting the outbreaks. The resulting dataset on outbreak detection performance is used to learn and evaluate a Bayesian network model structure and parameters for predicting detection performance. Indeed, the developed Bayesian network is used for predicting how well different outbreak detection methods will perform under different outbreak circumstances. We illustrate a variety of outbreak scenarios and utilize the Bayesian network inferences to find the most likely settings for detection methods and predict the detection performance in those scenarios. This work proposes a more generalized framework for performance evaluation of outbreak detection methods under a wide variety of outbreak circumstances. Figure 3.1 summarizes the workflow of this experiment.

3.2 Bayesian Networks

As discussed in Chapter 1, our first objective is to characterize the performance of detection algorithms based on outbreak types and characteristics. For this purpose, characteristics of outbreak and surveillance data are encoded as determinants of detecting outbreaks. We want to figure out how these determinants influence the performance of outbreak detection algorithms. In part of our work, we will use Bayesian networks to model graphically the relationships between determinants of outbreak detection and the sensitivity, specificity, and timeliness of detection algorithms. In this section, we briefly review the description of Bayesian networks (Pearl, 1988).

Probabilistic graphical models provide an effective approach to represent dependency relationships over a set of random variables based on probability theory and graph theory. The Bayesian network is a type of probabilistic graphical model. The structure of the Bayesian network is represented by a directed acyclic graph. In this graph, each node represents a random variable (or a group of random variables), and the edges (or links) express probabilistic relationships between random variables. If there is a link from node A to node B, then node A is the parent of node B. Each random variable can be discrete or continuous.

An example of a Bayesian network with three nodes is shown in Figure 3.2. In this example, the nodes C and O are the parents of the node D. Mapping to the problem of determinants of outbreak detection performance, C stands for classifier, O refers to outbreaks and D is the detection performance. Each node might be a group of random variables: C presents parameters of different outbreak detection classifiers, O is the collection of characteristics of outbreaks like signal size and signal magnitude, and D is decomposed to the sensitivity, specificity, and timeliness of detection. There is evidence that outbreak characteristics and classifier parameters are determinants of detection performance so the node O and C are the parents of node D in the corresponding Bayesian

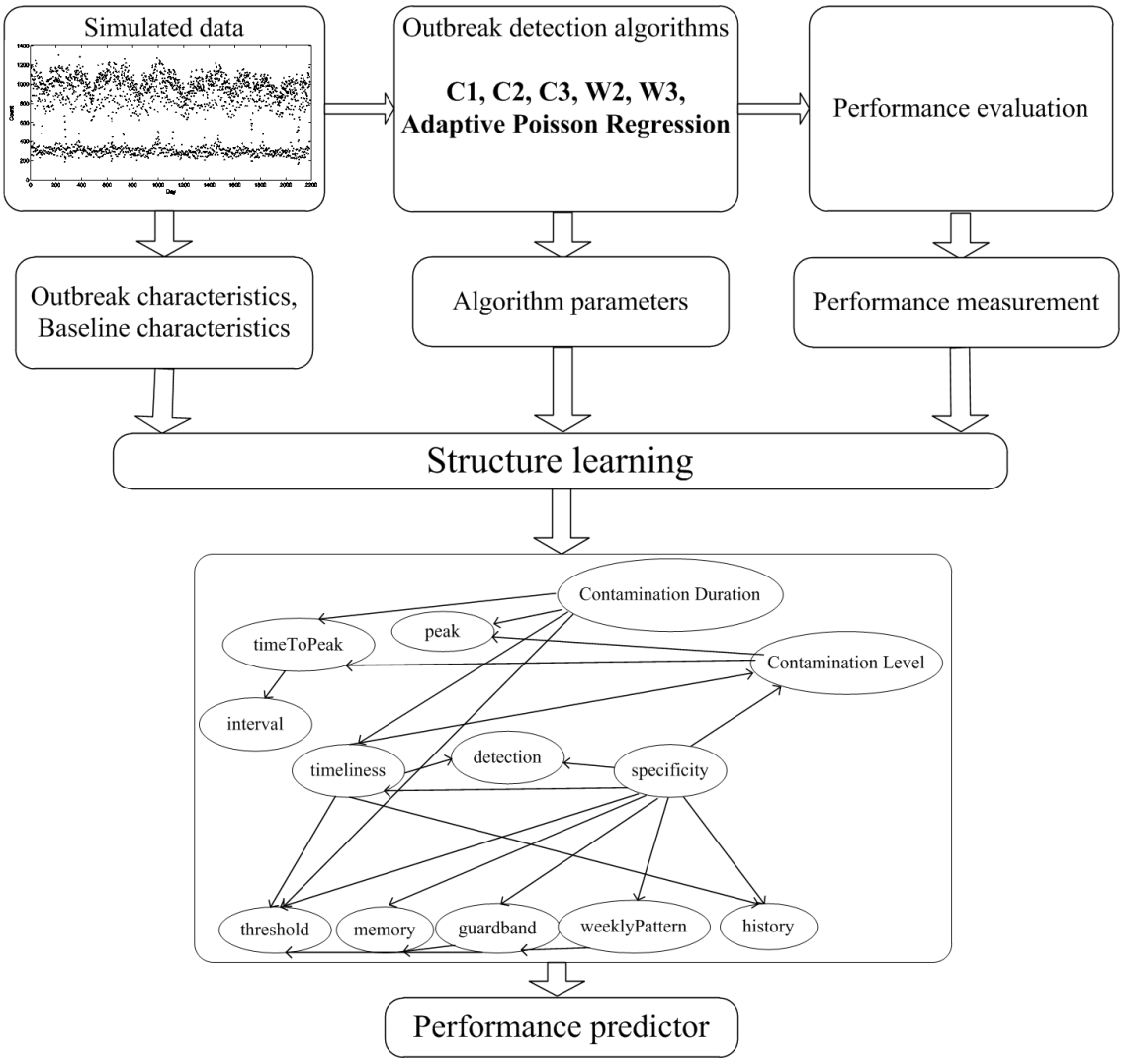

Figure 3.1 Workflow of learning bayesian networks to predict the detection performance

network.

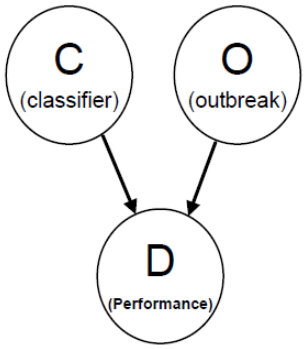

Figure 3.2 Graphical representation of a Bayesian network with three nodes

By definition, two random variable A and B are conditionally independent given random variable C (shown as $A \perp B|C$) if for all values of A, B, and C

$$P(A, B|C) = P(A|C)P(B|C) \tag{3.1}$$

Bayesian networks represent the joint distribution of variables in a factorized way. The joint distribution of n random variables $X_1, ..., X_n$ is expressed as:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa_{X_i}) \tag{3.2}$$

where Pa_{X_i} are the parents of node X_i. This equation gives a method to determine the joint distribution of random variables as a product of factors. In our example (figure 3.2) the chain rule for the joint distribution of C, O, and D is

$$P(C, O, D) = P(C)P(O)P(D|C, O) \tag{3.3}$$

One powerful aspect of a Bayesian network is that it does not impose restrictions on random variables (whether they are discrete or continuous): the independencies and chain rule are accurate for both cases. However, the presentation of the network depends on the random variables. Each variable is associated with its conditional probability distribution (CPD). The CPD of each variable captures its conditional probability, given its parents in the graph. A CPD of each discrete variable can be represented by a table which contains a row for each possible configuration of values of parents of that variable. The specification of variables will be different when a mixture of discrete and continuous variables appears in the network. For example, if the distribution of a continuous

random variable A is known (e.g., Gaussian distribution) and its parents are discrete variables, a specific Gaussian is fitted to A for each combination of values of its parents. Note that using different combinations of values of the parents is limited to the case of discrete-valued parents. Other types of representations are needed for continuous-valued parents.

We can use a Bayesian network to infer the probability distribution of a variable given the observed values of the other variables. In general inference, a Bayesian network can be used to compute the probability distribution for any subset of variables given the values or distributions for any subset of the remaining variables (Mitchell, 1997). We do not discuss the inference problem here; an extensive discussion can be found in Koller et al. (2007a).

In order to use Bayesian networks as the graphical model for representing data, the learning task is divided to two subtasks: learning the structure of the network (i.e., identifying the topology, the links, and directionality of the arrows) , and learning the network parameters (i.e., conditional probabilities) for a given network topology (Pearl, 1988). In learning the structure of a Bayesian network, a set of possible network structures and a scoring function are defined. The scoring function measures how well the model fits the observed data. Then heuristic search algorithms are used to find an optimal network structure however, the search space (i.e. the set of possible networks) consists of $2^{O(n^2)}$ for n variables and the problem is NP-hard (Koller et al., 2007b). One simple search procedure is the greedy hill-climbing search algorithm which starts from an arbitrary structure and chooses to move to its neighbours with the largest improvement in the score. This search algorithm continues until no modification improves the score. Greedy algorithms tend to get trapped in local maxima. However, global search strategies such as genetic algorithms and simulated annealing can produce better solutions at the cost of longer running times. In hybrid techniques of structural learning, researchers combined genetic algorithms with other techniques like expectation-maximization procedure or Tabu search Daly et al. (2011). Tabu search algorithm keeps a list of the most recently visited structures and avoid them i.e., improves the score for the structures that are not in the list (Koller et al., 2007b).

Naive Bayes and Tree Augmented Naive Bayes impose the Naive assumption on possible structures in order to learn the structure. Naive Bayes approach assumes that all variables are conditionally independent given the parent node. This classifier simplifies the structure of network by ignoring the correlations between variables (Duda et al., 1973). Some of other approaches are Maximum Spanning Tree (Lauritzen and Spiegelhalter, 1988) and Markov Blanket learning (Kindermann et al., 1980).

For learning the parameters of Bayesian network, several scenarios can be considered based on available structure and observed data. If the structure of the network is given in advance or it can be inferred from data, and the variables are completely observable and discrete, the CPD tables can be learned in a straightforward manner using maximum likelihood. But if only some of variables have been observed, the learning problem can be solved using Expectation-Maximization algorithm (Dempster et al., 1977) with the objective of maximizing the likelihood of the observed CPD table entries (Mitchell, 1997). Here, we describe the Expectation-Maximization algorithm based on the textbook Mitchell (1997).

Generally, the Expectation-Maximization (EM) algorithm can be applied where we want to estimate some set of parameters θ that describe an underlying probability distribution of data, given the observed portion of the full data. Suppose that a set of m instances has been drown independently: $X = \{x_1, ..., x_m\}$ denotes the observed data, $Z = \{z_1, ..., z_m\}$ denotes the unobserved data, and $Y = X \cup Z$ denotes the full data. The unobserved Z is a random variable whose probability distribution depends on unknown parameters θ and observed X.

The EM algorithm searches for the maximum likelihood of the full data Y. The distribution of Y is unknown because it is determined by θ. Suppose h is the current estimated values of the parameters θ and $h^{'}$ is the revised estimation of θ in each iteration of EM algorithm. The EM algorithm tries to maximize the expected value of probability distribution of full data which is determined by unknown θ by seeking for the $h^{'}$ that maximizes $E[\ln P(Y|h^{'})]$. The distribution governing Y is determined by the completely known values of X plus the distribution governing Z.

The EM algorithm begins with an arbitrary initial estimation h. Given the observed portion X and under the assumption that $\theta = h$, a function $Q(h^{'}|h)$ defines $E[\ln P(Y|h^{'})]$ as a function of $h^{'}$:

$$Q(h^{'}|h) = E[\ln P(Y|h^{'})|h, X] \tag{3.4}$$

The EM algorithm repeats the following two steps until convergence:

step 1: In *Expectation* step, $Q(h^{'}|h)$ is calculated using the current estimation h and the observed data X to estimate the probability distribution over Y:

$$Q(h^{'}|h) = E[\ln P(Y|h^{'})|h, X] \tag{3.5}$$

step 2: In *Maximization* step, the estimated h is replaced by the $h^{'}$ that maximizes Q:

$$h = \arg\max_{h'} Q(h^{'}|h) \tag{3.6}$$

The EM algorithm improves the likelihood of data over iterations and leads to parameter setting that maximizes (locally or globally) the likelihood (Mitchell, 1997).

3.3 Reduced Error Pruning Tree

Another learner that we will use for identifying the determinants of detection performance is the Reduced Error Pruning Tree (REP Tree). The REP Tree is a fast simple decision tree learner for classification and regression problems.

A decision tree is a sequence of binary selections in the data. After a decision tree is generated, it suffers the overfitting to the training data. Thus, the pruning process is applied to the tree to increase the accuracy. A pruning process cuts down sub-trees repeatedly until the error is reduced as small as possible. There are some variants in the pruning of the decision tree. The REP Tree uses a fast reduced-error pruning algorithm with backfitting. In the REP Tree, a sub-tree is pruned only if it dose not contain a sub-tree with a lower error than itself. The REP Tree uses the information gain heuristic to choose an attribute and a binary split on numeric attributes (Park et al., 2006). We will use a REP Tree for predicting the performance of detection algorithms based on the outbreak characteristics and the algorithm parameters.

3.4 Predicting Detection Performance using REP Trees

In the preliminary experiments, we investigated if the performance of outbreak detection algorithms is predictable given the algorithm settings and outbreak characteristics. First we developed a model based on Reduced Error Pruning Tree (REP Tree) to predict the detection performance as a regression problem i.e. how much are the sensitivity and specificity of an outbreak detection algorithm given the algorithm parameters.

In order to generate data for training the predictors of the performance, first we ran C1, C2, C3, W2, and W3 detection algorithms on 18,000 simulated time series and measured the sensitivity and specificity of detection. Table 3.1 shows the data features set. Each instance of this data set is the result of performance evaluation of an outbreak detection algorithm with a specific setting. For example, as a part of the training data, the highlighted instance in Figure 3.3 shows the performance of the C3 (guardband=2, memory=2, weekly pattern=0, threshold=3) when the contamination scale

of the outbreak was 0.000001 and contamination duration was 72. The training data contained 540,000 instances and we used 5-fold cross-validation to avoid overfitting.

We developed REP Tree models for predicting the sensitivity of the C and W algorithm using Weka machine learning software (Hall et al., 2009). In the first REP Tree model, we used algorithm parameters and outbreak characteristics to predict the sensitivity of C and W algorithms. The developed REP Tree is shown in Figure 3.4. In the second REP Tree, we assumed that the outbreak characteristics were not available and we only used algorithm parameters (i.e., alerting threshold, memory, guardband, and weekly pattern) to predict the sensitivity of C and W algorithms. The accuracy of two developed REP Trees are summarized in Table 3.2.

Here we define four measures of error for evaluating the accuracy of a predictors. Assuming a data set with N instances, the output values y_i, and the predicted values h_i ($1 \leq i \leq N$),

– *Mean Absolute Error (MAE)* is the average of absolute values of the differences between the output value and the corresponding predicted value:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - h_i| \tag{3.7}$$

– The *Root Mean Squared Error (RMSE)* is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - h_i)^2} \tag{3.8}$$

This error is always equal or greater than the MAE.

– The *Relative Mean Absolute Error* indicated how good a predicted hypothesis is relative to the

guardband	memory	weekly pattern	threshold	contamination level	duration	specificity	timeliness	sensitivity
0	0	0	0	1.00E-06	72	0.368	0.952	1
2	2	0	2	1.00E-06	72	0.306	1	0.667
2	2	0	3	1.00E-06	72	0.675	1	0.333
2	2	0	4	1.00E-05	240	0.958	0	0
2	2	0	5	1.00E-04	360	0.987	0	0
2	0	0	0	1.00E-06	480	0.37	0.929	0.5
2	0	0	1	1.00E-05	480	0.909	0.929	0.5
2	0	1	0	1.00E-06	168	0.393	0.958	1
2	0	1	1	1.00E-04	72	0.802	0.25	0.5

Figure 3.3 Part of algorithms performance data used for training the REP tree models

Table 3.1 Values of data features used to train and test REP Tree models

Data feature type	Data feature	Description	Value
Algorithm parameters	Memory	Number of recent observations include in the calculation of the expected value	0, 2
	Guardband	Gap between the sliding window and the test day	0, 2
	Weekly pattern	Binary value that indicates whether the algorithm considers the weekly pattern or not	0, 1
	Threshold	The alerting threshold	0, 1, 2, 3, 4, 5
Outbreak characteristics	Contamination level	-	10^{-6}, 10^{-5}, 10^{-4}
	Contamination duration	-	72, 120, 168, 240, 360, 480
Detection performance metrics	Detection	0,1	
	Specificity	[0,1]	
	Timeliness	[0,1]	

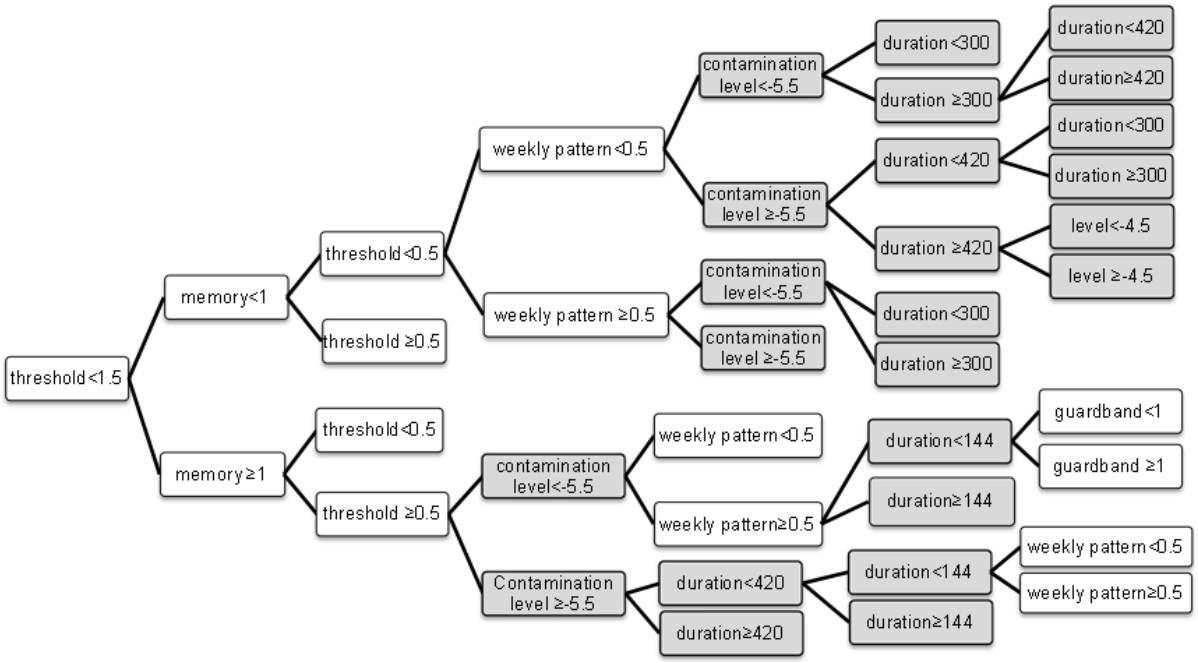

Figure 3.4 REP Tree for predicting sensitivity of detection based on algorithm parameters (White leaves) and outbreak characteristics (Gray leaves)

output value:

$$RelativeMAE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - h_i}{y_i}\right| \tag{3.9}$$

– The *Root Relative Mean Squared Error* indicates the goodness of prediction relative to the size of output:

$$RelativeRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{y_i - h_i}{y_i})^2} \tag{3.10}$$

According to Table 3.2, RMSE of the first model is 0.2344 while it is 0.275 for the second model so that, the accuracy of prediction model based on algorithm parameters and outbreak characteristics is higher than the model based on outbreak characteristics. This result shows that the C and W algorithm configuration influences the sensitivity of outbreak detection. Given the C-algorithm parameter settings, we can predict outbreak detection performance quantitatively. Also, in addition to algorithm settings, if some information about outbreak characteristics are provided, the model can predict the performance of detection algorithms more accurately.

