INTEGRATED APPROACH TO REAL-TIME BIOSURVEILLANCE IN A FEDERATED DATA SOURCE ENVIRONMENT

A THESIS SUBMITTED TO

THE DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING, UNIVERSITY OF MANITOBA

WINNIPEG, MANITOBA R3T 5V6 CANADA

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Shamir N. Mukhi, BSc, MASc, PEng, MIEEE

August, 2007

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

INTEGRATED APPROACH TO REAL-TIME BIOSURVEILLANCE IN A FEDERATED DATA SOURCE ENVIRONMENT

BY

Shamir N. Mukhi

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of

Manitoba in partial fulfillment of the requirement of the degree

DOCTOR OF PHILOSOPHY

Acknowledgement

Many thanks to Dr Robert McLeod for his support and guidance throughout the course of my research. His guidance, direction and insight into various engineering concepts and approaches have provided me with opportunity to learn and apply diverse knowledge base to my research area. Sincere thanks to Drs. Galit Shmueli, Attahiru Alfa and Vojislav Misic for their insight and valuable feedback. I would like to convey special thanks to Drs. Amin Kabani and Jeff Aramini for mentoring me with issues related to Public Health and Epidemiology.

I would like to thank my family for their support throughout the four years and putting up with all the late nights. To my wife Nushina, my son Zia and my daughter Hannah, thank you.

ABSTRACT

INTEGRATED APPROACH TO REAL-TIME BIOSURVEILLANCE IN A FEDERATED DATA SOURCE ENVIRONMENT

Shamir N. Mukhi, BSc, MASc, PEng, MIEEE
PhD in Electrical and Computer Engineering, University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
Supervisor: Dr. R. McLeod
Committee: Dr. McLeod, Dr. Alfa, Dr. Misic and Dr. Aramini
August, 2007

Health surveillance can be viewed as an ongoing collection, analysis, interpretation and dissemination of health related data for use in design, development, deployment, and evaluation of a given health system in multiple spheres (ex: animal, human, environment). As we move into a sophisticated technologically advanced era, there is a need for cost-effective and efficient health surveillance methods and systems that will rapidly identify potential bioterrorism attacks and infectious disease outbreaks. The main objective of such methods and systems would be to reduce the impact of an outbreak by enabling appropriate officials to detect it quickly and implement timely and appropriate interventions. Identifying an outbreak and/or potential bioterrorism attack days to weeks earlier than traditional surveillance methods would potentially result in a reduction in morbidity, mortality, and outbreak associated economic consequences. Proposed here are methods for achieving an integrated health surveillance system. Specifically, the research looks at a framework that would enable a user and/or a system to interpret the anomaly detection results generated via multiple aberration detection algorithms with some indication of confidence. Ideally, a framework takes into account the relationships between algorithms and produces an unbiased confidence measure for identification of start of an outbreak.

Keywords: Health, disease surveillance, outbreak, bioterrorism, federated sources, aberration detection, syndromic, infectious disease.

Contents

List of Figures

Acronymns

- **AOI**: Areas of Interest
- **ARTIST**: Architecture for Real-time Information Standardization and Transformatio
- **BioSTORM**: Biological Spatio-Temporal Outbreak Reasoning Model
- **CAIF**: Confidence-based Aberration Interpretation Framework
- **CDC**: Centers for Disease Control and Prevention
- **CEWS**: Canadian Early Warning System
- **COHORT**: Composite Occupational Health and Operational Risk Tracking
- **CUSUM**: Cummulative Sums
- **EARS**: Early Aberration Reporting System
- **EDI**: Electronic Data Interchange
- **ESSENCE**: Electronic Surveillance System for the Early Notification of Community-based Epidemics
- **EWMA**: Exponential Weighted Moving Average
- **FSA**: Forward Sortation Area
- **FTP**: File Transfer Protocol
- **GIS**: Geographical Information System
- **HL7**: Health Level 7
- **HTTP(S)**: Hypertext Transfer Protocol (Secure)
- **ICD**: International Classification of Disease
- **ISIS**: Infectious Disease Surveillance Information System

- **LDA**: Linear Discriminant Analysis
- **LEADERS**: Lightweight Epidemiological Advanced Detection Emergency Response System
- **LOINC**: Logical Observation Identifiers Names and Codes
- **MA**: Moving Average
- **NRDM**: National Retail Data Monitor
- **OTC**: Over The Counter
- **PANDA**: Patient-based ANomaly Detection and Assessment
- **RLS**: Recursive Least Squares
- **RODS**: Real time Outbreak and Disease Surveillance
- **SCS**: Source Classification Spatial hierarchy
- **SNOMED**: Systemized Nomenclature of Medicine
- **SOAP**: Simple Object Access Protocol
- **SOC**: Significance Of Contribution graph
- **SSIC**: Syndromic Surveillance Information Collection
- **SQL**: Structured Query Language
- **SVM**: Support Vector Machines
- **UPC**: Universal Product Code
- **VPN**: Virtual Private Network
- **WMA**: Weighted Moving Average
- **WSARE**: What's Strange About Recent Events
- **WSDL**: Web Services Description Lanugage
- **XML**: eXtensible Markup Language

Chapter 1

Introduction

As we move into an even more technologically advanced era, there is a need and an opportunity for a cost-effective and efficient health surveillance system that will rapidly identify bioterrorism attacks and infectious disease outbreaks [1]. The main objective of such a system is to reduce the impact of an outbreak by enabling appropriate officials to detect it quickly and implement timely, appropriate interventions. Identifying an outbreak and/or bioterrorist attack days to weeks earlier than traditional surveillance will result in a reduction in morbidity, mortality, and its economic consequences.

Fig. 1.1 illustrates a typical outbreak curve that is expected for number of cases immediately after a potential bioterrorist attack. As shown, there are three main points of interest labeled as A, B and C. Detection of an outbreak at point A is optimal and thus is referred to *early detection.* The number of cases are on the rise and hence detecting an outbreak at this point could be very effective and

[1] An outbreak in simple sense may be defined as a sudden increase in number of cases of a specific disease target typically confined to a small geographical area.

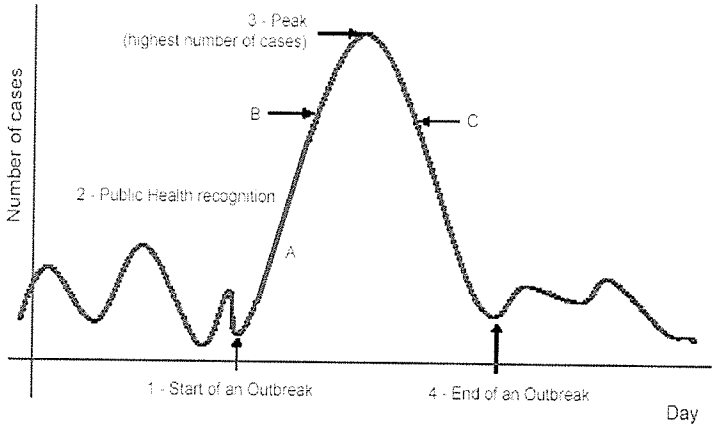

Figure 1.1: Points of Interest in an Outbreak curve

it can potentially aid in minimizing the impact of a potential bioterrorist attack. Point B illustrates increasing number of cases during an already established outbreak, however, it might not be too late if an outbreak is detected at this point. Finally, point C is location on the curve illustrating decreasing number of cases during the tail-end of an outbreak. Although traditional public health surveillance systems would like to identify an outbreak early on (at point A), typically they detect an outbreak in between points A and B (identified as 2), which is not really an ideal location on the curve. Numerous researchers are developing techniques and algorithms to assist public health to shorten the time difference between points A and 2. This is the *gain* that we need to strive to minimize (if not completely alleviate) in devising future surveillance systems.