We also developed two REP Tree models for predicting the specificity of C and W algorithms. The first model was given algorithm parameters and outbreak characteristics while the second model only knows algorithm parameters. The accuracy of two models were summarized in Table 3.3. This results show the accuracy of the two models (column 1 and 2) are the same. As Figure 3.5 shows the structure of the developed REP Tree, the outbreak characteristics were not contributed in building the REP Tree model for predicting the specificity. That might be the result of low variance of the specificity in the data set. In other words, regardless of outbreak scenarios, the algorithm parameters were enough informative to accurately predict the specificity.

In this chapter, we focused on building models to figure out first how determinants of performance and performance metrics interact with each other and second, how the performance of detection algorithms varies at different scenarios of outbreaks. Answering these questions will help

Table 3.2 Accuracy of predicting sensitivity using REP Tree models

	Data features	
	Algorithm parameters, Outbreak characteristics	Outbreak characteristics
Mean absolute error	0.1643	0.2071
Root mean squared error	0.2344	0.275
Relative absolute error	44%	55.46%
Root relative squared error	56.03%	65.72%

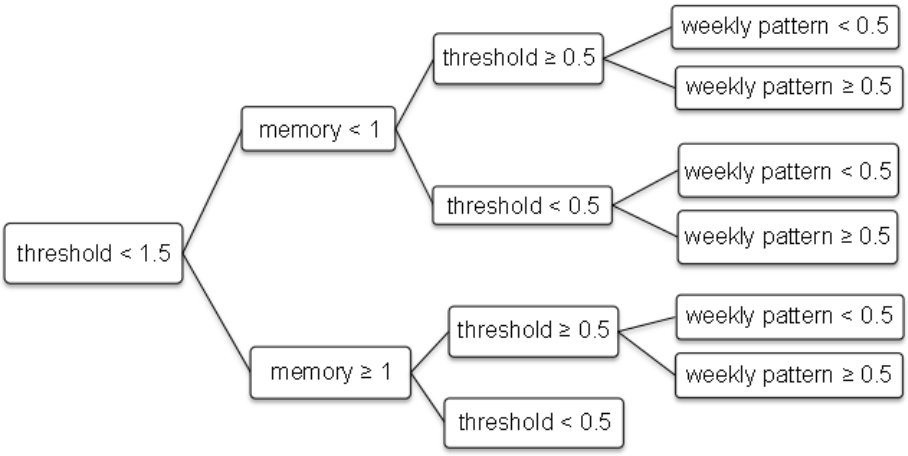

Figure 3.5 REP Tree for predicting specificity of detection based on algorithm parameters

Table 3.3 Accuracy of predicting specificity using REP Tree models

	Data features	
	Algorithm parameters, Outbreak characteristics	Outbreak characteristics
Mean absolute error	0.0035	0.0035
Root mean squared error	0.006	0.006
Relative absolute error	1.09%	1.09%
Root relative squared error	1.66%	1.66%

the practitioners in effectively adjusting detection algorithms based on different circumstances. The main finding of these preliminary experiments is that the algorithm parameters and outbreak characteristics are the determinants of outbreak detection performance. Given the determinants of performance, we can quantitatively estimate how well detection algorithms can detect outbreaks. However, REP Tree can predict one single performance metric (either sensitivity or specificity) as the target variable and the interaction between performance metrics can not be modelled by REP Trees. This will be addressed in section 3.5.

3.5 Predicting Detection Performance using Bayesian Networks

In this section, we evaluated the performance of a wider range of statistical methods for detecting the outbreaks. The resulting data on outbreak detection performance was used to learn and evaluate a Bayesian network model structure and parameters for predicting detection performance. Indeed, the developed Bayesian network is used for predicting how well different outbreak detection methods will perform under different circumstances. We illustrate a variety of outbreak scenarios and utilize the Bayesian network inferences to find the most likely settings for detection methods and predict the detection performance in those scenarios.

In the following experiment, we considered a set of widely used detection algorithms: C1, C2, C3, W2, W3 and Adaptive Poisson Regression in two modes of with and without guardband. We assembled a data set of the evaluation results of these algorithms together with their underlying parameters, and the characteristics of the simulated surveillance data explained above. Table 3.4 presents the features of this data.

This table shows that there are four types of features describing our data set: parameters of a detection algorithm (memory, guardband, weekly pattern, threshold, and history), characteristics of the GI baseline data (mean and standard deviation of the number of ED over recent seven days), characteristics of the outbreaks imposed to the baseline data (peak size, time to peak, interval of outbreak days, contamination level and duration of a contamination), and metrics used to measure the performance of a detection algorithm (detection, specificity, and timeliness). *Detection* is a binary variable which indicates whether each outbreak is detected or not. It can be seen as *sensitivity per outbreak*. It should be noted that the variables *detection* and *timeliness* are measured as described in 2.3 and should not be mixed with other definitions of algorithm detection sensitivity and timeliness. Each instance in the benchmarking data set consists of the result of evaluating an outbreak detection algorithm in a given surveillance situation. This dataset contained the total of 72,000 instances.

Table 3.4 Values of data features in training data

Data feature type	Data feature	Description	Value
Algorithm parameters	Memory	Number of recent observations include in the calculation of the expected value	0, 2
	Guardband	Gap between the sliding window and the test day	0, 2
	Weekly pattern	Binary value that indicates whether the algorithm considers the weekly pattern or not	0, 1
	Threshold	The alerting threshold	0, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 5
	Sliding window (history)	Size of the window used for the calculation of the expected value	7, 56
Baseline characteristics	Mean of 7 recent days		[400, 932]
	Standard deviation of 7 recent days		[159, 417]
Outbreak characteristics	Peak size	Number of additional counts of outbreak signal above the baseline	[3, 7845]
	Time to peak	Number of days from the onset of the signal to the peak day	[2, 26]
	Outbreak interval	Length of outbreak signal	[4, 52]
	Contamination level	-	10^{-6}, 10^{-5}, 10^{-4}
	Contamination duration	-	72, 120, 168, 240, 360, 480
Detection performance metrics	Detection	Whether or not the outbreak is detected	0,1
	Specificity	Probability of no alert when there is no outbreak	[0,1]
	Timeliness	Proportion of saved time to the outbreak duration	[0,1]

The binary variable *weekly pattern* was used to as a proxy to show the algorithm used, with zero indicating C1, C2 and C3, and 1 indicating W2, W3, and adaptive Poisson. The variable *sliding window (history)* shows the size of the window used to calculate the expected value of ED visits count in the ED visit time series. Its value is 7 days for C and W algorithms and 56 days for adaptive Poisson. The baseline characteristics are statistical description of the ED visit time series without outbreaks. In the data pre-processing step, continuous variables (e.g. *peak size*) were discretized using k-means for the ease of use in our Bayesian network model.

We used Bayesian networks (BNs) to model the relationships between detection performance, algorithm parameters, and outbreak characteristics. BNs can graphically represent these relationships Pearl (1988), and provide a tool for making a variety of inferences in the form of what-if analysis. The network graph structure and its parameters were learned from data using an optimization-based search method that tries to maximize a likelihood function over possible network configurations. We experimented constructing a Bayesian network using the algorithm benchmarking data and several structure learning methods, including Naive Bayes, Tree Augmented Naive Bayes, Maximum Spanning Tree, Markov Blanket learning, and Tabu search. Netica software package version 5.08 (Norsys, 1995) was used to learn the model parameters and to perform experiments.

The first BN that we developed determines the effect of the algorithm parameters and baseline and outbreak characteristics on the sensitivity and timeliness of outbreak detection. The BN structure was learned using Tree Augmented Naive Bayes. Figure 3.6 shows the structure of the learned BN for predicting the sensitivity. The accuracy of prediction was evaluated while the value of timeliness was missing because, at the time of detection, the timeliness is not observed. The accuracy of prediction of sensitivity by this network was 91.45% when the threshold of classification is 0.5. Figure 3.7 shows the ROC curve of the accuracy of prediction when the classification threshold changes between 0 and 1. The Area Under the ROC, another measure of classification accuracy, was 0.97.

Exploring the structure of Bayesian network is a useful way to find the determinants of detection performance and their influence. In order to identify the most influential variables in predicting of ability of detection, we examined the mutual information between variable detection and other variables of the network (Table 3.5). For two discrete random variables X and Y, the mutual information is defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) \tag{3.11}$$

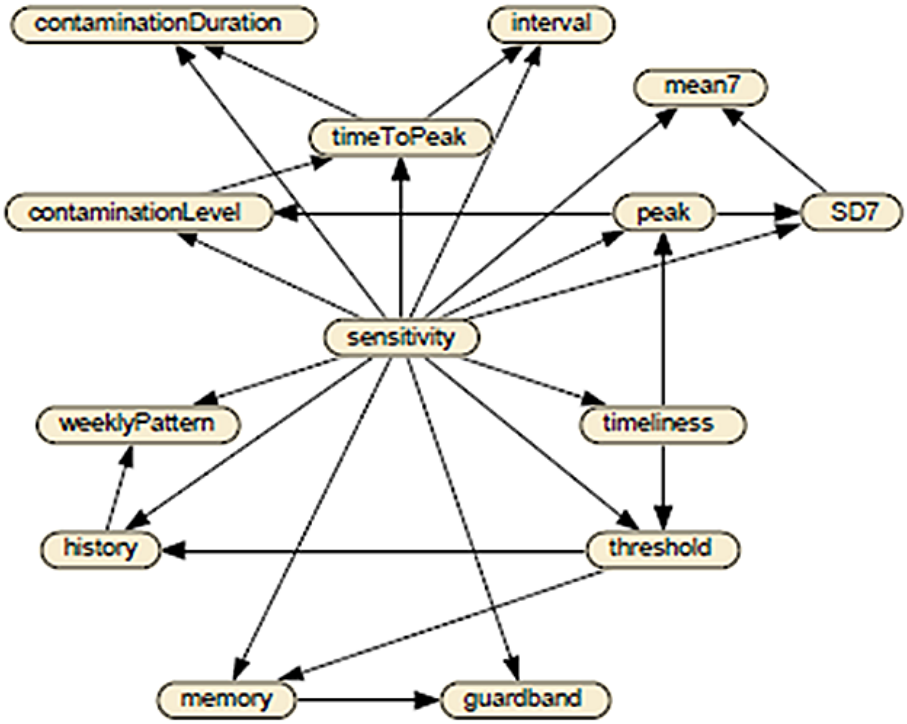

Figure 3.6 The structure of BN for predicting the sensitivity based on timeliness and all variables

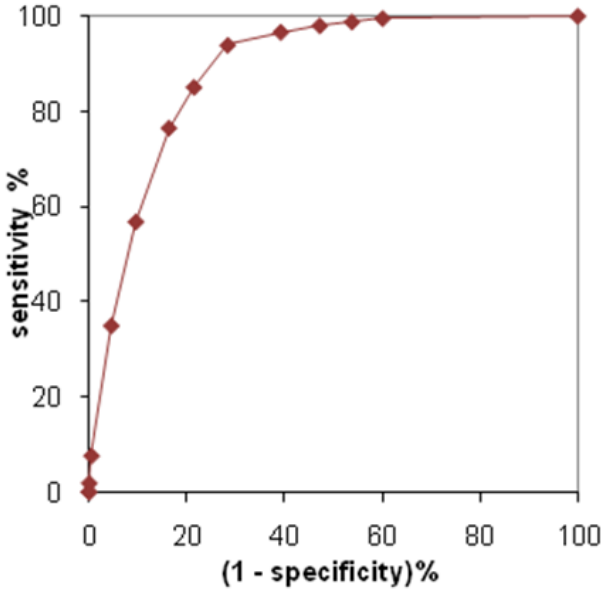

Figure 3.7 ROC curve to show the accuracy of developed BN in predicting the sensitivity

Table 3.5 Mutual information of the sensitivity and variables

Variable	threshold	peak	contamination level	weekly pattern	outbreak interval	time to peak	memory	mean 7 days	guardband	SD 7 days	historical window	contamination duration
%Mutual information	23.6	14.9	14.8	6.98	1.99	1.99	1.57	0.75	0.69	0.5	0.49	0.47

The mutual information measures the degree to which knowing one of the variables reduces uncertainty about the other, so the higher the mutual information, the stronger the relationship between the two variables. This measure is an indication of strength of the arcs of a Bayesian network.

Among algorithm parameters, the alerting threshold and the ability of the algorithm to account for a weekly pattern in the data had the highest mutual information with detection performance. In other words, they were the most informative variables for predicting the performance. Among outbreak characteristics, the contamination level and the peak size were the most influential parameters in predicting the ability of detection.

We also utilized the developed Bayesian network to predict the timeliness of detection of outbreaks. Note that the timeliness of detection is a performance metric which summarizes how rapid an outbreak can be detected by detection algorithms. We discretized the normalized values of timeliness in four levels. The error rate in prediction of timeliness was 24.76%. Examining the mutual information between timeliness and other variables of the network, the most informative variables in predicting the timeliness of detection were the alerting threshold, the contamination level and the peak size of the outbreak.

We examined the *belief bars* of variables when the specificity of detection is greater than 0.85 since it corresponds to an acceptable false alarm rate (1 false alert every week). Belief bars show the distribution of a variable. Each line of this graph shows a state of a variable along with a number expressing the belief (probability) of that state as a percentage. The variation of parameters in this range of specificity is still large. So we considered three classes of specificity in this range. The belief bars of the network are shown in figure 3.8 when the specificity is between 0.85 and 0.9. Notice that horizontal bars show the probabilities of states of each variable. The mean value $\pm$ standard deviation is shown below the belief bars. The red background highlights the algorithm

parameters when the sensitivity is 1. The figure shows that the higher detection rate (i.e. binary sensitivity) at an acceptable level of specificity can be obtained by Adaptive Poisson Regression with the threshold of 0.25 and without using guardband. It is also shown that the algorithms are more sensitive to the larger amount of contamination as expected.

In figure 3.9 and 3.10, the range of specificity is [0.9, 0.95) and [0.95, 1] respectively. The belief bars show that the sensitivity of detection is lower at the higher levels of specificity (pointed out by red arrows). But the timeliness doesn't significantly change in these three levels of specificity.

In order to examine how algorithm parameters should be configured to maximize the sensitivity and timeliness while keeping the false alarm rate at an acceptable level, we fixed the sensitivity and timeliness at the highest level when the specificity is greater than 0.95 (Figure 3.11). The configuration of algorithm parameters is highlighted by the red curve when the sensitivity is 1 and the timeliness is at the highest level. The best detection performance is the result of taking weekly patterns of time series into account and using 2 days of guardband without memory; this is the best setting regardless of using W algorithm or Adaptive Poisson. Also, as expected, the detection performance is higher when the contamination level is higher (pointed out by red arrows).

We learned the second network presented in Figure 3.12 in which we hid the outbreak characteristics and used the model to predict sensitivity. This model is of interest because, in practice, we can only configure algorithm parameters and set the information for baseline characteristics. This model was evaluated for prediction of sensitivity without knowing timeliness.

Figure 3.13 shows the ROC curve of the accuracy of prediction when the classification threshold changes between 0 and 1. The accuracy of this prediction was 80.0% when the threshold of classification is 0.5. The AUC was 0.9. The performance of the network was worse than the network that included all the variables, as expected. So it is concluded that outbreak and baseline characteristics are some of determinants of detection performance and considering these determinants will improve the performance of predictions.

We set the sensitivity to 1 and the timeliness to the highest level when the specificity is greater than 0.95 in Figure 3.14. Other nodes show the configuration of algorithm parameters. The best detection performance is the result of taking weekly patterns of time series into account and using 2 days of guardband without memory. This configuration is close to the W2 and Poisson regression algorithm with alerting threshold of 0 or 1, and is consistent with the result of the first network.