1.1 Why do we need Surveillance? - A Real Scenario

The first typical symptoms from many of the agents with a high probability of use for bioterrorist activity mimic those of influenza [1]. An effective influenza surveillance program is in unique position of being able to identify an outbreak of suspicious illness, possibly even a biological terror event. When presented with flu-like symptoms, a high incidence of positive influenza test results could indicate an influenza outbreak. Conversely, a high incidence of negative results might indicate a suspicious illness. An international influenza epidemic of 1918-1919 infected 20% to 40% of the world's population [2, 3, 4]. More than 20 million people died in less than a year, about 500,000 of them in the United States alone. More people died during this influenza pandemic than during World War I or the Bubonic Plague (also known as Black Death that occured between 1347 and 1351). It took the influenza virus of 1918 only three months to spread across the world. It has been cited as the most devastating epidemic in recorded world history. Known as the *Spanish Flu* or *La Grippe*, this influenza pandemic was a global disaster and not totally independant of events of World War I with its considerable movements of populations and groups of people.

In February 1957, the Asian influenza pandemic was first identified in the Far East [5]. Immunity to this strain was rare in people less than 65 years of age, and a pandemic was predicted. In preparation, vaccine production began in late May 1957, and health officials increased surveillance for flu outbreaks. Unlike the virus that caused the 1918 pandemic, the 1957 pandemic virus was quickly identified due to past experiences. Vaccine was available in limited supply by August 1957.

The virus came to the United States, with a series of small outbreaks over the summer of the same year. By December 1957, the worst seemed to be over. However, during January and February 1958, there was another wave of illness among the elderly. This is an example of the potential *second wave* of infections that can develop during a pandemic. The disease infects one group of people first, infections appear to decrease and then infections increase in a different part of the population.

The Hong Kong influenza pandemic was first detected in early 1968 [5]. The first cases in the United States were detected as early as September of that year, but illness did not become widespread until December. The number of deaths between September 1968 and March 1969 for this pandemic was 33,800 in the United States, making it the mildest pandemic in the 20th century. There could be several reasons why fewer people died due to this virus. First, the Hong Kong flu virus was similar in some ways to the Asian flu virus that circulated between 1957 and 1968. Earlier infections by the Asian flu virus might have provided some immunity against the Hong Kong flu virus and this may have helped to reduce the severity of illness during the Hong Kong pandemic. Second, instead of peaking in September or October, like the pandemic influenza had in the previous two pandemics, this pandemic did not gain momentum until near the school holidays in December. Since children were at home, the rate of influenza transmission among school children and their families declined. Third, improved medical care and antibiotics that were more effective for secondary bacterial infections were available for those who became ill.

About 10 years later in May 1977, influenza A/H1N1 virus was isolated in northern China. It spread rapidly and caused epidemic disease in children and

young adults worldwide [5]. This virus was similar to other A/H1N1 viruses that had circulated prior to 1957. Persons born before 1957 were likely to have been exposed to A/H1N1 and to have developed some immunity. Therefore, when A/H1N1 reappeared in 1977, many people over the age of 23 had some protection against the virus. By January 1978, the virus had spread around the world. Because illness occurred primarily in children, this event was not considered a true pandemic.

The most recent pandemic *scares* occurred in 1997, 1999 and 2003. In 1997, at least a few hundred people became infected with the avian A/H5N1 flu virus in Hong Kong and six out of 18 people who were hospitalized died [6]. Fortunately, this virus did not appear to be transmissible from person to person. To prevent the spread of this virus, all chickens (approximately 1.5 million) in Hong Kong were slaughtered. In 1999, a new avian flu virus - A/H9N2 - was identified that caused illnesses in two children in Hong Kong [7]. Although both of these viruses have not gone on to start pandemics, their continued presence in birds, their ability to infect humans, and the ability of influenza viruses to change and become more transmissible among people is an ongoing concern.

In 2003, the world witnessed a disease that was spreading from person to person and continent to continent. This illness was soon identified as Severe Acute Respiratory Syndrome (SARS), which is considered to be the *first severe and readily transmissible new disease to emerge in the twenty-first century* [8]. According to [9], within 4 months after the first global alert about the new disease, all known chains of transmission had been interrupted in an outbreak that affected 27 countries. As of July 11, 2003 in its daily summary [10], the World Health Organization reported 8,437 probable cases of SARS and 813 deaths worldwide,

and the toll has since risen to about 900 as some previously-ill individuals have succumbed.

While previous influenza pandemics were naturally occurring events, an influenza pandemic could be started with an intentional release of a deliberately altered influenza strain [11]. An influenza virus, especially one genetically manipulated for increased virulence, would be an attractive weapon for bioterrorists, according to physicians writing in the Journal of the Royal Society of Medicine [12]. As a potential biological weapon, influenza has several advantages over smallpox, including ready accessibility. According to [1], since the terrorist attacks of 2001, *We must . . . consider the possibility of malicious genetic engineering to create more virulent strains. Sequencing of the genome of the 1918 Spanish influenza virus is nearly complete; once it is published, unscrupulous scientists could presumably utilize candidate virulence sequences.* Influenza is usually transmitted by direct contact, but aerosol transmission also occurs and requires *27,000 times fewer virions to induce equivalent disease*, the article states.

Influenza virus is far more accessible than Smallpox virus, known stocks of smallpox are held only in secure facilities. As mentioned in [13], Smallpox is a serious, contagious, and sometimes fatal infectious disease. It is classified as a Category A agent by the Centers for Disease Control and Prevention. Category A agents are believed to pose the greatest potential threat for adverse public health impact and have a moderate to high potential for large-scale dissemination. The last naturally occurring case in the world was in Somalia in 1977. Influenza has other potential advantages over Smallpox as a weapon:

- Since influenza occurs naturally, an epidemic would have more time to

spread before triggering a major public health response.

- Post-exposure immunization is ineffective for influenza because of its short incubation period (one to four days); this is not the case with smallpox.
- Influenza is harder to eradicate than smallpox because it has animal and avian reservoirs.

In the list of potential bioterrorist agents, influenza is classified as a category C agent: *Emerging pathogens that could be engineered for mass dissemination in the future because of availability; ease of production and dissemination; and potential for high morbidity and mortality rates and major health impact* [14].

1.2 Canadian Public Health Infostructure

In October 2003, the Health Canada advisory committee on SARS and public health published the, *Naylor Report* [15] that provides recommendations for a public health infostructure, which is intended to increase capability in three major areas including *disease surveillance*, *outbreak management* and *emergency response*.

The report includes guidelines for the co-ordination of various government and non-government organizations, their roles, and their application of information technology for outbreak surveillance. It also discusses legislation which governs the collection and disclosure of private health information. It introduces existing laws for ensuring individual privacy and their implications for disease surveillance. The report suggests enacting a new *Canada Health Protection Act* which would

form the legislative basis for disease surveillance system to collect identifiable personal health information.