In summary, we quantified the dependence of detection performance on the type of outbreak, baseline data, and algorithm configurations. We assessed the performance of six different outbreak

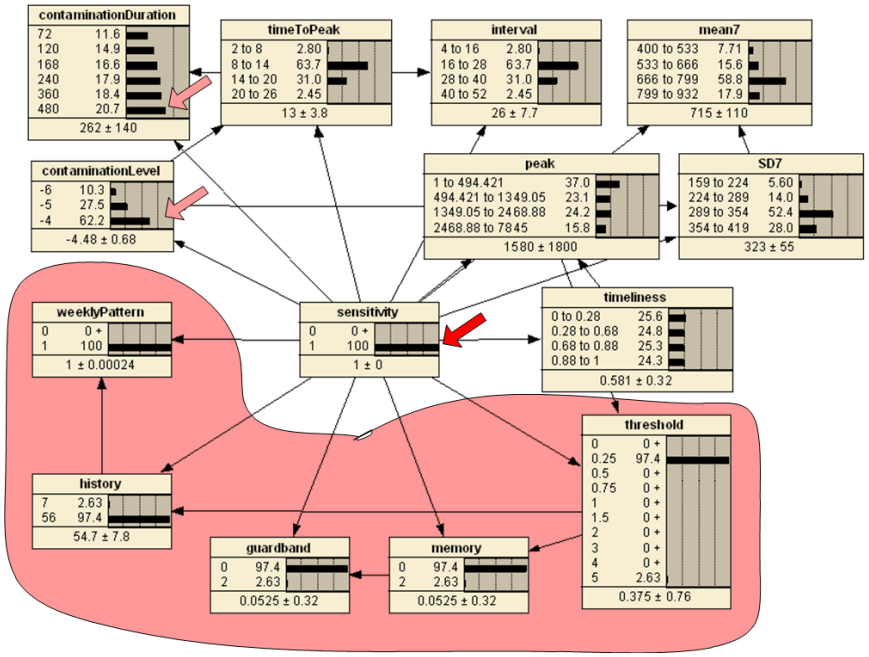

Figure 3.8 The belief bars of BN for the sensitivity, timeliness and other variables assuming the specificity of [0.85, 0.9). The background highlights the configuration of algorithm parameters and the arrows show the contamination scenario when the sensitivity is 1.

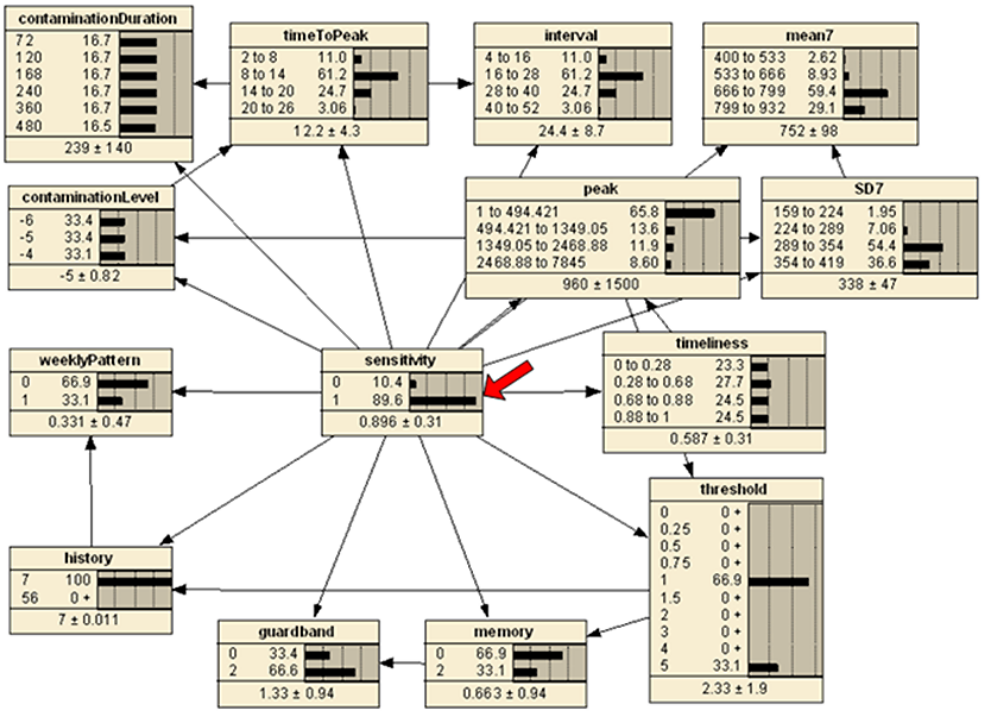

Figure 3.9 The belief bars of BN for the sensitivity, timeliness and other variables assuming the specificity of [0.9, 0.95)

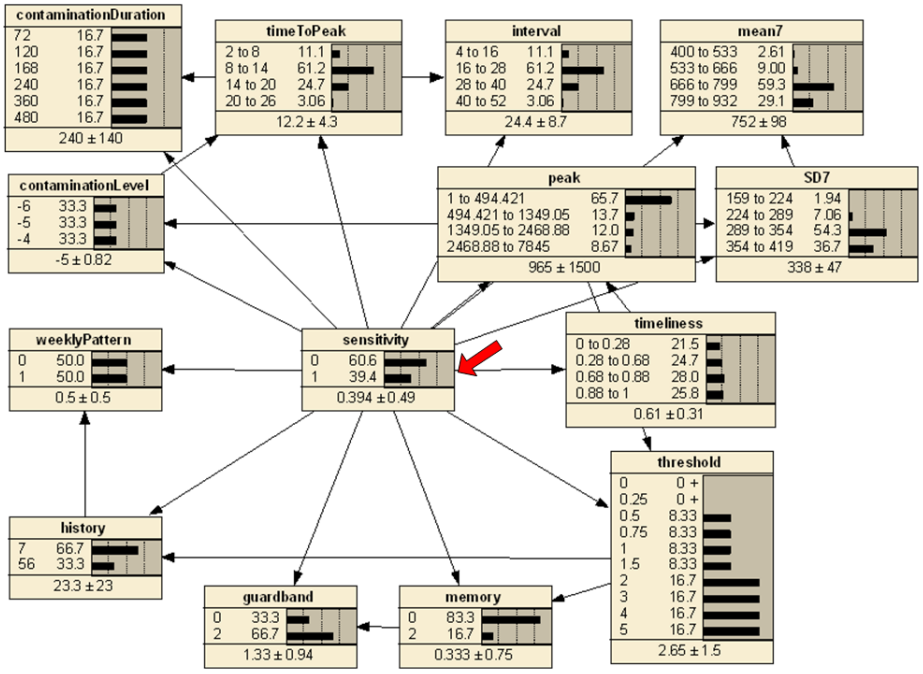

Figure 3.10 The belief bars of BN for the sensitivity, timeliness and other variables assuming the specificity of [0.95, 1]

detection algorithms using simulated and real surveillance data in eighteen scenarios and trained Bayesian networks to model the relationships between all surveillance attributes and the detection performance. Under the outbreak scenarios considered, algorithms that used a 2-day guardband and accounted for day-of-the-week variation in visits were predicted to have the best detection performance when operating at high specificity.

The experimental results quantify the effect of outbreak characteristics and algorithm configuration on the performance of detection algorithms. As expected, the most informative determinants of detection performance were the alerting threshold and the peak size of the outbreak, but our model also quantified the contribution to detection performance of algorithm features such as accounting for day-of-week and maintaining a guardband. Even when we developed a Bayesian network model where outbreak characteristics were unknown a priori, the model was able to predict detection performance with high accuracy (AUC = 0.9). Our modelling approach provides an important tool for quantitative evaluation of biosurveillance systems as new data sources or new methods are introduced in this field.

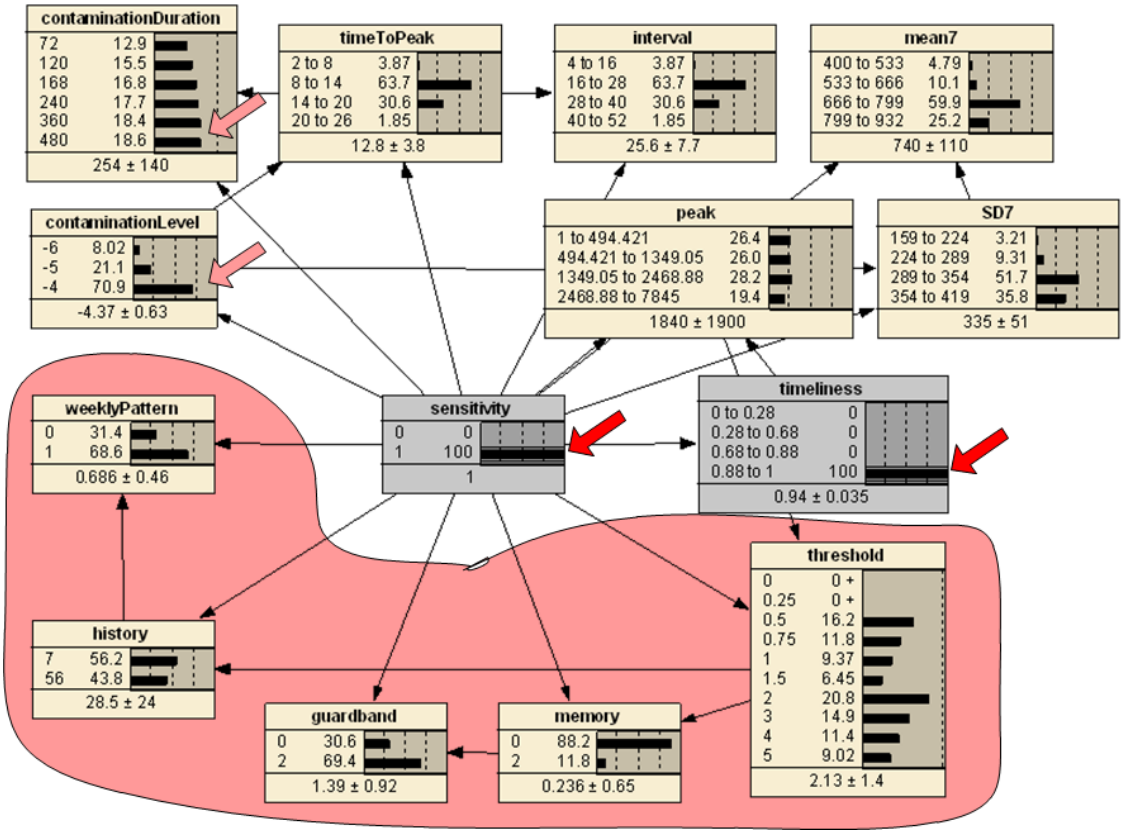

Figure 3.11 The belief bars of BN assuming the sensitivity of 1, specificity of [0.95, 1], and timeliness of [0.88, 1]. The red background highlights the configuration of algorithm parameters

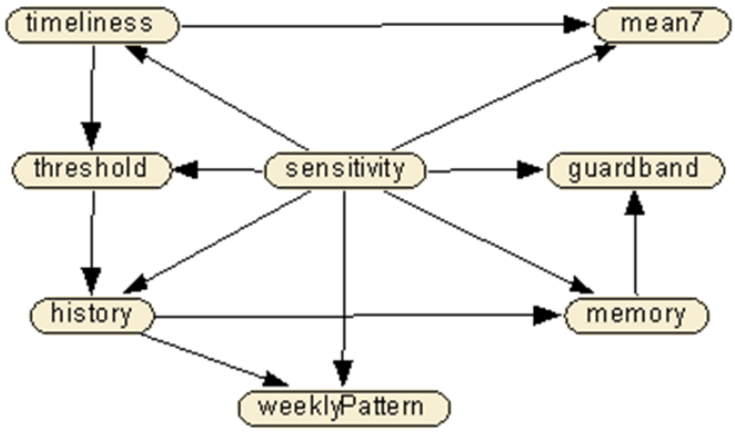

Figure 3.12 The structure of BN for predicting the sensitivity based on observable variables

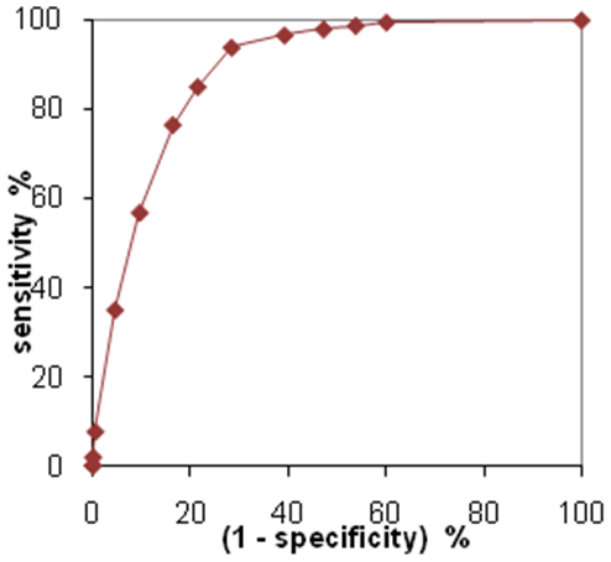

Figure 3.13 ROC curve of accuracy of the second developed BN in predicting the sensitivity

3.6 What-if scenarios

As we mentioned earlier, our model can be used to figure out which outbreak detection method is more appropriate compared to other methods given different outbreak scenarios, and how well that method can detect the outbreak. We examined different outbreak scenarios in Table 3.6. The first column of this table shows the variables that we set to a specific value or range. The probability of values for other variables was inferred from Bayesian network model and represented in the rest of the columns. In the inference, given the values of some nodes in the network, one can find the most probable configuration of the values for the rest of variables.

In the first scenario of outbreak (first row of Table 3.6), we set the specificity of detection to the values greater than 0.92 which corresponds to an acceptable false alarm rate for practitioners (i.e. less than one false alarm every 10 days). In this scenario, we assumed the outbreak was the consequence of high contamination in water (contamination level was 10^{-4} and contamination duration was 480). The most probable algorithm settings were inferred from Bayesian network and summarized in the next column. Following those settings, outbreaks with high contamination can be detected with the probability of 83.9% and the timeliness will be greater than 0.68 with the probability of 73.9%. As these most probable algorithm settings belong to W2 and C2 algorithms, it is concluded that among outbreak detection algorithms, C2 and W2 are more sensitive to outbreaks with high contamination while the specificity is greater than 0.92.

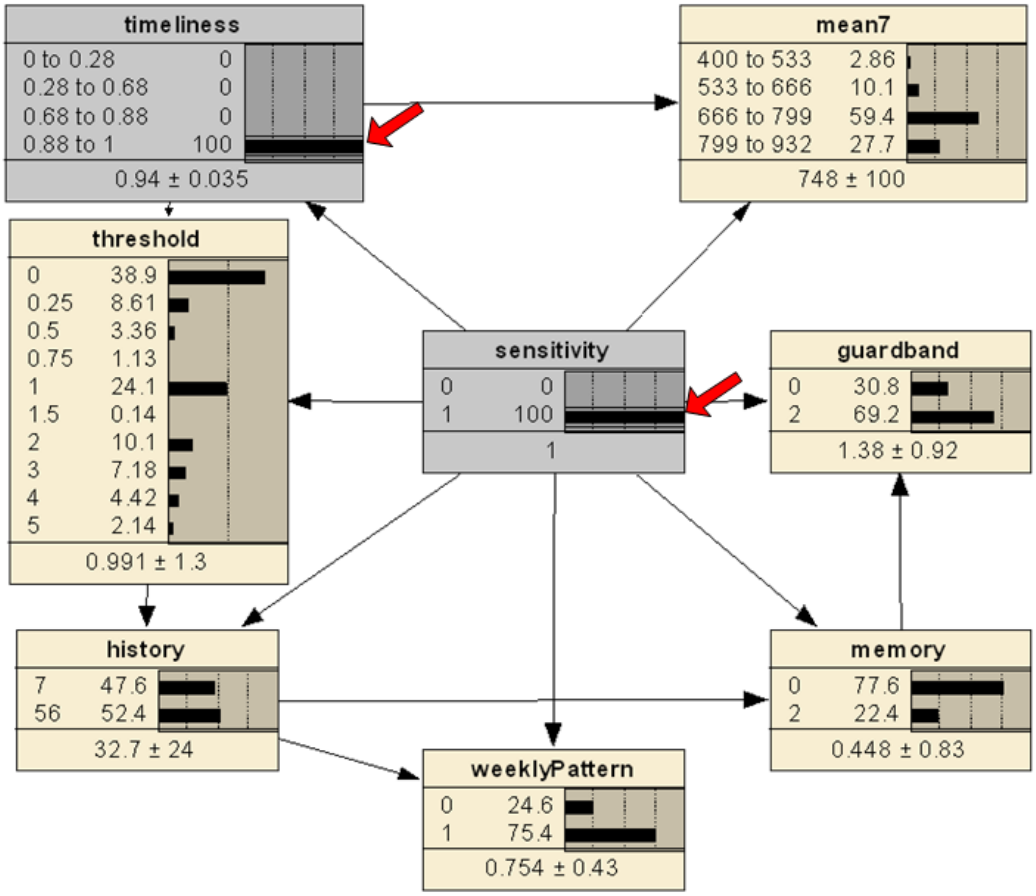

Figure 3.14 The belief bars of BN with observable variables at the sensitivity of 1, timeliness of [0.88, 1], and specificity of [0.95, 1]

Table 3.6 Predictions of Bayesian network for different outbreak scenarios

	Assumption	Inferred by Bayesian network	
Row	What-if scenario	Algorithm settings	Performance
1	specificity > 0.92 High contamination of outbreak: Contamination level = 10^{-4} Contamination duration = 480	P(guardband=2) = 65.3% P(memory=0) = 85.3% P(history=7) = 67.3% P(weekly=1) = 52.2%	P(detection) = 83.9% P(timeliness > 0.68) = 73.9%
2	specificity > 0.92 Small outbreak with early peak: $1 <$ Time to peak < 9 Outbreak interval < 18	P(guardband=2)= 63.6% P(memory=0) = 83.6% P(history=7) = 68.1% P(weekly=0) = 53.2% P(threshold$>$3.75) = 33.1%	P(detection) = 25.2% P(timeliness < 0.28) = 76.4%
3	$0.66 <$ specificity ≤ 0.92 Small outbreak with early peak: $1 <$ Time to peak < 9 Outbreak interval < 18	P(guardband=2)= 83.8% P(memory=0) = 67.2% P(history=7)= 67.6% P(weekly=1) = 81.9% P(threshold$<$1.2)= 65.6%	P(detection) = 89.5% P(timeliness > 0.88) = 31%
4	specificity > 0.92 Sparse baseline: $400 <$ Baseline mean < 706 $150 <$ Standard deviation < 320	P(guardband=2)= 64.2% P(memory=0) = 84.8% P(history=7)= 66.6% P(weekly=1) = 50.5%	P(detection) = 49.3% P(timeliness > 0.68) = 37.08%
5	specificity > 0.92 detection = 1 timeliness > 0.88	P(guardband=2)= 66.5% P(memory=0) = 92.8% P(history=7)=51.1% P(weekly=1) = 76.6% P(threshold$<$1.2)= 67.8%	

In the second scenario, we focus on small outbreaks that lead to early peak in the signal, i.e., the peak happens at most 8 days after the beginning of the outbreak. We were looking for the detection methods with the specificity greater than 0.92. The most probable setting of detection methods was obtained by analyzing inference of the Bayesian network and showed that C2 and W2 can detect small outbreaks with the probability of 25.2%. Their most probable timeliness is below 0.28.