The proposed Canada Health Protection Act outlines principles that would govern the collection and use of identifiable information. In addition, the proposed act also includes a proportionality principle that requires: collecting or disclosing as little identifying information as is required in order to achieve the public health objective; conversion to de-identified data as soon as possible and limiting access to identifiable personal health information; prohibiting the use of identifiable personal health information to make decisions about an individual in other contexts.

A health surveillance system requires collection of vast amounts of personal health information for potential record-linkage exercise in case of a true outbreak and thus need for patient traceability. It has been understood from the scenario of the SARS outbreak that very few abnormal cases are reported at an early stage. Personal information of individuals involved in those few cases can be a very critical pieces of information in detecting an outbreak at a very early stage. Human intervention is typically required to do such patient record-linkage due to privacy regulations. Information technology systems must seamlessly enable such linkage capability by potentially setting up automated processes that could be invoked during a real need.

1.3 Need for Early Detection

Increasing frequency of biological crisis, both natural (e.g. SARS, BSE) and intentional (e.g. anthrax), illustrate that health surveillance needs to be enhanced to meet the challenges facing today's society. One key issue is the lack of integration between different information systems that need to be accessed by various stakeholders for decision making, both front line health professionals and policy makers alike. Providing timely and accurate information requires contributions by the following main components among others: (1) data visualization (reports, trends and maps); (2) data quality and integrity; (3) data transfer systems; (4) data standardization systems; and (5) anomaly detection algorithms. One of the approaches that the research community has adopted is looking at diagnostic (ex: respiratory, gastro-intestinal) data using *syndromic surveillance* systems. It may be the ideal type of health surveillance to detect outbreaks whether they are intentional or naturally occurring events. Such systems monitor for disease by tracking symptom complexes that are representative of specific diseases being studied. The system looks for significant increases in the frequency of a given syndrome against a baseline and provides timely notification of any abnormal increase. The use of advanced technology in syndromic surveillance enhances the near *real time* detection and notification of outbreaks. A detailed view of some of the existing systems is presented in the later chapters.

Although considered a fairly recent approach, syndromic surveillance meets the need for a means to identify a significant health threat as early as possible and take steps to mitigate its effect. The recent outbreak of SARS in Asia is an example of an epidemic whose effects might have been more limited had it been detected earlier. Such disease outbreaks do not stop at geopolitical borders; they

affect people in neighboring jurisdictions, which underlines the need for public health entities to collaborate on systematically gathering and analyzing health data. The goal is to recognize disease patterns early and to notify health care workers and the public in order to safeguard public health and save lives.

To scan for possible outbreaks and/or bioterrorist attacks we need to, in addition to syndromic surveillance, investigate various types of data sources as well as different health systems (not just public health). In addition to clinical (diagnostic) data, we must look at pre-diagnostic data that describe health seeking behavior such as pharmaceutical over-the-counter (OTC) sales, weather, radiation data, and laboratory requisitions. Moreover, we must also consider health systems beyond public health including *animal* health and *environmental* health. Algorithms need to be developed to parse cross-relationships between these data sources to provide a comprehensive view at *health* of a given geographical area.

Another key component to *real-time* detection is the capability to extract data as soon as it enters the system. Most of the existing systems require manual data entry and/or automated routine data transfers. Due to the geographical setup of countries (esp. Canada), a much better way to get data from an already overworked clinical environment (hospitals and labs) is for the data to be gathered automatically. The clinician does not have to make the effort to create a manual public health report; instead, the report is generated and sent automatically as a by-product of daily workflow.

Most surveillance systems collect data from distributed geographic location into a central system. Data analysis or decision-making is accomplished at a central system. Local and regional data sources have to rely on the decisions made at the central system. This requires the system to be capable of close to real-time

algorithm execution and result management that facilitate quick access to results in a federated environment. Although this seems to be a common approach, with recent advances in communications related technologies, federated architectures that support distributed processing and analysis of data must also be considered.

1.4 Summary

In summary, an inter-disciplinary *health* surveillance approach needs to be embraced, where health officials, assisted by automated acquisition of data and generation of alarms using statistical tools, monitor disease indicators continually (real-time) or at minimum daily (aggregate) to detect outbreaks of diseases earlier and more completely than would otherwise be possible with traditional public health methods. Moreover, due to the hierarchical and disparate nature of the health system, various types of health-care facilities will have to interact and participate and thus generate massive amounts of data and result sets that need to be analyzed for anomalies and presented to the public health individuals for investigation and decision making.

There is a need for a software system to present trends and maps for diverse data types based on collecting disparate data as well as performing area-based parameterized anomaly detection and result management; a conceptual system, *architecture for real-time information standardization and transformation* (ARTIST), is proposed here that outlines major functional blocks within a typical real-time surveillance system. Moreover, due to vast amount of potential data from heterogeneous sources and availability of multiple anomaly detection algorithms, there is a definite need for an efficient algorithm or set of algorithms

to summarize the outcome of different anomaly detection algorithms applied to the same data source. A novel system that analyzes the results generated by execution of a set of algorithms to generate intelligence is proposed. This system, referred to as *confidence-based aberration interpretation framework* (CAIF), is based on understanding the relationships between the outcomes of various algorithms.

Furthermore, there is also a need to support existing centralized systems with a framework that can perform anomaly detection at the data providing source and in a distributed manner to respect privacy rules; this type of architecture will require further research work to investigate algorithms and processes to enable physical implementation which is beyond the scope of this research. However, effort will be made to identify further research areas to provide some guidance for future researchers interested in pursuing this further.

A detailed analysis of some of the existing surveillance systems and anomaly detection algorithms is necessary to form the basis for solving the three gaps as identified above. Chapter 2 provides a detailed literature review of some of the key surveillance systems that are in use today. A detailed review of analytical anomaly detection algorithms is provided in Chapter 3, which includes some of the most commonly used analytical algorithms by the existing systems. Chapter 4 presents a conceptual architecture for a typical real-time surveillance system that is based on studying various systems and provides a justification for the need of a framework that interprets anomaly detection outputs from various algorithms. The details of the proposed confidence-based anomaly interpretation framework are presented in Chapter 5, with corresponding simulation results discussed at

length in Chapter 6. Chapter 7 goes on further to compare the proposed framework with some of the commonly used techniques in pattern recognition. Chapter 8 presents a summary of the work done throughout the research.

Chapter 2

Review of Existing Surveillance Systems

Real-time public health data surveillance systems are becoming key tools to detect epidemics and initiate timely responses to outbreaks. An effective surveillance system would be able to detect the onset of any bioterrorism related or naturally occurring disease outbreak at an early stage. Such early warning can aid in effective mobilization of resources during an outbreak. The existing surveillance systems can be classified into three groups: clinician reporting based surveillance, laboratory based surveillance and syndromic surveillance.

Clinician reporting based surveillance is based on clinician reporting of disease counts after definitive testing and diagnosis. Traditional public health surveillance systems partially follow this approach, which requires waiting for confirmed results of diagnosis. This delay can seriously undermine the primary goal of initiating rapid response from public health departments.