In the third scenario, we considered the same scenario of outbreak, small outbreaks with early peak, though we set the specificity at a lower range ($0.66 <$ specificity < 0.92, 0.79 on average). In this case, the most probable configuration of detection methods is different from the same scenario with higher specificity (compare row 2 and 3). Under this assumption, the outbreak can be detected with the probability of 89.5% and the timeliness is above 0.88 in 31% cases. This kind of inference reveals the advantage of the Bayesian network model over simpler models like Logistic regression. Utilizing the Bayesian network model allows reasoning with uncertainty about complex relations of variables and predicting the performance of detection with relatively high accuracy.

In the fourth scenario, we set the specificity to be greater than 0.92 and we assumed the baseline data is sparse. In this scenario, the outbreaks happened when the mean of the baseline time series was relatively low. The most probable setting of algorithm parameter suggests that applying W2 and C2 algorithms will increase the chance of detecting outbreak. The detection probability is 49.3% under this circumstance of baseline data.

In the next scenario, we set the performance of detection to the highest level (i.e., specificity greater than 0.92, detection ratio of one, timeliness greater than 0.88) and looked for algorithm setting that is more likely to detect outbreaks with this performance. Regarding to inference of BN for this scenario, it is more probable to get higher detection performance by taking weekly patterns of time series into account, setting alerting threshold to the values less than 1.2, and using 2 days of guardband without memory. This setting belongs to W2 algorithm and Poisson regression with guardband.

3.7 Discussion and Conclusion

In this chapter, we analyzed outbreak detection performance for a range of algorithms that are widely used in public health practice, using an array of features related to the outbreak characteristics, baseline data, and the detection methods' parameters. We assessed the performance of seven different outbreak detection algorithms using simulated and real surveillance data for GI outbreaks in eighteen outbreak scenarios and trained Bayesian networks to model the relationships between all surveillance attributes and the detection performance. Our evaluation results show that even

when the outbreak characteristics were unknown a priori, the model was able to predict detection performance with high accuracy (AUC = 0.88).

The Bayesian network model developed in this work allows quantifying the effect of outbreak characteristics and algorithm configuration on the performance of detection algorithms. As expected, the most informative determinants of detection performance were the alerting threshold, which is a parameter of the detection method, and the contamination level and the peak size of the outbreak. But our model also quantified the contribution of other algorithm features such as accounting for day-of-week and maintaining a guardband or memory. We demonstrated how inference performed using our model can help to develop what-if analyses for using detection methods in practice, or to find an appropriate algorithm configuration given the desired level of detection performance for outbreak scenarios. Such an inferential tool gives insight about the features of detection methods that are important to provide better performance. We also described how the model can be utilized to predict the expected performance of detection methods in different surveillance situations. One limitation in comparing surveillance methods is the lack of data for benchmarking. This limitation was addressed in our work by using simulated data. Our approach is similar in spirit to the research reported by Lewis et al. (2013), who used simulation of influenza outbreaks to evaluate spatiotemporal outbreak detection methods. However, they did not quantify the effects of algorithm parameters on detection performance, and we believe this is an important contribution of our research. Our approach can be extended to allow a coherent evaluation of new algorithms and new data sources as needed. Using our current models, we can evaluate outbreak detection performance for new algorithms different than the C, W, and Adaptive Poisson algorithms. Any configuration of considered parameters in our model different than the ones belonging to these algorithms can be thought of as a new detection method and can also be evaluated.

We presented several scenarios of outbreaks and desired performance, and used inference to suggest the best algorithm and parameter setting to use, as well as to quantify the expected performance. These scenarios are by no means exhaustive, and are meant as examples of what kinds of inference can be performed. Of course, in order to use this model as a tool for what-if analysis in the public health sector, an adequate interface would also need to be developed, but this goes beyond the scope of our work.

We used emergency department visits as the baseline time series for outbreak detection. In recent years, non-traditional data sources have been introduced in public health and surveillance systems. This includes mobile phone data (Buckee et al., 2013), social data (Alasaad et al., 2013), micro-blogging (Donelle and Booth, 2012), Twitter feeds and Google search queries (Ginsberg et al., 2008). While introducing and combining new data sources, especially in the era of big data,

are promising directions for research in biosurveillance systems, the evaluation of their relevance and significance will be extremely important. Evaluation studies such as (Wilson et al., 2008) are needed to compare these new data sources to the existing ones, and the approach that we describe could be used to consider the relative contribution to detection performance of data sources and algorithms.

A number of extensions to this work may improve the generalizability of the results. We used simulated outbreaks superimposed on real surveillance data; therefore, the results are affected by the quality of the simulation. Our approach can be extended by using real surveillance data and including more detection methods. Spatio-temporal data simulations, as well as using additional health care utilization information in addition to ED visits, should be studied as well.

CHAPTER 4
HIERARCHICAL MIXTURE OF EXPERTS FOR OUTBREAK DETECTION

4.1 Machine Learning Combining Methods

As discussed in Chapter 1, one of the objectives of our research is improving the performance of outbreak detection algorithms. We propose to combine the prediction of several available outbreak detectors because improved performance can often be obtained by combining multiple algorithms in some way, instead of using one isolated algorithm. In this section, we describe some of aggregating methods that combine the output of several learners for solving classification problems or regression problems and generate a single output.

4.1.1 Majority Voting

Majority voting is an approach to combine prediction or classification methods. Assume that each classifier outputs a label for the instance i. The majority voting method finds the label which has been voted by the majority of the classifiers and outputs that label for the instance i. The predictions of different classifiers have the same weight and are treated equivalently in the unweighted majority voting method.

4.1.2 Hierarchical Mixture of Experts

Instead of averaging the prediction of several learners, an alternative form of combining learners is to select one learner based on the input variables and ask the selected leaner to make the prediction. A widely used combination method of this kind is a decision tree that can be described as a sequence of binary selections. In this method, the division of the input space is based on hard splits. It means that each learner is responsible for making predictions for input values of its corresponding subspace. By softening this limitation, a probabilistic combination method is obtained which is known as mixture of experts. If we have a set of K experts (i.e. learners), a probabilistic mixture is formed by

$$P(y|x) = \sum_{k=1}^{K} g_k(x) P_k(y|x) \tag{4.1}$$

where $P_k(y|x)$ is the probability of output y predicted by the *k*th expert and $g_k(x)$ represents the input-dependent mixing coefficient for that expert. The mixing coefficients are also known as *gating functions*. The gating functions $g_k(x)$ must satisfy the usual constraints of mixing coefficients, $0 \leq g_k(x) \leq 1$ and $\sum_k g_k(x) = 1$ (Bishop, 2006).

Now suppose that each expert in the mixture can be a mixture of experts itself. This multilevel gating network leads to a more flexible structure, known as hierarchical mixture of experts (HME) (Jordan and Jacobs, 1994). HME follows the strategy of divide-and-conquer in machine learning. This approach divides the input space into nested sequences of regions and fits simple hypotheses within these regions. Dividing the data may have favourable effect on the bias of the final predictor but it generally increases the variance. One variance-reduction solution is using soft splits in data. Soft splitting allows data to lie simultaneously in multiple regions, in contrast with hard splitting, where the subsets of data are not overlapping. The second solution for reducing variance is utilizing piecewise constant or piecewise linear functions. These functions reduce variance at a cost of increased bias (Jordan and Jacobs, 1994).

A two-level HME is shown in Figure 4.1. In the tree-based representation of HME, experts are leaves and gating networks are placed at the internal nodes. Each expert produces an output μ_{ij}. Gating networks receive the input x and generate the gating coefficients. Gating coefficients determine the proportion of contribution of each expert in forming the output of that gating network (Jordan and Jacobs, 1994).

Learning of the structure and adjusting the parameters of the HME is treated as a maximum likelihood problem. Jordan and Jacobs (1994) applies the Expectation-Maximization algorithm (described in section 3.2) for learning the structure on HME.

The HME method has been used in the speech recognition problem. Jacobs et al. (1991) developed the mixture of experts model for a speaker independent, four-class vowel discrimination problem in which the data formed two pairs of overlapping classes and different experts learned to concentrate on only one pair of classes. They compared this model with standard backpropagation networks and showed that the mixture of experts requires only about half as many epochs of backpropagation network to reach the same error criterion. The idea behind this application is that if a training data set can be naturally divided into subsets that correspond to subtasks, using a combination of experts and a gating network that decides which expert should be used for each subset will reduce the interference.

4.2 HME for Outbreak Detection

In this section, we want to train a classifier that predicts if there is an outbreak on a day or not. We used the predictions of statistical detection algorithms, C1, C2, C3, W2, and W3 with the threshold of 2, as the input of the HME whose goal is to predict the outbreaks. We used a training set with 111,034 instances and 5 features. The training set was the prediction of outbreak detection

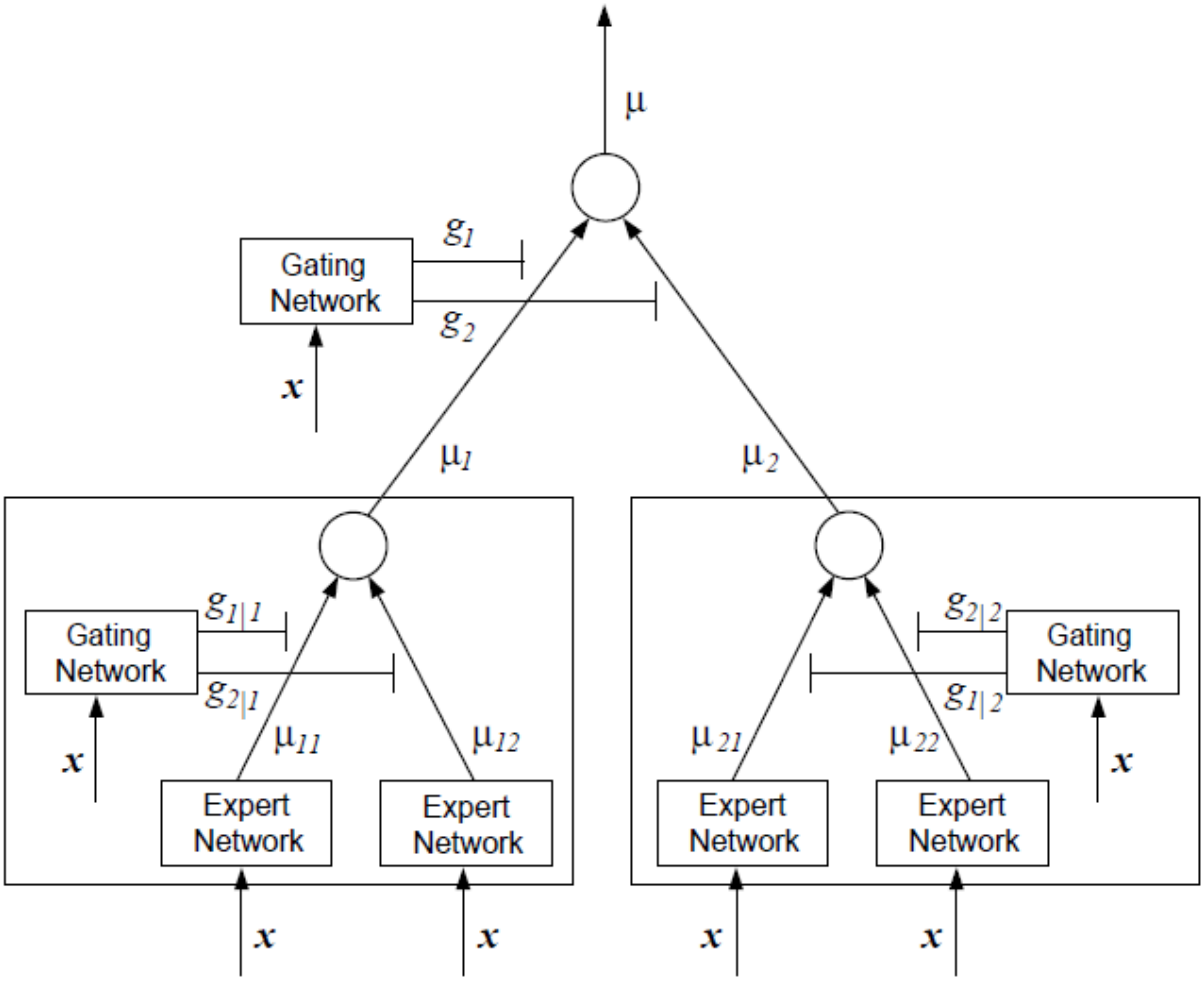

Figure 4.1 A two-level hierarchical mixture of experts (Jordan and Jacobs, 1994)

algorithms for 1,800 time series with different levels of contamination. We also created a testing set of 3,880,800 instances obtained from 1,800 time series that were not included in the training process.

We used the Bayes Net Toolbox of Matlab [1] 1.07 software for HME structure developed by Pierpaolo Brutti. The HME structure was learned from a batch training data set using desired number of iterations of the Expectation-Maximization (EM) algorithm. The architecture of the HME can be adjusted by choosing the number and the type of gating layers and experts. We used this set of procedures in Matlab to develop an HME for outbreak detection. We built an HME with 2 softmax gating levels and 4 softmax experts. The accuracy of prediction for training set was 61.21% and the confusion matrix was $\begin{bmatrix} 58093 & 1657 \\ 41403 & 9881 \end{bmatrix}$. The confusion matrix is defined in Table 4.1. The classification accuracy for the testing set was 95.70% and the confusion matrix was $\begin{bmatrix} 3703855 & 125949 \\ 40893 & 10103 \end{bmatrix}$. The sensitivity of detection of HME with 5 features was 0.464 and the specificity was 0.967.

We enhanced the features of training data by including 7 more features that show the predictions of W3 algorithm in 7 recent days. The idea behind including these predictions is that if there has been an alert in recent days, it is more likely an outbreak occurred. We created the training and testing data similar to the previous experiment. The accuracy of prediction of an HME with 2 softmax gating levels and 4 softmax experts on training data was 64.33% and the confusion matrix was $\begin{bmatrix} 49282 & 10468 \\ 29136 & 22148 \end{bmatrix}$. The accuracy for testing was 82.45% and the confusion matrix was $\begin{bmatrix} 3177394 & 652410 \\ 28482 & 22514 \end{bmatrix}$. The detection performance of the HME was summarized in Table 4.2. The result shows that including the recent predictions of a statistical detection algorithm did not improve the classification accuracy of HME. However, the sensitivity of detection was slightly better in the second experiment.

1. https://code.google.com/p/bnt

Table 4.1 Confusion matrix for binary classification

	Predicted class	
Actual class	-	+
-	$N_{-,-}(x)$	$N_{-,+}(x)$
+	$N_{+,-}(x)$	$N_{+,+}(x)$

We created another training set with 15 features. This time, we used the predictions of C1, C2, C3, W2, and W3 algorithm with 3 different values of threshold per algorithm. We used the training and testing data with the same size as previous ones. An HME with 2 softmax gating levels and 4 softmax experts classified the training data with the accuracy of 64.72% and the confusion matrix of $\begin{bmatrix} 49941 & 9809 \\ 29359 & 21925 \end{bmatrix}$. The accuracy of prediction in testing set was 83.45% (and the confusion matrix was $\begin{bmatrix} 3216980 & 612824 \\ 29248 & 21748 \end{bmatrix}$. The detection performance is shown in Table 4.2. So far, The highest classification accuracy was obtained by the HME with 5 features, the predictions of C1, C2, C3, W2, and W3 algorithm.

Since the performance of W3 algorithm is usually higher than other detection algorithms, we built an HME model with only using the predictions of W3 with different thresholds. We evaluated this model with 4, 5, and 6 alerting thresholds of W3 (thresholds of 1.5, 2, 3, 3.5, 4, 4.5, 5). The classification accuracy of the three models was the same. An HME with 2 softmax gating levels and 4 softmax experts classified 62.69% training instances correctly with the confusion matrix of $\begin{bmatrix} 48608 & 11142 \\ 30277 & 21007 \end{bmatrix}$. The accuracy of prediction on testing data was 80.63% with the confusion matrix of $\begin{bmatrix} 3107912 & 721892 \\ 29743 & 21253 \end{bmatrix}$. The detection performance of this HME structure is almost the same as the performance of W3 algorithm.