Laboratory based surveillance depends on data collected directly from laboratories including requisition, test dates and confirmed dates. An example of such a system includes the Canadian National Enteric Surveillance Program (NESP) [16] which looks at weekly confirmed cases of enteric diseases across the country.

Syndromic surveillance refers to methods that collect and analyze data about syndromes from physicians prior to definitive diagnoses [17]. A syndrome is a recognizable complex of symptoms and signs, which indicate a specific disease or condition for which a direct cause is not necessarily understood. Such systems are also being used in monitoring of pharmaceutical data such as over-the-counter (OTC) sales of health care products [18, 19] and telephone triage [19, 20].

According to [21], the factors that make syndromic surveillance more effective over traditional surveillance include the following:

- Some diseases, such as Tularemia, have a short incubation period (the period between the exposure and the start of symptoms). The absence of clinical signs decreases the probability of clinical diagnosis. Some diseases require special tests or are not detectable through initial routine tests (e.g. Viral hemorrhagic fever).

- Syndromic surveillance systems can monitor non-traditional data sources such as grocery and over-the-counter medication sales, school absenteeism, or video/audio systems for monitoring coughs in public places [22, 23], among others. These non-traditional data sources can be helpful for early outbreak detection because the outbreak footprint is likely to exist in these data earlier than in medical or public health data.

Although, real-time syndromic surveillance is attractive and timely, significant concerns have been raised about the performance of some algorithms used in such systems [23, 24, 25].

- The studies in [21, 25, 26] show that some algorithms used in syndromic surveillance systems are susceptible to reporting false positives, that is, detect an anomaly during *peace* time. Most syndromic surveillance systems set their anomaly detection thresholds to be as sensitive as possible to minimize the risk of missing important events, producing frequent false alarms, which may be determined to be false positives by subsequent investigation. These systems face inherent trade-offs between timeliness and number of false positives. False positives have a negative impact on public health surveillance because they can lead to expensive resource utilization for further investigation and can cause undue concern among the general public.

- Syndromic surveillance systems can monitor data from multiple heterogeneous sources. These include traditional sources such as emergency department, hospital admission, and laboratories, as well as non-traditional sources such as retail stores, pharmacies, and schools. Therefore, a shared vocabulary (also known as *ontology*) is needed [27] for describing outbreak conditions so that different systems under different scenarios can be aggregated and studied.

- Depending on the number of data providers, there can be enormous amount of data collected and thus, a need for large number of algorithmic execution to detect potential anomalies. This creates a complexity in management of these executions so that results can be communicated in an efficient manner.

This section presents a summary of some of the major existing syndromic surveillance systems around the globe in terms of their objectives, data collection and analysis methods.

2.1 Electronic Surveillance System for the Early Notification of Community-based Epidemics (ESSENCE)

A syndromic surveillance system undertaken by Department of Defense, USA as part of its Global Emerging Infections Surveillance and Response System [28]. The primary objective of ESSENCE is to enable early detection disease outbreak due to bio-terrorist attacks in the Washington D.C. area. ESSENCE collects data from numerous emergency rooms and primary care clinics at multiple military treatment facilities. These data providers send diagnoses data of each patient encounter using the ICD-9-CM [1] codes. The system started collecting data electronically on a daily basis beginning in December 1999. ICD-9-CM codes are grouped into seven syndrome clusters for analysis: *Respiratory, Gastrointestinal, Fever, Dermatological, Hemorrhagic, Neurological and Coma.* ESSENCE builds a baseline level for each of these seven groups and monitors any fluctuations. It uses two main approaches for anomaly detection:

1. Linear regression is used for groups whose counts show weekly and seasonal variation.

[1] International Classification of Diseases, 9th Revision, Clinical Modification

2. Poisson regression is used for groups whose daily counts is much smaller, that do not show weekly trends.

ESSENCE generates a warning if the actual count crosses a threshold that corresponds to a 95% confidence interval of expected count. In addition to data analysis, ESSENCE also incorporates visualization of the data through geographic information system (GIS) tool which plots cases by patient home ZIP code. Such visualization would help observe outbreak trends in a region.

Another system extended from ESSENCE is ESSENCE-II [24]. It integrates analysis of clinical and non-clinical data sets for early detection of outbreaks. Clinical data include Emergency room syndromes, private-practice billing codes and veterinary syndromes, whereas non-clinical data include absenteeism, nurse hotline calls, sales of OTC medications. In addition to the data collection in the form of ICD-9 codes as in original ESSENCE system, ESSENCE-II collects data in the form of chief complaints. ESSENCE-II uses natural language processing methods to convert free-text chief complaints into syndrome groups. For outbreak detection, ESSENCE uses only temporal detection methods. However, ESSENCE-II has used both temporal and spatial detection methods. Temporal detection methods include Exponential Weighted Moving Average (EWMA) and methods used in the Early Aberration Reporting System (EARS) [29], whereas spatial detection methods use the Kulldroff scan statistic [30]. ESSENCE-II has described a set of steps to develop simulated outbreak scenarios to test its performance [31]. It presents an evaluation of detection performance based on using different sources of data.

2.2 Real time Outbreak and Disease Surveillance (RODS)

This is a public health surveillance system developed at the RODS laboratory of the Center for Biomedical Informatics at the University of Pittsburgh [32] that has been in operation in western Pennsylvania since 1999. Similar to ESSENCE, the primary design objective of RODS is to automatically collect any amount or type of data from as wide area as necessary and perform data analysis in close to real-time.

In RODS architecture, computing systems at each hospital or health-care facility stores patients' registration with chief complaint data. These registration data are sent to the central system as Health Level 7 (HL7) messages over a secure virtual private network (VPN). RODS uses a naive Bayesian classifier to classify the free-text chief complaints into one of seven syndrome categories: *constitutional, respiratory, gastrointestinal, neurological, botulinic, rash, hemorrhagic*, which are stored in a centralized database. The system employs various anomaly detection algorithms including Recursive Least Square (RLS), Cumulative Sums (CUSUM) and a non-standard form of EWMA and CUSUM.

2.3 Syndromic Surveillance Information Collection (SSIC)

SSIC system [33] has been developed as a result of collaboration of the Clinical Information Research Group at the University of Washington and Public

Health-Seattle and King County. SSIC employs automated collection of data from heterogeneous sources, normalizes and stores data in a central database and provides secure, remote access to data for public health professionals running aberration detection software. The surveillance system collects anonymized patient data from multiple health-care systems in King County, Washington. Data are collected in the form of ICD-9 codes or keywords present in chief complaints. Data is extracted on daily basis from the remote sites into central system. Data are transmitted through secure Hyper Text Transfer Protocol (HTTPS). Data are normalized into common eXtensible Markup Language (XML) format before being entered into the database. Some aberration detection software written in SAS is being used at the central system. No additional information on the performance or efficiency of the SSIC system is available at this time.

2.4 Biological Spatio-Temporal Outbreak Reasoning Model (BioSTORM)

BioStorm [34] introduces a novel knowledge-based approach to integrate surveillance data and knowledge from disparate data sources for rapid epidemic detection. Statistical time-series methods attempt to detect abnormality using raw disease counts, whereas humans use knowledge and concepts to reason about increased trends and sudden spikes. Epidemiologists can use qualitative knowledge to understand the temporal and semantic relationships among data from different sources. The goal of BioStorm is to incorporate qualitative knowledge into statistical analysis. BioStorm identifies three sources of data:

1. Pre-clinical data: school or work absenteeism.
2. Clinical pre-diagnostic data: test orders, signs and symptoms.
3. Diagnostic results: test results and case interviews.