Table 4.2 summarizes the detection performance of 4 developed HME structures with different classification thresholds in terms of the timeliness, specificity, and sensitivity of detection. Also, the detection performance of statistical outbreak detection algorithms is shown in this table. Figure 4.2 plots the ROC curve of developed HME models and W2 and W3 algorithms. The best accuracy of classification obtained by the HME with 5 features, the prediction made by C1, C2, C3, W2, and W3 algorithm. Regarding to the ROC curve, the W3 algorithm and the HME using W3 predictions show the best detection performance.

4.3 HME for Different Scales of Contamination

We generated a data set including the prediction of C1, C2, C3, W2, and W3 detection algorithms and 7 recent predictions of W3 algorithm. The hypothesis is that information on whether or not in recent days an outbreak has been detected will improve the certainty of predictions in a surveillance system. We created various training and testing data sets using the surveillance data with different scales of contamination. Figure 4.3 shows the workflow of this experiment.

Table 4.2 Detection performance of developed HME structures vs. detection algorithms

Algorithm	Threshold	Specificity	Sensitivity	Timeliness
C1	0	0.377	0.8698	0.9674
	1	0.927	0.5548	0.7554
	2	0.996	0.2203	0.2921
C2	0	0.991	0.2237	0.2986
	1	0.998	0.1696	0.2166
	2	1	0.1227	0.1496
C3	2	0.991	0.2706	0.3222
	3	0.998	0.1849	0.2261
	4	1	0.1404	0.1603
W2	2	0.969	0.4634	0.655
	3	0.995	0.347	0.4749
	4	1	0.2907	0.3993
W3	3	0.6042	0.7688	0.9396
	4	0.8144	0.6577	0.8612
	5	0.903	0.554	0.729
HME with 5 features	0.3	0	1	1
	0.4	0.28	0.904	0.975
	0.5	0.967	0.464	0.657
	0.6	0.968	0.464	0.657
	0.7	0.969	0.463	0.655
HME with 12 features	0.3	0.101	0.964	0.991
	0.4	0.52	0.781	0.922
	0.5	0.83	0.582	0.777
	0.6	0.968	0.464	0.656
	0.7	0.97	0.458	0.644
HME with 15 features	0.3	0	1	1
	0.4	0.76	0.65	0.828
	0.5	0.84	0.557	0.745
	0.6	0.942	0.49	0.657
	0.7	0.987	0.398	0.552
HME with 5 features (W3 predictions)	0.3	0	1	1
	0.4	0.598	0.769	0.94
	0.5	0.812	0.658	0.861
	0.6	0.902	0.554	0.729
	0.7	0.902	0.554	0.729

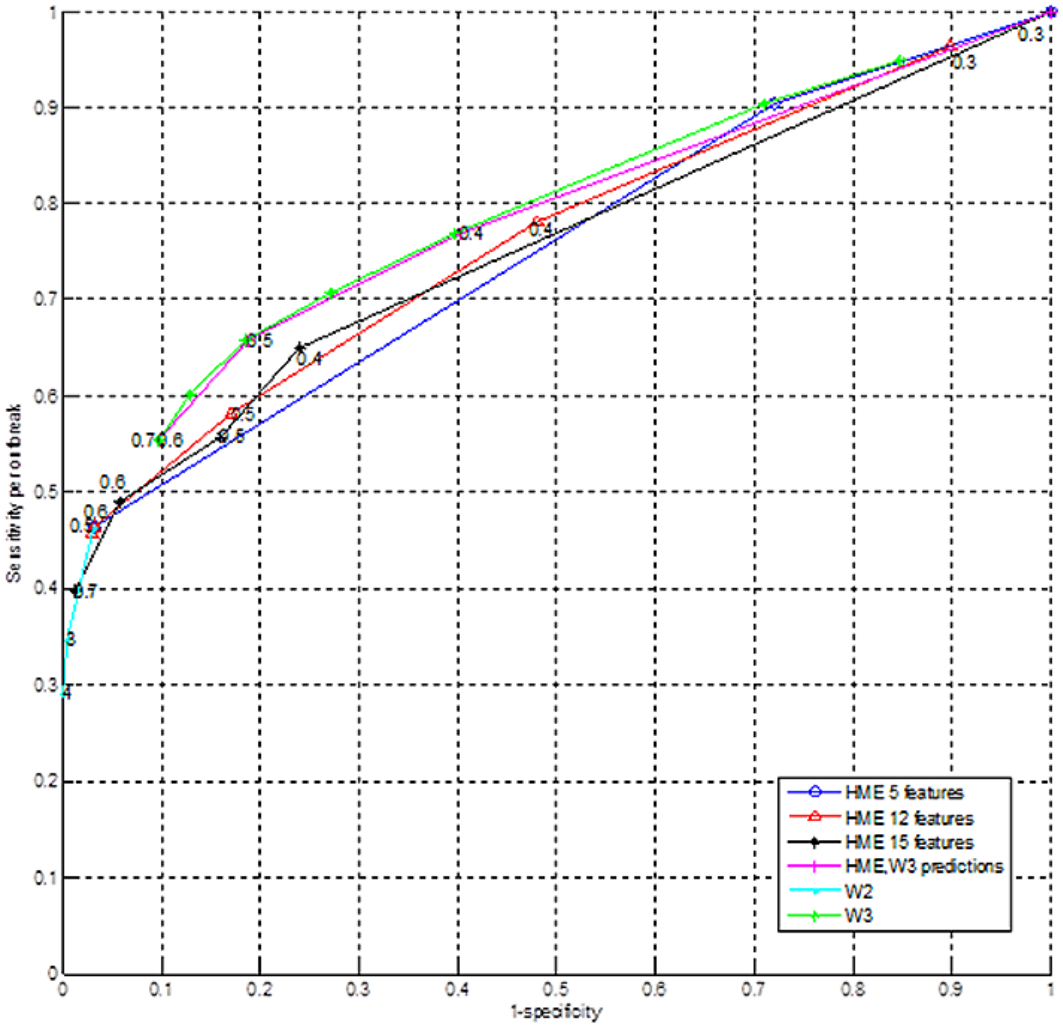

Figure 4.2 ROC curve of HME structures vs. W2 and W3

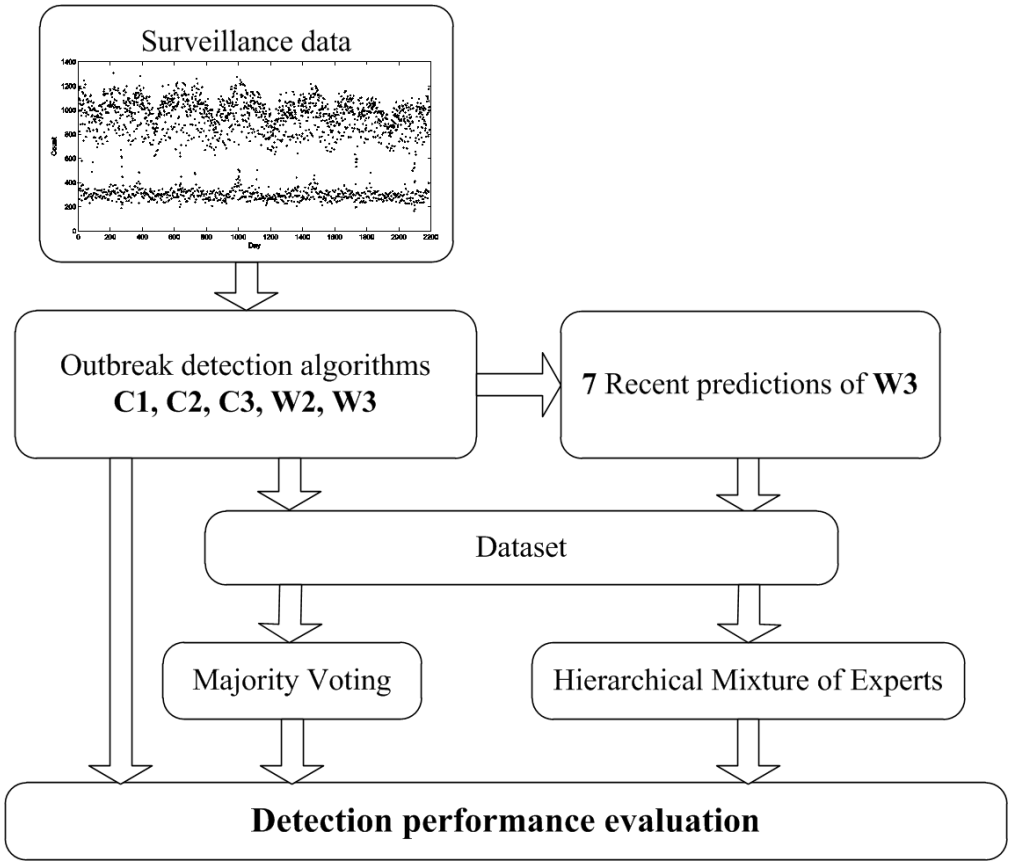

Figure 4.3 Workflow of learning combining methods for outbreak prediction

In the first experiment, we used the predictions of C and W algorithms for 300 time series with low contamination to create the training set with 14538 instances. To evaluate the accuracy of developed model, we built a testing set from 90 time series that are not included in the training process (194040 instances). We used the training data to learn an HME structure with 5 gating levels and 32 experts in the lowest level of hierarchy. We also evaluated the majority voting algorithm on the testing data.

Figure 4.4 shows the trade-off between the sensitivity and the specificity of majority voting and the HME with different thresholds of classification. It also illustrates the ROC curves of C1, C2, C3, W2, and W3 detection algorithms with different alerting thresholds evaluated on the same testing time series. Focusing on upper left side of the ROC curves, the best sensitivity of the HME was 0.711 at the specificity of 0.898 and threshold of 0.4. At the best point of ROC of majority voting, the sensitivity was 0.2 and the specificity was 0.946.

In the second experiment, we used simulated surveillance time series with higher scales of contamination to create the training and testing data sets. Because the level of contamination of water affects the outbreak characteristics (e.g. outbreak duration, outbreak peak size, shape of the signal), therefore, the performance of different detection algorithms varies based on different outbreak characteristics among these contamination scenarios. We created a training set based on predictions of C and W family algorithms for 75 time series with high contamination including 4494 instances and a testing set from untouched 30 time series containing 64680 instances. The ROC curves of majority voting and HME are shown in Figure 4.5 using different thresholds. It also plots the ROC curves of C and W family algorithms evaluated on the same time series with high contamination. The sensitivity of majority voting was 1 when the specificity was 0.946. In contrast, the sensitivity of HME is 1 at the specificity of 0.963.

In the third experiment, we used time series with different levels of contamination in order to have an overall view of the detection performance of algorithms. We built a training data set based on 150 time series and a testing set on 60 time series. Figure 4.6 shows the ROC curve of the developed HME and majority voting versus C and W family algorithms.

To assess the timeliness of developed methods versus C and W family algorithms, we evaluated the timeliness of algorithms when the specificity was set to 0.99. The false positive ratio obtained from this configuration was 1%. We chose this level of specificity because the alerting ratio of 1 in 100 days is a practical one for public health surveillance (Xing et al., 2011). We evaluated the timeliness testing on 3 types of simulated time series data: time series with low contamination, time series with high contamination, and time series with low and high contamination. The time-

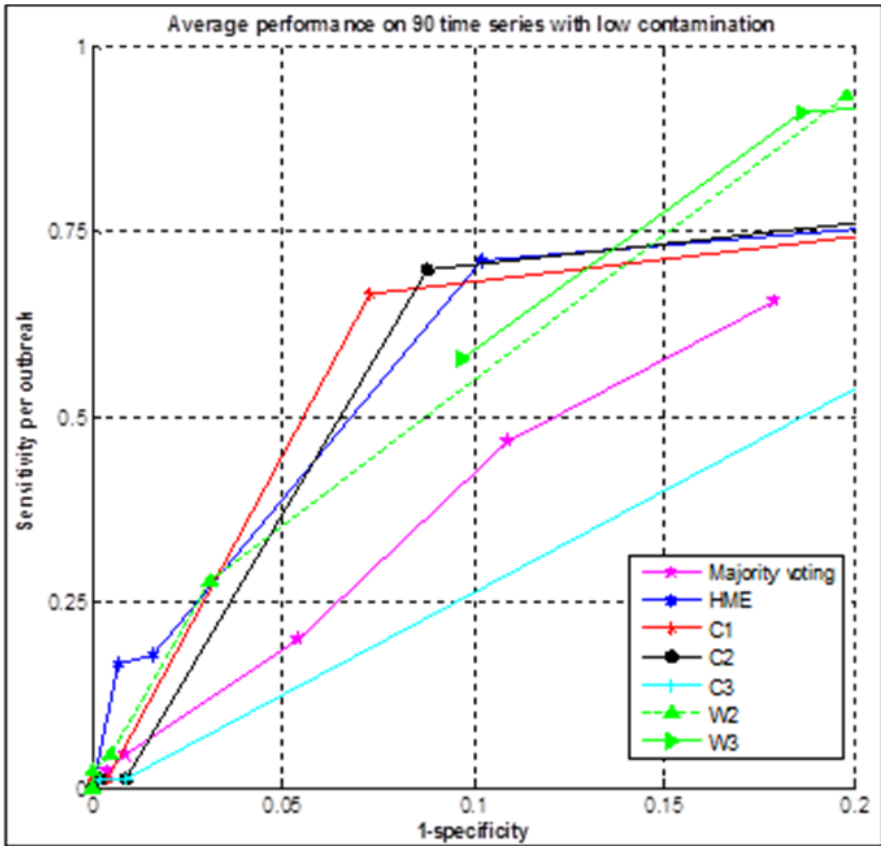

Figure 4.4 ROC curve of Majority voting, HME, and C W family detection algorithms evaluated on surveillance time series with low contamination

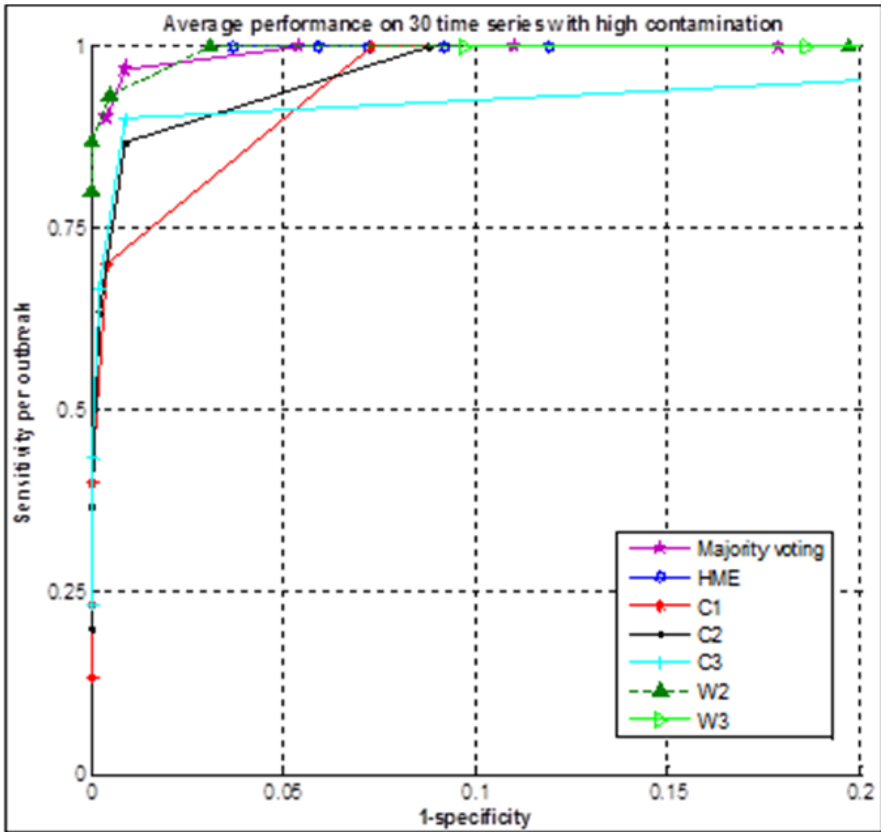

Figure 4.5 ROC curve of Majority voting, HME, and C W family detection algorithms evaluated on surveillance time series with high contamination

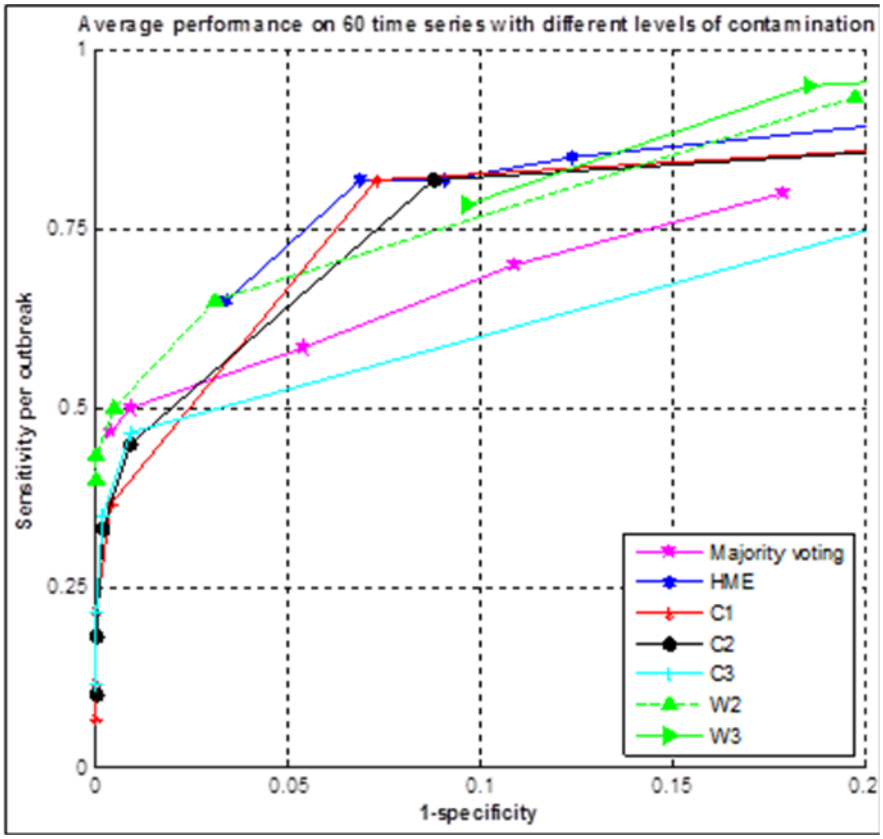

Figure 4.6 ROC curve of Majority voting, HME, and C W family detection algorithms evaluated on surveillance time series with different scales of contamination

liness of the voting majority, HME, and C and W family algorithms with the specificity of 0.99 is summarized in Table 4.3 for 3 testing data sets.

In the first row of the table, the best timeliness belongs to majority voting, however, this comes with the sensitivity of 0.044 which is lower than the sensitivity of other algorithms. In the second row, W2 has the best timeliness of detection. The best timeliness of detection over the mixed testing data was obtained by W2, however, the timeliness of the developed HME is very close to the best timeliness.

The experimental results show that the detection performance of the developed HME algorithm in terms of sensitivity and specificity is higher than simple majority voting algorithm. So it is valuable to build an HME to aggregate different predictions rather than using simple majority voting algorithm. The results also show that the detection performance of C and W algorithms is dependent on the outbreak characteristics (i.e. contamination level) and there is no one single algorithm that always outperforms other methods under different circumstances. Although the detection methods behave with various performance levels, the developed HME algorithm is competitive to the best detection algorithm in all three experiments and the level of contamination of surveillance time series does not influence the relative performance of the HME. Hence, the developed detection algorithm based on HME is more robust under different circumstances.

4.4 HME with Baseline Characteristics

We created other HME structures with new feature sets. In addition to the predictions of outbreak detection methods, we used some of baseline characteristics as the features of training data. For example, we used the count of patients divided by the mean of number of patients in 7 recent days as a feature. Another baseline characteristic used as a feature is the Z-score: the difference of the count of patients and the mean of patients in 7 recent days divided by the standard deviation. Other features of training data are the predictions of C1, C2, C3, W2, W3, Poisson regression with and without guardband, and GLRT algorithm (Described in 2.2.5). Table 4.4 shows the accuracy of some of HME models trained by new feature set. The first, second and last models have 2 gating

Table 4.3 Timeliness of Majority voting, HME, and C W family detection algorithms with the specificity of 0.99

Testing data	Majority voting	HME	C1	C2	C3	W2	W3
Low contamination	**1**	0.5	0.143	0.143	0.143	0.534	0.917
High contamination	0.727	0.682	0.688	0.74	0.731	**0.804**	0.75
Mixed high and low contamination	0.727	0.773	0.685	0.731	0.731	**0.796**	0.75

networks and 4 experts at the bottom of the structure while others have 5 gating networks and 32 experts.

This result shows that considering the baseline characteristics could slightly improve the accuracy of prediction, specially when both features were used to train the model. The accuracy of the models trained by data with high contamination is higher than other models. Because outbreaks with high contamination are usually bigger in size and easier to detect.

4.5 Discussion and Conclusion

In this chapter, we proposed to combine various outbreak detection methods for the purpose of improving the overall detection performance in surveillance systems. We described a framework based on HMEs that can be employed for this method fusion. In addition, we used a majority voting strategy as a simpler alternative for aggregating detection methods. Our experimental evaluation of these two approaches to method fusion does not seem to provide improvement over the best detection methods for the particular surveillance scenarios used in this work. However, HME outperforms most algorithms tried in our experiments.

In the developed models, we use the predictions of C and W family detection algorithms, however, the models can be extended to consider the predictions of other detection algorithms, like Negative binomial CUSUM and Poisson Regression. Because they may provide more diversity of predictions.

We developed the aggregating methods based on temporal surveillance data. We can extend our work to consider difference sources of surveillance data, like spatiotemporal surveillance data to detect outbreak event in several regions. We can also use other sources of surveillance data, like school and work absenteeism rates and others. In the developed HME, the predictions of detection algorithms were fed to all the experts of the structure. The model can be improved by assuming that each detection algorithm is an expert in the HME architecture and its vote is weighted based on determinants of detection performance (e.g. outbreak characteristics, desired false alarm rate).

Table 4.4 Accuracy of HME models

Features	Training/testing data	Accuracy	Confusion matrix
Predictions of C1, C2, C3, W2, W3, Poisson regression with and without guardband, GLRT	100 partial time series with low contamination, 4014 instances	66.76%	$\begin{bmatrix} 2669 & 7 \\ 1327 & 11 \end{bmatrix}$
	25 time series with low contamination, 53,900 instances	98.94%	$\begin{bmatrix} 53327 & 247 \\ 324 & 2 \end{bmatrix}$
count divided by the mean of 7 days, Predictions of C1, C2, C3, W2, W3, Poisson regression with and without guardband, GLRT	100 partial time series with low contamination, 4014 instances	66.74%	$\begin{bmatrix} 2670 & 6 \\ 1329 & 9 \end{bmatrix}$
	25 time series with low contamination, 53,900 instances	98.98%	$\begin{bmatrix} 53352 & 222 \\ 324 & 2 \end{bmatrix}$
same features	300 partial time series with high contamination, 25854 instances	80.50%	$\begin{bmatrix} 16420 & 816 \\ 2674 & 5944 \end{bmatrix}$
	50 time series with high contamination, 107,800 instances	94.82%	$\begin{bmatrix} 101217 & 5097 \\ 479 & 1007 \end{bmatrix}$
Z-score, Predictions of C1, C2, C3, W2, W3, Poisson regression with and without guardband, GLRT	300 partial time series with low contamination, 12120 instances	66.65%	$\begin{bmatrix} 8075 & 5 \\ 4036 & 4 \end{bmatrix}$
	50 time series with low contamination, 107,800 instances	99.33%	$\begin{bmatrix} 107084 & 50 \\ 666 & 0 \end{bmatrix}$
same features	300 partial time series with high contamination, 25854 instances	82.71%	$\begin{bmatrix} 15461 & 1775 \\ 2694 & 5924 \end{bmatrix}$
	50 time series with high contamination, 107,800 instances	89.01%	$\begin{bmatrix} 94962 & 11352 \\ 493 & 993 \end{bmatrix}$
same features	360 partial time series, 16840 instances	64.06%	$\begin{bmatrix} 7305 & 1115 \\ 4937 & 3483 \end{bmatrix}$
	180 time series, 388080 instances	85.66%	$\begin{bmatrix} 330774 & 53118 \\ 2532 & 1656 \end{bmatrix}$
Z-score, count divided by the mean of 7 days, Predictions of C1, C2, C3, W2, W3, Poisson regression with and without guardband, GLRT	360 partial time series, 25260 instances	74.68%	$\begin{bmatrix} 16069 & 771 \\ 5624 & 2796 \end{bmatrix}$
	180 time series, 388080 instances	94.21%	$\begin{bmatrix} 364363 & 19529 \\ 2924 & 1264 \end{bmatrix}$

CHAPTER 5
COST ANALYSIS OF OUTBREAK DETECTION METHODS

In most of previous works, disease outbreak detection methods have been evaluated based on specificity-sensitivity trade off. However, in practice, missing outbreaks are more costly than false alarms. In this chapter, we analyze some of commonly used outbreak detection methods considering the separate cost of missing outbreaks and false alarms. We applied those methods to the simulated data and estimated the total cost of missing outbreaks and false alarms based on a study on cost analysis in addition to accuracy of outbreak detection. We also developed a cost-sensitive decision tree which predicts the outbreak based on the prediction of commonly used detection methods while minimizing the total cost of outbreaks.

The cost of missing outbreak depends on the delay in detecting the outbreak and the level of contamination of outbreak. The later an outbreak is detected, the greater the cost. Also, an outbreak with higher contamination results in the higher cost. On the other hand, false alarms decrease the reliability of the system lead to unnecessarily costly preventions, e.g., boiling water and consuming bottled water. However, investigating the cost of false alarms is not straightforward. In this study, we estimated the cost of missing outbreaks based on a cost analysis study. The cost is an estimated function of delayed time in outbreak detection. For every simulated surveillance time series, this cost is different due to the different number of symptomatic cases, the contamination level, and the boil-water advisory time. We assumed the cost of false alarms is a fraction of missing outbreaks.

5.1 Background

5.1.1 Cost-sensitive Classification

In some of real-life decision making situations, the assumption of equal misclassification costs is unrealistic. For example, in medical diagnosis, a false negative prediction, i.e. failing to detect a disease, may have fatal consequences, whereas a false positive prediction, i.e. diagnosing a disease for a patient that does not actually have it, may be less serious. In the cases where the cost of misclassifying plays a great role, cost-sensitive learning algorithm is critical.

Typically, the cost information is represented in the form of a cost matrix *C*, where each row represents a single predicted class and each column an actual class. Table 5.1 illustrates the cost matrix for the case of two classes. The cost of a true positive is denoted $C_{+,+}(x)$, the cost of a true negative is $C_{-,-}(x)$, the cost of a false positive is $C_{+,-}(x)$, and the cost of a false negative is $C_{-,+}(x)$ (Viaene and Dedene, 2005). The *reasonableness* conditions imply that the cost of

labelling a data instance incorrectly is always greater than the cost of labelling it correctly, such that, $C_{+,-}(x) \geq C_{-,-}(x)$ and $C_{-,+}(x) \geq C_{+,+}(x)$ (Elkan, 2001). The cost of $C_{+,+}(x)$ and $C_{-,-}(x)$ are usually set to zero.

The optimal decisions are not changed if all the entries in the cost matrix are multiplied by a positive constant. This scaling corresponds to altering the unit of costs. Similarly, the optimal decisions are unchanged if a constant is added to the entries of the matrix. This shifting refers to changing the baseline from which the costs are measured. Any cost matrix can be transformed into a simpler matrix by scaling and shifting while the decisions are the same (Elkan, 2001).

The error rate of a cost-sensitive classifier might be higher than regular classifiers however, the cost of misclassification is less than the regular one. Because the objective function of a cost-sensitive classifier is based on minimizing the misclassification cost.

Existing works on cost-sensitive learning can be categorized into two groups, one is to design cost-sensitive learning algorithms directly, and the other is to design a wrapper to convert existing cost-insensitive base learning algorithms into cost-sensitive ones. Some examples of wrapper method, or cost-sensitive meta-learning, are relabelling, weighting, and threshold adjusting.

Relabelling is an approach to make an error-based learner cost-sensitive by manipulation of the training data instance target labels. The learner can be treated as a black box, requiring no knowledge of its internals or change to it. MetaCost is based on relabelling each training data instance with its optimal target label with the minimum cost, and then learning the final classification model using the relabelled training data (Domingos, 1999).

The weighting approach is based on having different initial weights for instances to reflect the given misclassification cost. This influence the learner to focus on instances with higher misclassification cost. This approach can adapt to existing learning algorithms in order to minimize the misclassification cost. As an example, it has been combined with standard tree induction process to include both minimum error trees and minimum cost trees (Ting, 1998).

Table 5.1 Cost matrix for binary classification (Viaene and Dedene, 2005)

	Actual class	
Predicted class	-	+
-	$C_{-,-}(x)$	$C_{-,+}(x)$
+	$C_{+,-}(x)$	$C_{+,+}(x)$

Thresholding is a threshold adjusting approach for cost-sensitive meta-learning. This method finds the best probability estimate from training instances as the threshold, and uses it to predict the class label of test instances: a test example with predicted probability above or equal to this threshold is predicted as positive; otherwise as negative. The total misclassification cost for each possible probability estimates on the training examples can be calculated. Thresholding chooses the best threshold that minimizes the total misclassification cost. To reduce overfitting, thresholding uses cross-validation and searches for the best probability from the validation sets (Sheng and Ling, 2006).

Decision trees are the widely used classification methods in machine learning. A common approach to constructing a decision tree is to grow a full tree and then prune it back. The pruning process performs a post-order traversal of the tree and replaces a subtree by a single leaf when the estimated error of the leaf is lower than the subtree. Pruning of a decision tree is a desirable task in order to avoid overfitting the training data. Bradford et al. (1998) described several pruning methods for decision trees whose goal is minimizing the cost (loss) rather than error. They found that while pruning reduces the size of the tree and the classification loss, there is no pruning method dominating others on all datasets and different mechanisms are better for different cost matrices.

Assume a regular decision tree classifies instance x in class $+$ if the posterior probability of $+$ is bigger than the probability of class $-$. We developed a tree that classifies instance x in class $+$ if the $P(+) * C_{-,+}(x)$ is less than the $P(-) * C_{+,-}(x)$. The tree is built according to splitting criteria that minimize the total cost, instead of minimizing entropy.

In spite of cost-sensitive classifiers and cost-sensitive pruning approaches simplicity, we did not follow the existing studies for our problem because the misclassification cost of every instance in this data is different and there is no single fixed values of $C_{+,-}(x)$ and $C_{-,+}(x)$ for the whole dataset.

The cost matrix of cost-sensitive binary classifiers is defined as $2 * 2$ matrix, however, we defined the cost matrix as an $N * 2$ matrix for a dataset of N instances with different misclassification costs. We considered misclassifying cost per instance in building the classifier rather than misclassifying cost per class.

5.1.2 Decision Trees

There are various simple, but widely used, models that work by partitioning the input space into cuboid regions and then assigning a simple model to each region. The process of selecting a specific model, given a new input, can be described by a sequential decision making process corresponding

to the traversal of a binary tree (Bishop, 2006). A decision tree is a sequence of binary selections in the data. Internal nodes of the tree are tests on the values of different attributes. Each training example falls in precisely one leaf and its class is determined based on the majority of instances in that leaf.

For creating a decision tree, all possible splits on every feature are examined and a split with best optimization criterion is selected. For a continuous attribute, a tree can split halfway between any two adjacent unique values found for this attribute. For a categorical attribute with *L* levels, a classification tree needs to consider $2L^{-1}-1$ splits to find the optimal split. Alternatively, a heuristic algorithm can find a good split. If the split leads to a child node having too few observations which is less than the minimum leaf, another split with the best optimization criterion subject to the Minimum Leaf constraint is selected. The split is imposed and the procedure is repeated until meeting stopping criterion. Some of stopping rules are as follows:

- When the node is pure. A node is pure if it contains only observations of one class.
- Any split imposed on this node would produce children with fewer than Minimum Leaf observations.

There are several different optimization criteria for classification decision trees. Some of them are:

1. Gini's Diversity Index: The Gini index of a node is

$$1 - \sum_i p^2(i) \tag{5.1}$$

   where the sum is over the classes *i* at the node, and *p(i)* is the observed fraction of classes with class *i* that reach the node. A node with just one class is a pure node and has Gini index 0; otherwise the Gini index is positive. So the Gini index is a measure of node impurity (Bishop, 2006).

2. Deviance : With *p(i)* defined as for the Gini index, the deviance of a node is

$$-\sum_i p(i)\log p(i) \tag{5.2}$$

   A pure node has deviance 0; otherwise, the deviance is positive.

3. Twoing rule: Twoing is not a purity measure of a node, but is a different measure for deciding how to split a node. If L(i) denotes the fraction of members of class i in the left child node after a split, and R(i) denotes the fraction of members of class i in the right child node after a

split, the split criterion is chosen to maximize

$$P(L)P(R)(\sum_i |L(i) - R(i)|)^2 \tag{5.3}$$

where *P(L)* and *P(R)* are the fractions of observations that split to the left and right respectively. If the expression is large, the split made each child node purer. Similarly, if the expression is small, the split made each child node similar to each other, and hence similar to the parent node, and so the split did not increase node purity (Olshen and Stone, 1984).

5.1.3 Outbreak Costs

Anomalies in surveillance data are suggestive in detecting outbreaks but not conclusive. Wagner et al. (2005) studied the decision to issue a boil-water advisory in response to a spike in sales of diarrhea remedies or wait for the results of testing water. They modelled the decision analysis in a decision tree structure like Figure 5.1.

This model consisted of a decision node *Issue boil-water advisory* with two possible actions: *Act Now* to issue a boil-water advisory or *Wait* for confirmation; a chance node *Crypto outbreak* with two possible values for when there is a Cryptosporidium outbreak or not. The probability *p* is the probability of an outbreak. The costs incurred for the events are $C_{+,+}, C_{+,-}, C_{-,+}$, and $C_{-,-}$. $C_{+,+}$ represents the outcome of true alarms: when the decision maker issued a boil-water advisory and the outbreak occurred. Similarly, $C_{+,-}$ describes the outcome of false alarms and $C_{-,+}$ is the cost of false negatives. The cost of no advisory while there is no outbreak, $C_{-,-}$, is set to zero.

In this work, we estimated the cost of missing outbreaks (i.e. false negatives, $C_{-,+}$) based on a cost analysis study. We used a fraction of average cost of missing outbreaks as the cost of false alarms ($C_{+,-}$). The total cost is the sum of these two outcomes. First, we analyzed some of commonly used outbreak detection methods regarding to the cost of missing outbreaks and false alarms. We used our simulated data of waterborne disease outbreaks with varying duration and magnitude. We applied those methods to the simulated data and estimate the total cost of missing outbreaks and false alarms in addition to accuracy of outbreak detection. Afterwards, we develop a detection method whose goal is not only predicting the outbreaks accurately, but also minimizing the total cost of missing outbreaks and false alarms.