BioStorm also proposes to create two knowledge bases:

1. Syndrome knowledge base, which defines relationship of individual data to syndrome for illness.
2. Epidemic knowledge base, which defines relationship of population-level-aggregate-data to the state of epidemic.

Pre-clinical data, clinical pre-diagnostic data and diagnostic data are analyzed to detect cases of syndromes using rules and relationships of syndrome knowledge base. After this stage, a spatial registration method converts data into common spatial representation (e.g. aggregation of syndromes to ZIP codes) to facilitate spatial analysis. Next, data are normalized to account for variations in data sources due to temporal cycles. Once data are normalized, BioStorm uses spatial and temporal statistical methods to detect outbreaks. BioStorm proposes to use epidemic ontology in the detection process.

2.5 National Retail Data Monitor (NRDM)

The National Retail Data Monitor [18] is a syndromic-surveillance based approach that collects and analyzes sales of over-the-counter (OTC) health care products

to detect outbreak. It has been observed that at the early onset of epidemic, the mass population that have been exposed to disease agents may show common syndromes and are more likely to self-treat with OTC health care products that to see a physician. Therefore, monitoring sales of health care products can detect outbreaks at early stage and give health professionals some precious lead-time to respond. The National Retail Data Monitor receives data daily from 10,000 stores that belong to national chains. Sales data include products' Universal Product Code (UPC). Each of these UPCs is mapped one of pre-defined product categories. Sales of products can be aggregated to clusters of zip codes. Unusual patterns of sales can be visualized and monitored on maps.

2.6 Lightweight Epidemiological Advanced Detection Emergency Response System (LEADERS)

LEADERS [35] introduces the concept of pattern-based surveillance whereby experts are interviewed to create specific patterns which are then applied to collected data. LEADERS supports three types of data feeds:

1. Continuous Data Feeds: LEADERS published an XML Document Type Definition (DTD) and plans support for Health Level 7 (HL7) data via XML feed.

2. Web-based Manual Data Collection: Data entry based on individual patient encounters using minimal data entry tasking.

3. Bulk Data Load: In-Patient Data and Out-Patient Encounter Data.

After September 11 2001 various hospitals nationwide have been connected to this system. It was deployed at the 2001 World Series event. Some of the other prior events include: 2001 Super Bowl in Tampa, Florida, 2001 World Series in Phoenix, Arizona, and New York City and 2000 Republican National Convention.

The system performs four levels of monitoring:

- Population Monitoring: examples, a rapidly increasing number of diagnosed cases of a specific disease in a normally healthy population; and lower attack rates among people who had been indoors, especially in areas with filtered air or closed ventilation systems, compared with people who had been outdoors.
- Diagnosis Monitoring: examples, bubonic plague; diagnosis of patients with a positive tuberculosis test; and dengue fever.
- Syndrome Monitoring: examples, respiratory; gastrointestinal; fever; and rash.
- Event Monitoring: examples, any single syndrome rises ten percent higher than the running event average for all event locations; and total admissions among all hospitals for an event rise ten percent.

2.7 Composite Occupational Health and Operational Risk Tracking (COHORT)

COHORT [36] provides real-time surveillance of the medical care and treatment of specified groups of military personnel across multiple medical health facilities throughout the world.

COHORT leverages existing normalized clinical data available from operational decentralized Integrated Clinical Database (ICDB) sites. Locally deployed ICDB systems support the Military Health System health care providers who deliver clinical services to all enrolled members of the military health care community. As entries and updates are made on the local ICDB system, a software agent transmits the new or updated medical data to the centralized COHORT database. In effect, the data are made available to COHORT at the same time the data is made available to the local health care provider.

The system uses a variety of commercial off-the-shelf (COTS) software to develop reports, charts and database. The architecture is based on J2EE technology with an Oracle database backend. The system also utilizes Oracle Data Mining (ODM), which incorporates supervised and unsupervised learning models. Supervised learning models, sometimes called *directed models*, are used to predict a value, or probability. These techniques are appropriate for scenarios where you identify a dependent variable and want to model how a group of independent variables influence it.

2.8 Infectious Disease Surveillance Information System (ISIS)

In the Netherlands, an automated outbreak detection system for all types of pathogens has been developed within an existing electronic laboratory-based surveillance system called ISIS [37]. Features include the use of a flexible algorithm for daily analysis of data and presentation of signals on the Internet for interpretation by health professionals.

Over 90% of the 76 microbiologic laboratories in the Netherlands are associated with public hospitals; less than ten percent are private laboratories not associated with hospitals. Other than ten notifiable infectious diseases, microbiologic laboratories have no legal requirement to provide data for surveillance. Since 1994, ISIS has collected anonymous positive and negative test results on more than 350 pathogens directly from voluntarily participating laboratories on a daily basis in a fully automated system that uses electronic data interchange.

The system uses two types of statistical algorithms for identifying outbreaks:

1. The first is a rolling seven-day total calculated daily: This variation is based on an algorithm that calculates expected weekly totals of a certain pathogen and a threshold value of 2.56 standard deviations from the mean (equivalent to a 99% confidence interval). A seven-day window advances day-by-day as new data enter the system and a new seven-day observed total is calculated daily and compared with the expected value for that epidemiologic week (Monday to Sunday). If the observed total is over the threshold, a signal is generated.

2. The second algorithm variation is a four-week rolling total calculated daily: Each week, this algorithm calculates an expected total for the previous four weeks and a 99% confidence interval. A four-week window advances day-by-day and a new four-week observed total is compared with the expected total for the four epidemiologic weeks ending with the current week.

2.9 Canadian Early Warning System (CEWS)

A Canadian based online syndromic surveillance system [38, 39, 40] that receives data from multiple diverse sources and performs analysis using numerous algorithms. Automated anomaly alerts are generated from abnormal data trends based on a set of algorithmic processes and provides graphing and mapping capabilities to assist in the analysis of data trends.

Primary goals of CEWS as a community-based health surveillance program are: (1) to consolidate and integrate health data and information from many *points of care* (emergency room, telehealth, lab, and over-the-counter drug sales data) in order to decrease delays in public health event detection and response; and (2) to reduce the impact of a bioterrorism event by minimizing the delay in detection, increase the speed of characterization, and ultimately enhance the timeliness of action.

At present three real-time data sources have been connected:

1. *Health Links -Info Sante* a nurse advice line available to the entire population of Winnipeg. Collecting chief complaints that can be further divided into syndrome groups, and demographic information.

2. *E-Triage* The triage system in place in every emergency department in the Winnipeg Regional Health Authority (WRHA). Collecting the presented complaint, demographic information, and triage scores.

3. *OTC drug sales* daily sales from various national chain drug stores. Currently have approximately twenty stores providing information on gastrointestinal (GI) products, and Memorandum of Understanding (MoUs) with other large chain stores are currently in the works for greater data coverage.

2.10 Early Aberration Reporting System (EARS)

The Early Aberration Reporting System (EARS) [29], developed by the Centers for Disease Control (CDC) and Prevention, is a collection of aberration detection methods to enable national, state and local health departments analyze public health surveillance data. EARS collects data on infectious diseases such as hepatitis A, hepatitis B, measles, mumps, and influenza-like illness. Based on the availability of baseline data, aberration detection methods in EARS are classified into three categories:

1. Long-term implementation methods which require a baseline of three or more years of data to be available. These include detection methods such as Historical Limits Method [41], Log-linear regression Model [42], quality control Cumulative Sums (CUSUM) Method, quality control Compound Smoothing Technique [43], Cyclical Regression Model.

2. Long-term implementation methods with limited baseline data of seven days to three years. These include detection methods that are variants of

CUSUM.

3. Short-term drop-in surveillance methods with minimal baseline data of one to six days of data. These include detection methods such as P chart, moving average and CUSUM.

2.11 BioSense

According to [44], BioSense is a CDC initiative to support enhanced early detection, quantification, and localization of possible biologic terrorism attacks and other events of public health concern on a national level. The system has implemented three national data sources: (1) laboratory test orders; (2) military treatment facilities; and (3) veteran affairs treatment facilities. The data is analyzed based on eleven syndromes: botulism, fever, gastrointestinal, hemorrhagic illness, localized cutaneous lesion, lymphadenitis, neurologic, rash, respiratory, severe illness and death, and specific infection.

Two main analytical techniques are used within the system. The first one is an adaptation of a Generalized Linear Mixed Modeling (GLMM) technique [45] called the Small Area Regression And Testing (SMART). This model takes into account multiple parameters including ZIP codes, say of the week, holiday, and day after holiday. The second technique is an adaptation of CUSUM approach.

2.12 Summary

There are many more systems in place today in various parts of the world, example of which have been discussed in [46, 47, 48, 49, 50, 51, 52, 53, 54]. An important point to highlight is that most of these systems collect data from remote locations to a centralized repository and then perform data analysis with various types of front-ends for data presentation. As discussed, this centralized data collection approach results in large amounts of data. However, such systems must comprise of efficient methods and processes to deal with large amounts of data and anomaly interpretations. Furthermore, with declining cost of memory storage devices and ease of getting access to aggregate data, such systems can play a vital role in identifying multi-disciplinary health threats. Chapter 4 presents a conceptual architecture that can be used to deal with these concerns and also supports multiple diverse data sources as well as effective data analysis and presentation.

Chapter 3

Techniques for Anomaly Detection

3.1 Introduction

With recent development in technology, real-time statistical disease monitoring has become an important issue. Biological terrorism and warfare has motivated development of computer systems for automatically detecting abrupt and epidemiologically significant changes in public health surveillance data.

Careful investigation of historical data in the literature has shown that different diseases provide different patterns of variation. Most can be classified into one of the following four types [55] (see the Fig 3.1).

- Type A: Diseases with periodic patterns and outbreaks occurring arbitrary. Examples include Campylobacter and Salmonella.

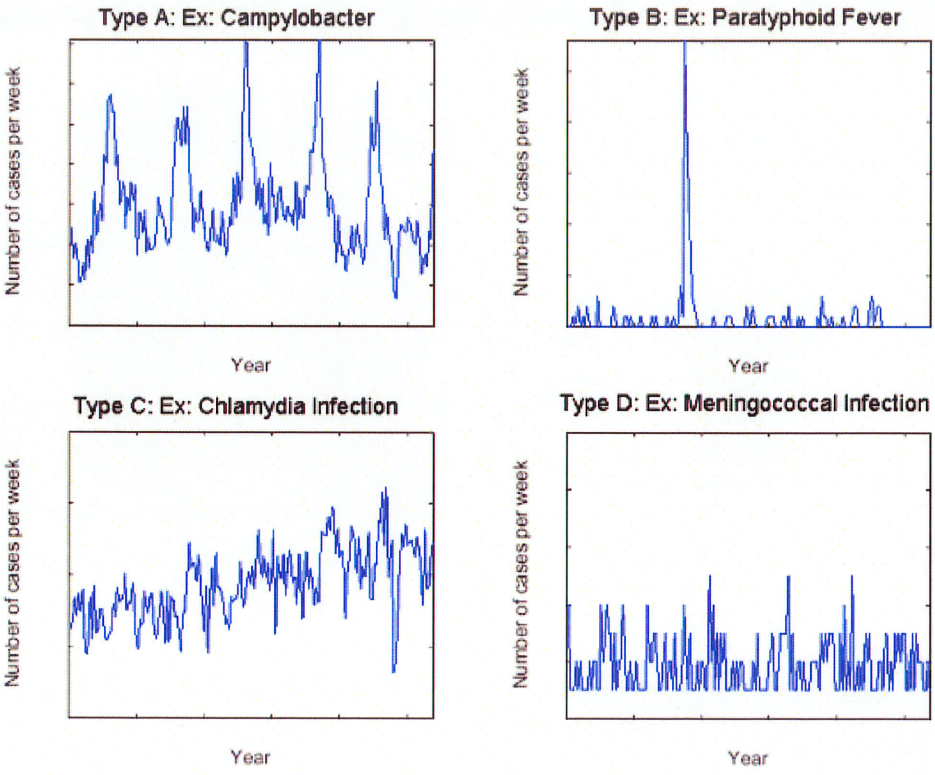

Figure 3.1: Trend Classifications (adapted from [55])

- Type B: Low frequent diseases with significant visible outbreaks. Examples include Paratyphoid fever and Tularemia.
- Type C: Diseases without any outbreaks. Example includes Chlamydia infection.
- Type D: Low frequent diseases without any outbreaks. Examples include Meningococcal infection and Typhoid fever.

Researchers are typically interested in monitoring diseases of Type A, which require sophisticated algorithms to parse through noisy and variant data.

The following defines some key parameters typically used to measure the performance of a statistical algorithm for anomaly detection. As per [56], *sensitivity* and *specificity* (also referred to as predictive value positive) are the two main parameters as defined below:

Detected by surveillance	Condition present: Yes	Condition present: No	
Yes	True positive A	False positive B	A+B
No	False negative C	True negative D	C+D
	A+C	B+D	Total

Figure 3.2: 2 by 2 Table (adapted from [56])

Sensitivity is a measure of how well the statistical method identifies relevant events. What a relevant event is, can only be decided by hand, by an epidemiologist. It comprises all events that correspond to outbreaks or other events of public health interest. Using the 2 by 2 table (Fig 3.2),

$$sensitivity = \frac{A}{A+C}$$

Specificity or Predictive value positive is defined as relevant warnings performed over total warnings. Specificity is a measure of how well the statistical method filters out extraneous material from the relevant events. Using the 2 by 2 table,

$$specificity = \frac{A}{A+B}$$

3.2 Moving Average (MA)

Moving average method is directly applicable to a scalar signal (such as daily number of respiratory cases) [57]. This method, more commonly used in computational finance, simply compares the average count during the current time period against a threshold. It is a trend following method that smooths the volatile swings in the data.

The method is used for anomaly detection in many surveillance systems. Most commonly used time-windows for averaging are 3-Day, 5-Day and 7-Day windows.

$$\frac{1}{N}\sum_{i=0}^{N-1} t_i$$

where N = 3, 5, 7 and t_i is the actual count for day i.