5.1.4 Cost Analysis

Despite the relatively minor symptoms and short duration of cryptosporidiosis, an outbreak is costly to society due to the large number of cases and the complications that arise in a small pro-

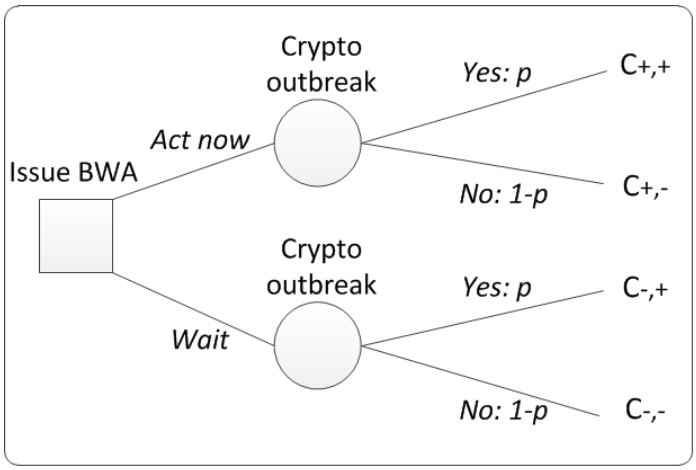

Figure 5.1 A decision tree model to issue a boil-water advisory (Wagner et al., 2005)

portion of cases. A cost calculator was developed that calculates this cost from the societal perspective, accounting for such factors as lost productivity to estimate the overall economic burden of cryptosporidiosis. The estimated cost of an outbreak of cryptosporidiosis can be broken down into two parts: direct and indirect costs, where direct costs are those that directly result from illness prevention or health-care utilization and indirect costs are those resources that could not be acquired as a result of illness or of provision of care to a person who is ill.

Included in the direct costs are the costs of medications, outpatient, and inpatient costs. A calculation of the total cost of medications takes into account the duration of illness before and after care is sought, the percentage of symptomatic individuals who are expected to take medications, the cost of the medicines, and the typical dosage. For the cost of outpatient visits, the fee-for-service billing information was used from the Manuel des Médecins Omnipraticiens of the Régie de l'Assurance Maladie du Québec (RAMQ). Since the cost of an emergency department visit has not been quantified for Québec, the ambulatory care data from the Ontario Case Costing Initiative (OCCI) was used. This database also includes costs of hospitalization for typical and atypical inpatient cases, as well as the average length of stay. Physician fees are not included, except for those who are on salary. It was assumed that emergency department physicians are not on salary, while those who work in the hospital are on salary.

Indirect costs include age-specific lost productivity and mortality. A calculation of lost productivity takes into account the percentage of individuals employed and the average daily wage, as reported by Statistics Canada, as well as the number of work-days lost due to illness generated by the Disease component. Those who are unemployed have no lost productivity; however, in cases under 17 or over 65 years of age, a caregiver incurs the costs of lost productivity.

Since the anticipated costs per individual case vary largely depending on the level of medical care acquired, all simulated cases of symptomatic disease were divided into four mutually exclusive categories based on the highest level of care obtained: 1) No medical attention 2) Emergency department visits 3) Physician visits 4) Hospitalization.

Since the costs of cryptosporidiosis are also largely dependent on age, the cost calculator differentiates between five age groups for cost calculations: less than five years of age, 5-17, 18-49, 50-64, and more than 65 years of age. Among those who seek care, immunocompromised individuals are considered as a separate group, due to the increased cost of medical care for them (cite Rolina).

5.2 Experimental Results

5.2.1 Simulated Surveillance Data

In the experiments for analyzing the cost, we used the simulated surveillance data which was described in section 2.3 of Chapter 2. In the simulation scenarios for generating this data, two parameters were varied systematically: the duration of water contamination, which was varied over 6 values (72, 120, 168, 240, 360 and 480 hours), and the cryptosporidium concentration, which was varied over 3 levels ($10^{-6}, 10^{-5}, 10^{-4}$, corresponding to 0.01, 0.1, and 1 c.parvum oocyst per 1 Litre, respectively). Then, several scenarios of boil-water advisory were assumed in the simulation. In other words, it was assumed that a boil-water advisory is issued 240, 288, 336, 384, 432, 480, 528, 576, 624, or 672 hours after the starting time of the simulation. The advisory which comes 1000 hours after the simulation represents that no advisory was issued and the outbreak signal was not affected. The size of the superimposed signal is usually decreased after issuing boil-water advisory.

5.2.2 Linear Approximation of Cost Based on Time of Advisory

We chose 50 time series from each scenario of contamination and boil-water advisory. We calculated the cost of outbreaks for these 5850 different time series. We fitted Polynomial regression models (degree: 1, 2, 3) to approximate the cost based on the advisory time using statistical curve fitting toolbox in MATLAB. We did not remove outliers because they correspond to expensive outbreaks. These models are summarized in Table 5.2 and shown in Figure 5.2. Table 5.3 shows the goodness of fit.

The R-square is the square of the correlation between the response values and the predicted response values. A value closer to 1 indicates that a greater proportion of variance is accounted for by the model. R-square indicates the goodness of fitted models.

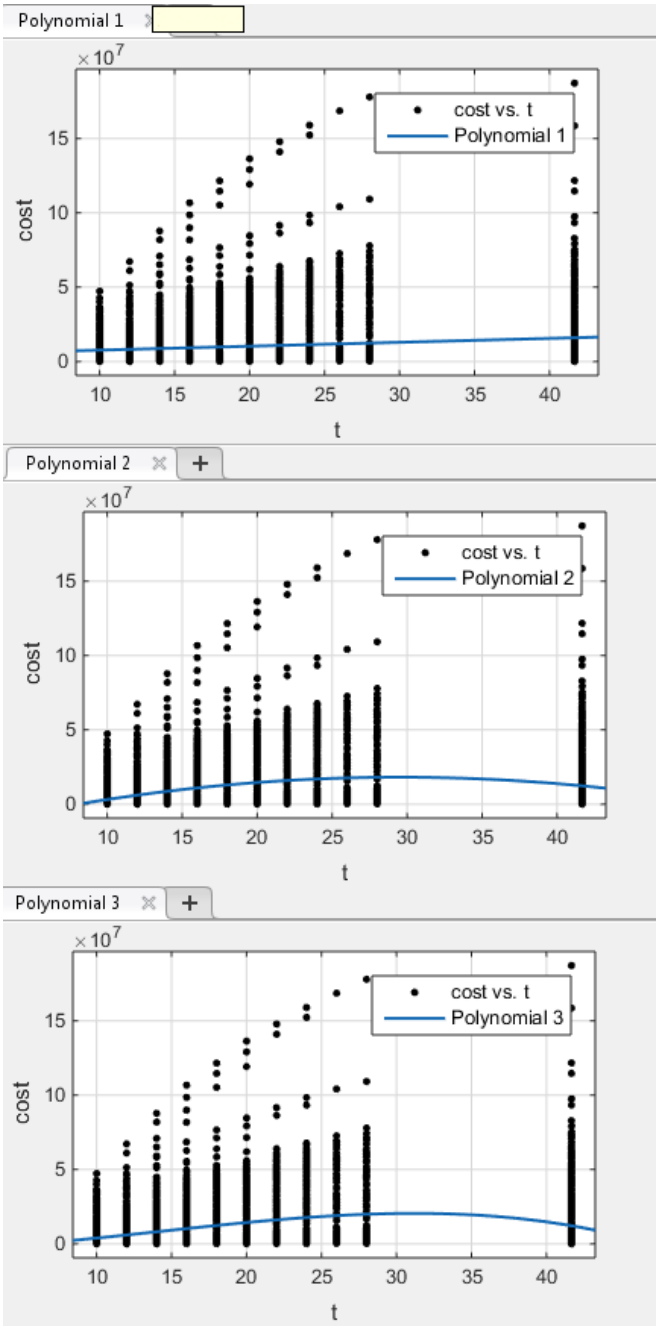

Figure 5.2 The illustration of cost function approximators

Table 5.2 Cost function approximators

Model	Function	Coefficients (95% confidence bounds)
Polynomial 1	$cost(t) = p1 * t + p2$	p1= 2.63e+05 (2.221e+05, 3.038e+05) p2= 5.029e+06 (4.114e+06, 5.945e+06)
Polynomial 2	$cost(t) = p1 * t^2 + p2 * t + p3$	p1= -3.945e+04 (-4.396e+04, -3.494e+04) p2= 2.329e+06 (2.089e+06, 2.568e+06) p3= -1.621e+07 (-1.88e+07, -1.362e+07)
Polynomial 3	$cost(t) = p1 * t^3 + p2 * t^2 + p3 * t + p4$	p1= -1230 (-2064, -395.7) p2= 5.196e+04 (-1.02e+04, 1.141e+05) p3= 3.482e+05 (-1.016e+06, 1.713e+06) p4= -3.576e+06 (-1.253e+07, 5.373e+06)

Table 5.3 The goodness of fit for cost function approximators

Model	R-square	Adjusted R-sq	SSE	RMSE
Polynomial 1	0.02656	0.02639	1.596e+18	1.652e+07
Polynomial 2	0.07311	0.0728	1.519e+18	1.612e+07
Polynomial 3	0.07444	0.07396	1.517e+18	1.611e+07

Table 5.3 shows that the polynomial models with higher degree of freedom are better fits for the data (in terms of goodness of fit i.e., SSE and RMSE) however, considering the fact that the late advisories lead to the higher cost of outbreak, the approximation model should be an ascending curve. So, we chose linear regression to approximate the cost.

5.2.3 Feature Selection for Outbreak Detection

We wanted to create a binary classifier which classifies outbreak days versus non-outbreak days based on the predictions made by statistical outbreak detection algorithms. We used predictions of C1, C2, C3, W2, and W3 algorithms in addition to 7 recent predictions of W3 algorithm. Because there is more chance of an outbreak if there were some alerts in recent days. Considering these 12 features, we created a training dataset of 4098 instances with the contamination scale of 10^{-6} (low contamination) and we grew a decision tree without cross-validation. We used MATLAB statistical toolbox to develop decision trees. We used Gini Diversity Index as the optimization criterion. The accuracy of tree with those 12 features was 0.7970. Table 5.4 compares the accuracy of outbreak detection of developed decision trees using different feature sets.

We also trained several decision trees to select the feature set which leads to the highest accuracy of prediction. In later feature sets, we assumed some statistics that show how many algorithms alerted in recent days (i.e. number of votes). The accuracy of a model including 3 recent number of votes in addition to 12 features above was 0.8804. The accuracy of a model including 5 recent

number of votes and 12 features was 0.9078.

The model trained by 19 features, 7 recent number of votes and 12 features predicted the outbreaks with the accuracy of 0.9153. We grew a tree using 12 features, 7 recent number of votes and predictions of C1, C2, C3, W2 and W3 algorithm. The accuracy was 0.9112. Finally, we grew a tree using 7 recent number of votes. The accuracy was 0.9017.

The feature selection process shows that considering the number of votes in recent days increased the accuracy of prediction by decision tree. Also, excluding the predictions of W3 algorithm from feature set did not improve the accuracy. The highest accuracy was obtained when the feature set consisted of 19 features, 7 recent number of votes, 7 recent predictions of W3 algorithm, and predictions of C1, C2, C3, W2 and W3 algorithm. This feature set was used in our future experiments.

5.2.4 Developing a Decision Tree

The training data we used for training decision trees was based on the predictions of detection algorithms for every single day in the time series. For every instance, the distance from the onset of the outbreak is known, so, given the cost estimator function, the cost of missing outbreak per instance can be calculated. We used this cost as the cost of misclassifying the instance as a "non-outbreak day" if the instance belongs to "outbreak days" class. The cost of misclassifying a "non-outbreak day" as an "outbreak day" is a fraction of the average cost of missing outbreak; because the cost of a false alarm is usually less than the cost of missing outbreak. The objective of the developed decision tree is to minimize the number of misclassified cases and minimize the cost of misclassification.

We used MATLAB statistical toolbox to develop decision trees. We used Gini Diversity Index as the optimization criterion. We used a training set of 1000 instances to learn the decision tree knowing the misclassification cost of instances. We tested the tree with 3000 unseen instances whose misclassification cost is unknown. Then we calculated the total cost for misclassified instances.

We evaluated the accuracy and the total cost of misclassification of the developed model compared with a grown regular decision tree which did not take th e misclassification cost into account. Figure 5.3 shows a part of the regular decision tree and Figure 5.4 shows the same levels of the developed decision tree considering misclassification cost. None of the trees were pruned.

In the evaluation of regular decision tree, the accuracy of prediction for the training data was 0.889. For testing data, the accuracy of prediction was 0.7197. According to cost function approx-

Table 5.4 Accuracy of outbreak detection by different feature sets

Features	Accuracy	Confusion matrix
Predictions of C1, C2, C3, W2, W3, 7 recent predictions of W3	0.7970	$\begin{bmatrix} 2643 & 213 \\ 619 & 623 \end{bmatrix}$
Predictions of C1, C2, C3, W2, W3, 7 recent predictions of W3, 3 recent number of votes	0.8804	$\begin{bmatrix} 2706 & 150 \\ 340 & 902 \end{bmatrix}$
Predictions of C1, C2, C3, W2, W3, 7 recent predictions of W3, 5 recent number of votes	0.9708	$\begin{bmatrix} 2728 & 128 \\ 250 & 992 \end{bmatrix}$
Predictions of C1, C2, C3, W2, W3, 7 recent predictions of W3, 7 recent number of votes	0.9153	$\begin{bmatrix} 2726 & 130 \\ 217 & 1025 \end{bmatrix}$
Predictions of C1, C2, C3, W2, W3, 7 recent number of votes	0.9112	$\begin{bmatrix} 2730 & 126 \\ 238 & 1004 \end{bmatrix}$
7 recent number of votes	0.9017	$\begin{bmatrix} 2703 & 153 \\ 250 & 992 \end{bmatrix}$

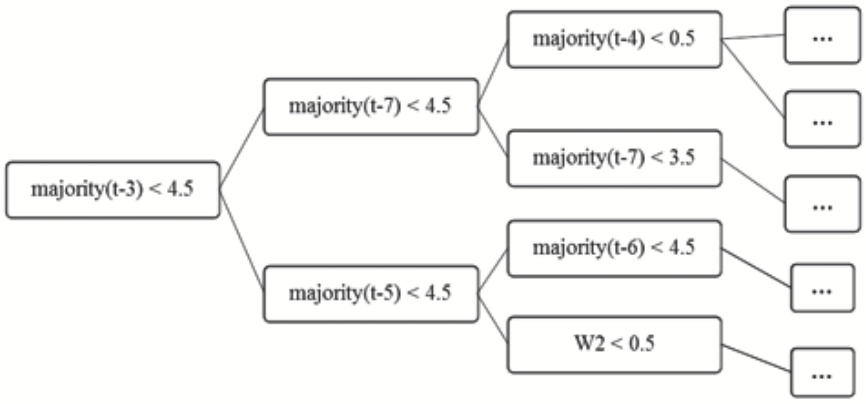

Figure 5.3 Part of regular decision tree

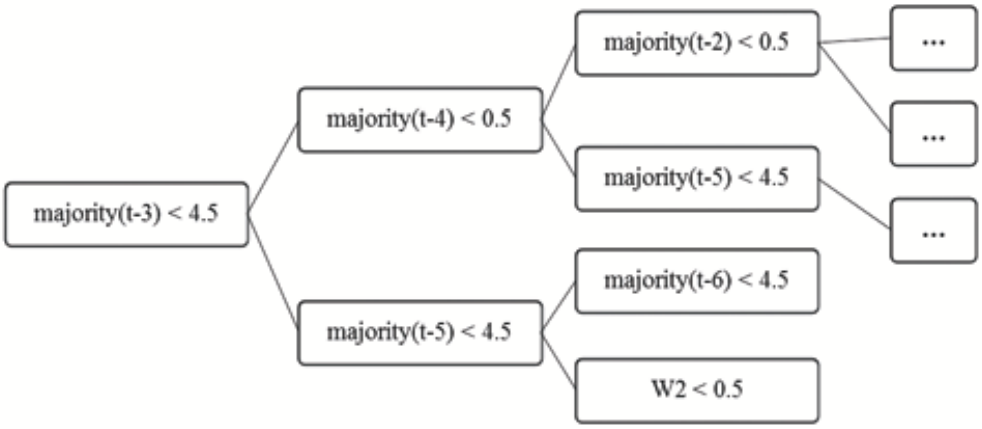

Figure 5.4 Part of decision tree which minimizes the misclassification cost

imation, the total cost of misclassified instances was 8.4750e+09.

The accuracy of prediction by the decision tree that takes misclassification cost into account for the training data was 0.952. For testing data, the accuracy of prediction was 0.7217. According to cost function approximation, the total cost of misclassified instances was 8.2806e+09.

We also evaluated the accuracy and the cost of statistical outbreak detection algorithms and majority of votes on testing data in Table 5.5. An alert is issued based on the majority of votes of yesterday if the number of positive votes is more than half of voters (i.e. 3 algorithms).

Next, we learnt the decision tree from time series with low contamination. We tested the developed tree with a testing data including 4758 instances with high contamination (scale = 10^{-4}). The accuracy of prediction was 0.7341. The total misclassification cost was 1.1954e+10. We also evaluated the performance of individual detection algorithms on high contamination time series in Table 5.6. The last row of the table shows a decision tree learnt from high contamination time series which considers the misclassification cost. The last column of the Table assumes that an alert cannot be triggered every day and it is postponed to even days, for example if an outbreak is detected on day 11, an alert is issued on day 12. Based on this assumption, the alert days are day 10, 12, 14, 16, 18, etc. This assumption follows the assumptions of simulated data. In this case, obviously, the cost of missing outbreaks is higher.

5.3 Discussion and Conclusion

In this chapter, we developed a cost-sensitive classifier for outbreak detection which is not only sensitive to the cost of misclassification, but also considers a different misclassification cost for

Table 5.5 Performance evaluation of outbreak detection algorithms

Algorithm	Accuracy	Confusion matrix	Estimated cost (x10^9)
C1	0.6153	$\begin{bmatrix} 1601 & 514 \\ 640 & 245 \end{bmatrix}$	11.123
C2	0.6113	$\begin{bmatrix} 1580 & 535 \\ 631 & 254 \end{bmatrix}$	11.117
C3	0.4983	$\begin{bmatrix} 1039 & 1076 \\ 429 & 456 \end{bmatrix}$	11.837
W2	0.6673	$\begin{bmatrix} 1837 & 278 \\ 720 & 165 \end{bmatrix}$	10.642
W3	0.5763	$\begin{bmatrix} 1397 & 718 \\ 553 & 332 \end{bmatrix}$	11.232
Majority voting	0.611	$\begin{bmatrix} 1564 & 506 \\ 661 & 269 \end{bmatrix}$	11.545
Cost-sensitive decision tree	0.7217	$\begin{bmatrix} 1709 & 367 \\ 468 & 456 \end{bmatrix}$	8.2806
Regular decision tree	0.7197	$\begin{bmatrix} 1722 & 354 \\ 487 & 437 \end{bmatrix}$	8.4750

Table 5.6 Performance evaluation of outbreak detection algorithms on time series with high contamination

Algorithm	Accuracy	Confusion matrix	Estimated cost (x10^9)	Cost ($x10^9$, alerting next day)
C1	0.6761	$\begin{bmatrix} 2572 & 700 \\ 841 & 645 \end{bmatrix}$	15.225	15.335
C2	0.6873	$\begin{bmatrix} 2553 & 719 \\ 769 & 717 \end{bmatrix}$	14.498	14.601
C3	0.5553	$\begin{bmatrix} 1407 & 1865 \\ 251 & 1235 \end{bmatrix}$	14.317	14.351
W2	0.7671	$\begin{bmatrix} 2942 & 330 \\ 778 & 708 \end{bmatrix}$	12.226	12.327
W3	0.6877	$\begin{bmatrix} 2109 & 1163 \\ 323 & 1163 \end{bmatrix}$	10.910	10.951
Majority voting	0.7007	$\begin{bmatrix} 2603 & 669 \\ 755 & 731 \end{bmatrix}$	13.852	13.950
Decision tree learnt from low contamination	0.7341	$\begin{bmatrix} 2719 & 553 \\ 712 & 774 \end{bmatrix}$	11.954	-
Decision tree learnt from high contamination	**0.9617**	$\begin{bmatrix} 3221 & 51 \\ 131 & 1355 \end{bmatrix}$	**2.2679**	-

every instance of data. In evaluation part, we showed that the total cost of the developed classifier is less than the cost of C1, C2, C3, W2, W3, and majority of votes algorithms. However, those algorithms are not able to optimize the cost of outbreaks directly. In addition, the accuracy of outbreak detection of cost-sensitive decision tree is competitive with statistical algorithms. We also tested the developed approach in the case that no prior knowledge about the contamination level was provided. The results showed the high performance of the model both in accuracy of detection and minimizing the total cost.

The developed approach used an approximated function to calculate the cost of missing outbreaks based on a study however, the function can be improved by modifying the cost parameters. Moreover, using a fraction of the cost of missing outbreaks as the cost of false alarms can be replaced by a more accurate estimation. Modifying the cost function will not affect the development of the cost-sensitive decision tree.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Contributions

In this thesis, we tackled three problems in disease outbreak detection applying Machine Learning methods. In this chapter, we summarize the results, conclusions, and possible future works.

We used simulated outbreak data overlaid on real healthcare utilization data to learn a Bayesian network model to predict detection performance as a function of outbreak characteristics and surveillance system parameters. This model can predict the performance metrics of commonly used outbreak detection methods with high accuracy. The model can also quantify the influence of different outbreak characteristics and detection methods parameters on detection performance in a variety of practically relevant surveillance scenarios. In addition to identifying outbreak characteristics expected to have a strong influence on detection performance, the model suggests the role for different algorithm features.

We investigated how outbreak detection methods can be combined in order to improve the overall detection performance. We used Hierarchical Mixture of Experts (HME), a probabilistic model for combining classification methods which has been well-studied in computer science and statistics. We used simulated surveillance data to train a HME in order to aggregate predictions from several outbreak detection methods. The developed HME algorithm was competitive to the best detection algorithm in the experiments. The developed detection algorithm based on HME was more robust under different circumstances and the level of contamination of surveillance time series did not influence the relative performance of the HME.

We analyzed some commonly used outbreak detection methods regarding to the cost of missing outbreaks and false alarms using the simulated outbreak data overlaid on real healthcare utilization data. We estimated the total cost of missing outbreaks and false alarms in addition to the accuracy of outbreak detection based on a study on cost analysis. We fitted a polynomial regression function to estimate that cost based on delayed time in detecting outbreaks. Then, we developed a cost-sensitive decision tree which predicts the outbreak based on the prediction of commonly used detection methods while minimizes the total cost of outbreaks. Based on the experimental results, using the developed cost-sensitive decision tree decreases the total cost of outbreak while maintaining the accuracy of outbreak detection at levels competitive with commonly used methods.

6.2 Future Work

Our approach in predicting detection performance by Bayesian network model can be extended to allow a coherent evaluation of new algorithms and new data sources as needed. In particular, using our current model, we can evaluate outbreak detection performance for the algorithms different than the C, W, and Adaptive Poisson algorithms which use the same parameters. Also, any configuration of considered parameters in our model different than the ones belonging to these algorithms can be thought of as a new detection method and can also be evaluated.

We presented several scenarios of outbreaks and desired performance, and used inference to suggest the best algorithm and parameter setting to use, as well as to quantify the expected performance. These scenarios are by no means exhaustive, and are meant as examples of what kinds of inference can be performed. Of course, in order to use this model as a tool for what-if analysis in the public health sector, an adequate interface would also need to be developed, but this goes beyond the scope of our work.

We used emergency department visits as the baseline time series for outbreak detection. In recent years, non-traditional data sources have been introduced in public health and surveillance systems. This includes mobile phone data (Buckee et al., 2013), social data (Alasaad et al., 2013), micro-blogging (Donelle and Booth, 2012), Twitter feeds and Google search queries (Ginsberg et al., 2008). While introducing and combining new data sources, especially in the era of big data, are promising directions for research in biosurveillance systems, the evaluation of their relevance and significance will be extremely important. Evaluation studies such as (Wilson et al., 2008) are needed to compare these new data sources to the existing ones, and the approach that we describe could be used to consider the relative contribution to detection performance of data sources and algorithms. To extend the model for new sources of data, the methodology of learning the Bayesian network model will not change.

A number of extensions to this work may improve the generalizability of the results. We used simulated outbreaks superimposed on real surveillance data; therefore, the results are affected by the quality of the simulation. Our approach can be extended by using real surveillance data and including more detection methods. Spatio-temporal data simulations, as well as using additional health care utilization information in addition to ED visits, should be studied as well.

This practice was an example of applications of Bayesian network models which can be used in other domains to reveal te influence of different variables, find the optimal configurations, and predict the behaviour of systems in unseen scenarios.

In the aggregating model, we use the predictions of C and W family detection algorithms, however, the models can be extended to consider the predictions of other detection algorithms, like Negative binomial CUSUM and Poisson Regression, which may provide more diversity of predictions.

We developed the aggregating method based on temporal surveillance data. We can extend our work to consider different sources of surveillance data, like spatiotemporal surveillance data to detect outbreak event in several regions. We can also use other sources of surveillance data, like school and work absenteeism rates and others.

In the developed HME, the predictions of detection algorithms were fed to all the experts of the structure. The model can be improved by assuming that each detection algorithm is an expert in the HME architecture and its vote is weighted based on determinants of detection performance (e.g. outbreak characteristics, desired false alarm rate). Moreover, we showed the usefulness of HME when the predictions of classifiers are not robust in practice. This structure can help in aggregating classifiers in other problems as well.

To estimate the cost, we used a model developed through an extensive literature study and an approximation based on interpolation since exact estimations of all possible classifications are expensive to obtain. This interpolation cost model can be improved further. Moreover, we used a fraction of the cost of missing outbreaks as the cost of false alarms; this can be replaced by a more accurate estimation. Modifying the cost function will not affect the development of the cost-sensitive decision tree, as this is an input to the algorithm.

We developed a cost-sensitive classifier which considers the misclassification cost of every training instance. The idea can be applied to the problems in which, instead of a fixed misclassification cost per class, the cost is different for every instance. Also, the model can optimize any parameter of interest (e.g., cost of false alarms) rather than only optimizing the accuracy. We learned cost-sensitive decision trees in this case, however, any other classifier can be modified to consider the misclassification cost.

REFERENCES

S. Alasaad *et al.*, "War diseases revealed by the social media: massive leishmaniasis outbreak in the syrian spring," *Parasites & vectors*, vol. 6, no. 1, p. 94, 2013.

C. M. Bishop, *Pattern recognition and machine learning*. springer New York, Inc., 2006, vol. 4.

J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk, et C. E. Brodley, "Pruning decision trees with misclassification costs," dans *Machine Learning: ECML-98*. Springer, 1998, pp. 131–136.

J. Brillman, T. Burr, D. Forslund, E. Joyce, R. Picard, et E. Umland, "Modeling emergency department visit patterns for infectious disease complaints: results and application to disease surveillance," *BMC medical informatics and decision making*, vol. 5, no. 1, 2005.

C. O. Buckee, A. Wesolowski, N. N. Eagle, E. Hansen, et R. W. Snow, "Mobile phones and malaria: modeling human and parasite travel," *Travel medicine and infectious disease*, vol. 11, no. 1, pp. 15–22, 2013.

D. L. Buckeridge, A. Okhmatovskaia, S. Tu, M. O'Connor, C. Nyulas, et M. A. Musen, "Predicting outbreak detection in public health surveillance: Quantitative analysis to enable evidence-based method selection," dans *AMIA Annual Symposium Proceedings*, vol. 2008. American Medical Informatics Association, 2008, p. 76.

D. Buckeridge, "Outbreak detection through automated surveillance: a review of the determinants of detection," *Journal of Biomedical Informatics*, vol. 40, no. 4, pp. 370–379, 2007.

D. Buckeridge, A. Okhmatovskaia, S. Tu, M. O'Connor, C. Nyulas, et M. Musen, "Understanding detection performance in public health surveillance: modeling aberrancy-detection algorithms," *Journal of the American Medical Informatics Association*, vol. 15, no. 6, pp. 760–769, 2008.

D. Buckeridge, C. Jauvin, A. Okhmatovskaia, et A. Verma, "Simulation analysis platform (snap): a tool for evaluation of public health surveillance and disease control strategies," dans *AMIA Annual Symposium Proceedings*, vol. 2011. American Medical Informatics Association, 2011, p. 161.

H. Burkom, "Development, adaptation, and assessment of alerting algorithms for biosurveillance," *Johns Hopkins APL Technical Digest*, vol. 24, no. 4, pp. 335–342, 2003.

H. Burkom, S. Murphy, et G. Shmueli, "Automated time series forecasting for biosurveillance," *Statistics in Medicine*, vol. 26, no. 22, pp. 4202–4218, 2007.

R. Daly, Q. Shen, et S. Aitken, "Learning bayesian networks: approaches and issues," *The Knowledge Engineering Review*, vol. 26, no. 02, pp. 99–157, 2011.

A. Dempster, N. Laird, et D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," dans *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 155–164.

L. Donelle et R. Booth, "Health tweets: An exploration of health promotion on twitter," *OJIN: The Online Journal of Issues in Nursing*, vol. 17, no. 3, 2012.

R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

C. Elkan, "The foundations of cost-sensitive learning," dans *International joint conference on artificial intelligence*, vol. 17, no. 1. Citeseer, 2001, pp. 973–978.

T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine Learning*, vol. 31, pp. 1–38, 2004.

R. Fricker, "Some methodological issues in biosurveillance," *Statistics in Medicine*, vol. 30, pp. 403–415, 2011.

J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, et L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2008.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

L. Hutwagner, W. Thompson, G. Seeman, et T. Treadwell, "The bioterrorism preparedness and response early aberration reporting system (ears)," *Journal of Urban Health: Bulletin of the New York Academy of Medicine*, vol. 80, no. Supplement 1, pp. i89–i96, 2003.

L. Hutwagner, T. Browne, G. Seeman, et A. Fleischauer, "Comparing aberration detection methods with simulated data," *Emerg Infect Dis*, vol. 11, no. 2, pp. 314–316, 2005.

M. Izadi, D. Buckeridge, A. Okhmatovskaia, S. W. Tu, M. J. O'connor, C. Nyulas, et M. A. Musen, "A bayesian network model for analysis of detection performance in surveillance systems," dans *AMIA Annual Symposium Proceedings*, vol. 2009. American Medical Informatics Association, 2009, p. 276.

M. Izadi et D. Buckeridge, "Decision theoretic analysis of improving epidemic detection," dans *AMIA Annual Symposium Proceedings*, vol. 2007. American Medical Informatics Association, 2007, pp. 354–358.

M. L. Jackson, A. Baer, I. Painter, et J. Duchin, "A simulation study comparing aberration detection algorithms for syndromic surveillance," *BMC Medical Informatics and Decision Making*, vol. 7, no. 1, p. 6, 2007.

R. Jacobs, M. Jordan, S. Nowlan, et G. Hinton, “Adaptive mixtures of local experts,” *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

M. Jordan et R. Jacobs, “Hierarchical mixtures of experts and the em algorithm,” *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

R. Kindermann, J. L. Snell *et al.*, *Markov random fields and their applications*. American Mathematical Society Providence, RI, 1980, vol. 1.

D. Koller, N. Friedman, L. Getoor, et B. Taskar, “Graphical models in a nutshell,” *Introduction to statistical relational learning*, pp. 13–55, 2007.

——, “2 graphical models in a nutshell,” *STATISTICAL RELATIONAL LEARNING*, p. 13, 2007.

S. L. Lauritzen et D. J. Spiegelhalter, “Local computations with probabilities on graphical structures and their application to expert systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 157–224, 1988.

B. Lewis, S. Eubank, A. M. Abrams, et K. Kleinman, “in silico surveillance: evaluating outbreak detection with simulation models,” *BMC medical informatics and decision making*, vol. 13, no. 1, p. 12, 2013.

J. Lombardo et D. Buckeridge, *Disease surveillance: a public health informatics approach*. Wiley-Blackwell, 2007.

T. Mitchell, *Machine learning*, sér. McGraw-Hill series in computer science. McGraw-Hill, 1997. [En ligne]. Disponible: http://books.google.ca/books?id=diu9QgAACAAJ

J. Nelder et R. Wedderburn, “Generalized linear models,” *Journal of the Royal Statistical Society. Series A (General)*, pp. 370–384, 1972.

Norsys, “Netica bayesian network software from norsys,” 1995. [En ligne]. Disponible: http://www.norsys.com

A. Okhmatovskaia, A. D. Verma, B. Barbeau, A. Carriere, R. Pasquet, et D. L. Buckeridge, “A simulation model of waterborne gastro-intestinal disease outbreaks: Description and initial evaluation,” dans *AMIA Annual Symposium Proceedings*, vol. 2010. American Medical Informatics Association, 2010, p. 557.

L. Olshen et C. J. Stone, “Classification and regression trees,” *Wadsworth International Group*, 1984.

J. Park, H. Tyan, et C. Kuo, “Internet traffic classification for scalable qos provision,” dans *Multimedia and Expo, 2006 IEEE International Conference on*. IEEE, 2006, pp. 1221–1224.

J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

V. S. Sheng et C. X. Ling, "Thresholding for making classifiers cost-sensitive," dans *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 476.

K. M. Ting, *Inducing cost-sensitive trees via instance weighting*. Springer, 1998.

J. I. Tokars, H. Burkom, J. Xing, R. English, S. Bloom, K. Cox, et J. A. Pavlin, "Enhancing time-series detection algorithms for automated biosurveillance," *Emerging infectious diseases*, vol. 15, no. 4, p. 533, 2009.

J. Tokars, H. Burkom, J. Xing, R. English, S. Bloom, K. Cox, et J. Pavlin, "Enhancing time-series detection algorithms for automated biosurveillance," *Emerging Infectious Diseases*, vol. 15, no. 4, pp. 533–539, 2009.

S. Viaene et G. Dedene, "Cost-sensitive learning and decision making revisited," *European journal of operational research*, vol. 166, no. 1, pp. 212–220, 2005.

M. M. Wagner et G. Wallstrom, "Methods for algorithm evaluation," *Handbook of biosurveillance*, pp. 301–310, 2006.

M. M. Wagner, G. L. Wallstrom, et A. Onisko, "Issue a boil-water advisory or wait for definitive information? a decision analysis," dans *AMIA Annual Symposium Proceedings*, vol. 2005. American Medical Informatics Association, 2005, p. 774.

M. Wagner, A. Moore, et R. Aryel, *Handbook of biosurveillance*. Academic Press, 2006.

S. Wallenstein et J. Naus, "Scan statistics for temporal surveillance for biologic terrorism," *MMWR Morb Mortal Wkly Rep*, vol. 53, no. Suppl, pp. 74–78, 2004.

R. E. Watkins, S. Eagleson, R. G. Hall, L. Dailey, et A. J. Plant, "Approaches to the evaluation of outbreak detection methods," *BMC public health*, vol. 6, no. 1, p. 263, 2006.

N. Wilson, K. Mason, M. Tobias, M. Peacey, Q. Huang, et M. Baker, "Interpreting google flu trends data for pandemic h1n1 influenza: the new zealand experience." *Euro surveillance: bulletin européen sur les maladies transmissibles= European communicable disease bulletin*, vol. 14, no. 44, pp. 429–433, 2008.

J. Xing, H. Burkom, et J. Tokars, "Method selection and adaptation for distributed monitoring of infectious diseases for syndromic surveillance," *Journal of Biomedical Informatics*, vol. 44, pp. 1093–1101, 2011.