Fig 3.3 [58] illustrates an example of a 3-Day, 5-Day and 7-Day moving averages. As expected, a 7-Day window produces significant smoothing of the signal as compared to 5-Day and 3-Day windows.

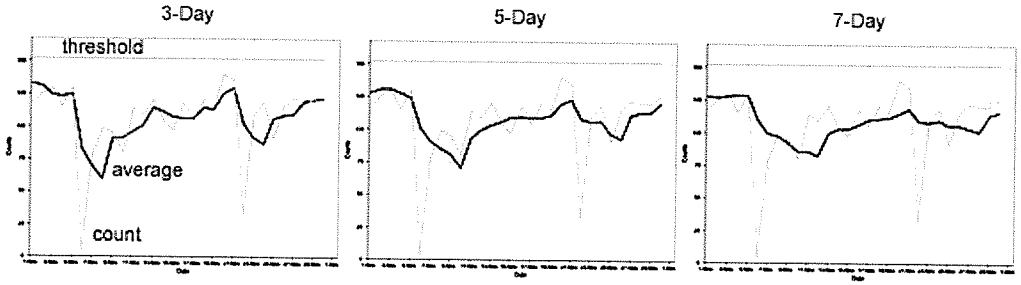

Figure 3.3: Moving Average variants [58]

Typically, an anomaly is generated when the actual moving average for a given day is over $\alpha\sigma$ where σ is the standard deviation of a predefined window of retrospective daily data and α is a constant which is normally set to two. All counts that fall between the average value μ and threshold $\mu + \alpha\sigma$ are flagged as Yellow or Orange alerts that convey some sort of activity that is likely to cause an anomaly in the near future.

Fig 3.4 illustrates various points of interest on the chart for a 5-Day moving average.

3.3 Weighted Moving Average (WMA)

The idea of moving averages of successive samples can be generalized as discussed in [57]. In principle, instead of a simple arithmetic moving average, a geometric moving average may be computed as follows:

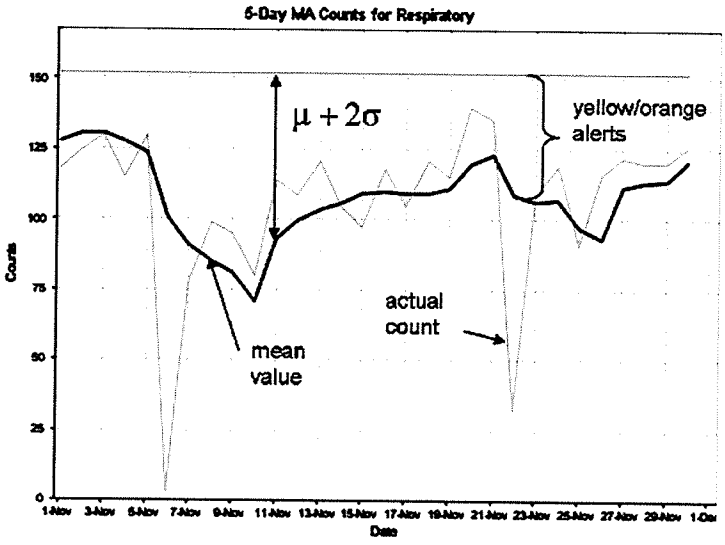

Figure 3.4: Points of Interest

$$y_n = \lambda_n t_n + \lambda_{n-1} t_{n-1} \ldots + \lambda_{n-k} t_{n-k}$$

where y_n is the weighted moving average value for given day n and k is the number of past observations. This can be generalized as:

$$y_n = \sum_{i=0}^{k} \lambda_{n-i} t_{n-i}$$

where λ is a linearly diminishing weight that assigns lesser importance to past data while giving more significance to most recent data. Fig 3.5 illustrates linearly diminishing weights for a 10-Day weighted moving average approach.

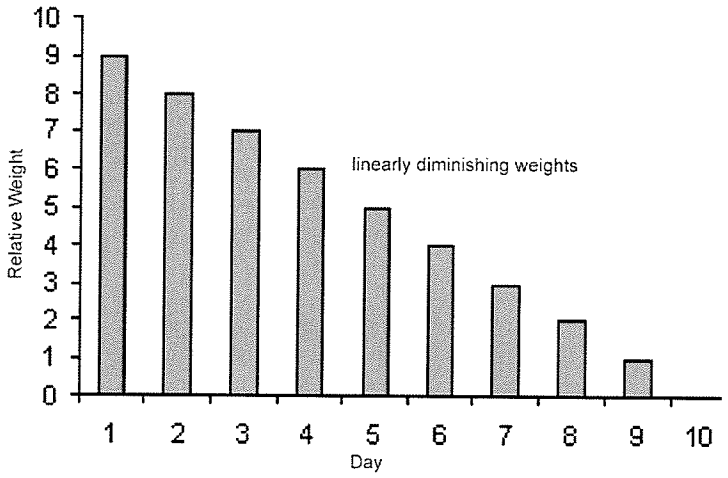

Figure 3.5: Weights for Weighted Moving Average

3.4 Exponential Weighted Moving Average (EWMA)

An simple moving average is special case of a weighted moving average with all the weight factors equal to 1 provided the window of data being observed is the same. One can use any weight factors, but a particular set referred to as *Exponentially Weighted Moving Average* has proven useful in applications ranging from air defense radar to stock trading [59, 57].

The concept of exponential weighting is not restricted to a specific time window. It can be applied at any length with diminishing weights . Fig 3.6 compares the weight factors for an exponentially smoothed ten-day moving average with a simple moving average that weighs every day equally, where x-axis represents the day and the y-axis represents the weight.

Typically an exponential smoothing gives today's measurement higher significance than the simple moving average would assign it, yesterday's measurement

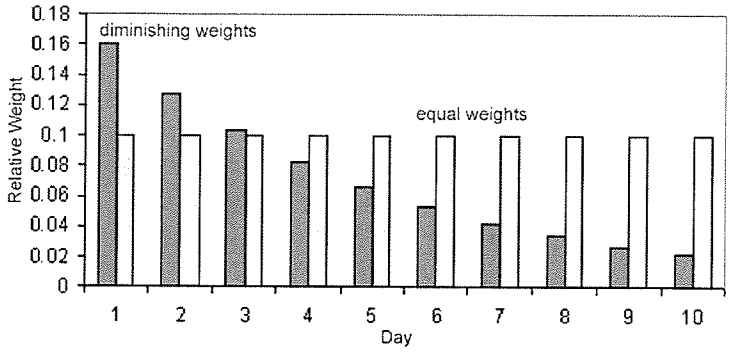

Figure 3.6: Weights for Exponentially Weighted Moving Average

a little less than that, and each successive day less than its predecessor. The recursive implementation of EWMA can be mathematically formulated as:

$$EWMA_t = \lambda y_t + (1 - \lambda) EWMA_{t-1}$$

where λ is the weighting parameter and y_t is the count for a given day t.

The weight factors in an exponentially weighted moving average are successive powers of $(1 - \lambda)$ called the *smoothing constant*. Smoothing constants less than 1 weigh recent data more heavily, with the bias toward the most recent measurements increasing as the smoothing constant decreases toward zero. If the smoothing constant exceeds one, older data are weighted more heavily than recent measurements.

Fig 3.7 shows the weight factors resulting from different values of the smoothing constant. Note how the weight factors are all one when the smoothing constant is one.

Smoothing constants between 0.5 and 0.9 provide a rapid degradation of

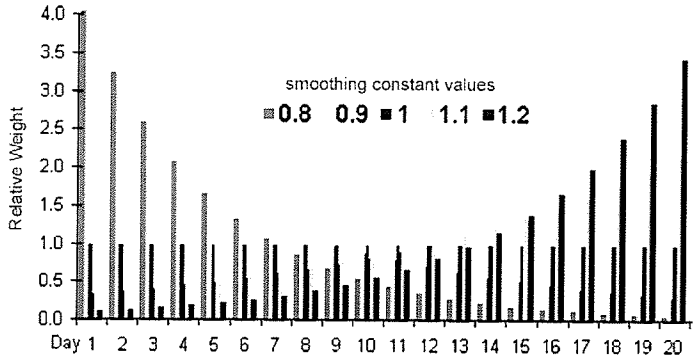

Figure 3.7: Smoothing Constants and Weights

weights for older data as compared to more recent measurements, thus there is no need to restrict the moving average to a specific number of days; one can average a large number of data points and let the weight factors computed from the smoothing constant automatically discard the old data as it becomes irrelevant to the current trend.

3.5 CUSUM: CUmmulative SUM

Developed in the early 1950s, the cumulative sum is a method for highlighting changes from a production average level [59]. Cusum has been frequently used in syndromic surveillance:

- In [60] researchers tried to detect clusters of Nosocomial Infection in data from 1995-2000. They compared cusum and a moving average and concluded cusum to be superior of the two.
- In [62] authors analysed retrospective data (1985-1996) of Mycoplasma pneumonia, using a thirteen week moving average to calculate the predicted

> value. They concluded: At national level, however, where data are further aggregated, cusums could quickly detect both focal and non-focal increases in low prevalence conditions, facilitating investigations in order to establish the reason for the increase.

A cusum chart is basically a graphical representation of the trend in the outcomes of a series of consecutive procedures performed over time. It is designed to quickly detect change in performance associated with an unacceptable rate of adverse outcome. At an acceptable level of performance, the cusum curve runs randomly at or above a horizontal line (no slope). However, when performance is at an unacceptable level, the cusum slopes upward and will eventually cross a decision interval. These are horizontal lines drawn across a cusum chart. Thus it provides an early warning of an adverse trend.

By convention the first cusum value (S_0) is set to zero. It is assumed that the population at risk is nearly constant in each considered period. A positive slope indicates a change above the expected, a zero slope is indicative of a period when the observed number of events is the same as the expected number, and a negative slope indicates that events have fallen below expected levels. An alarm is generated when the cusum exceeds a chosen threshold, h .

The difference between the observed number of cases and the expected number of cases is normalized as follows:

$$Z_t = \frac{X_i - \mu_0}{\sigma_{X_i}}$$

Then the test statistic, S_t, is calculated as follows:

$$S_t = max(0, S_{t-1} + (Z_t - k))$$

where k is the reference value and S_{t-1} is the previous Cusum. If $S_t > h$ then the algorithm flags the current measurement as a possible outbreak. The values of h and k are simulated individually for each disease.

3.6 EARS - C1, C2 and C3

The three methods C1, C2 and C3 within the early aberration reporting system (EARS) are based on a positive 1-sided Cusum calculation [29]. The methods C1-MILD, C2-MEDIUM, and C3-ULTRA were named according to their degree of sensitivity, with C1 being the least sensitive and C3 the most sensitive. For C1 and C2, the cusum threshold reduces to the mean plus three standard deviations (SD). The mean and SD for the C1 calculation are based on information from the past seven days. The mean and SD for the C2 and C3 calculations are based on information from seven days, ignoring the two most recent days. These methods take into consideration daily variation because the mean and SD used by the methods are based on a week's information. These methods also take seasonality into consideration because the mean and SD are calculated in the same season as the data value in question (see Fig 3.8).

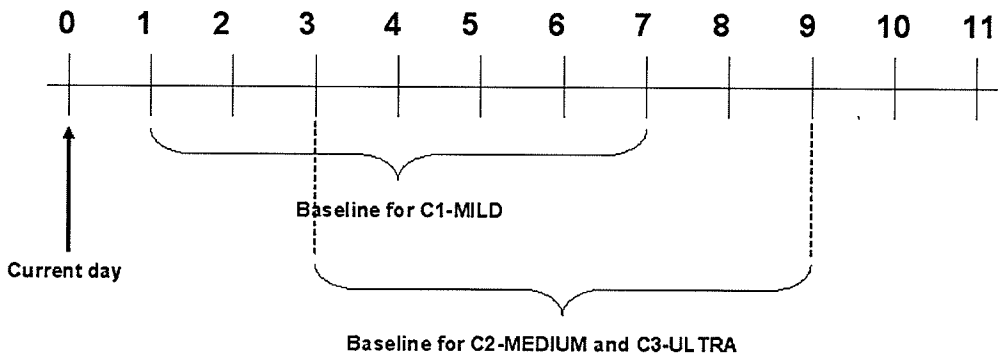

Figure 3.8: EARS Algorithms

3.7 Summary

In summary, significant efforts have been, and continue to be invested, in the development of effective analytical algorithms to detect anomalies so that the outbreak curve can be shortened. For instance, What's Strange About Recent Events (WSARE) [63, 64, 65] and Patient-based ANomaly Detection and Assessment (PANDA) [66] are algorithms that focus on individual level data rather than purely temporal data.

With recent advances in technology for data integration, there is potential for a vast amount of data from heterogeneous sources, which introduces new set of challenges that must also be addressed on a continual basis.

Furthermore, with the availability of dozens of different aberration detection algorithms, it is possible, if not probable, to get different results from different algorithms when executed on the same dataset. A meta-algorithm (or an envelope algorithm) that utilizes the strengths of multiple anomaly detection algorithms needs to be developed.

Chapter 4

ARTIST: A Conceptual System

As noted in the first chapter, an inter-disciplinary health surveillance approach needs to be embraced in which automated data acquisition and real-time decision support resources assist health officials monitor disease trends and detect outbreaks of diseases earlier and more completely than would otherwise be possible with traditional public health approaches. It is now well established that effective early outbreak detection in particular requires examining data sources beyond the traditionally monitored laboratory-based results. Public health surveillance information can be found in a number of non-traditional data sources, including syndromic data (e.g. emergency room visits, pharmaceutical sales, school absenteeism, telehealth), social data (e.g. news items), and risk factor data (e.g. weather, social events, water quality). Full utilization of both traditional and non-traditional public health surveillance data presents new challenges however, particularly the need to exchange data among disparate sources while preserving personal privacy.

This chapter proposes a conceptual architecture for a real time surveillance system and identifies fundamental and support components required to deal with various functions related to data collection, analysis and presentation. The discussion is intended to provide an understanding of various functional components of a typical real time surveillance system including a critical surveillance gap relating to interpreting anomaly detection outcomes produced by individual aberration detection algorithms. Analysis of this gap forms the basis of the research and is presented in detail in the next chapter.

4.1 Introduction

Based on a review of some existing syndromic surveillance systems together with personal experience in the research and development of public health surveillance resources, a conceptual model of a typical real-time data surveillance system was developed (Fig 4.1).

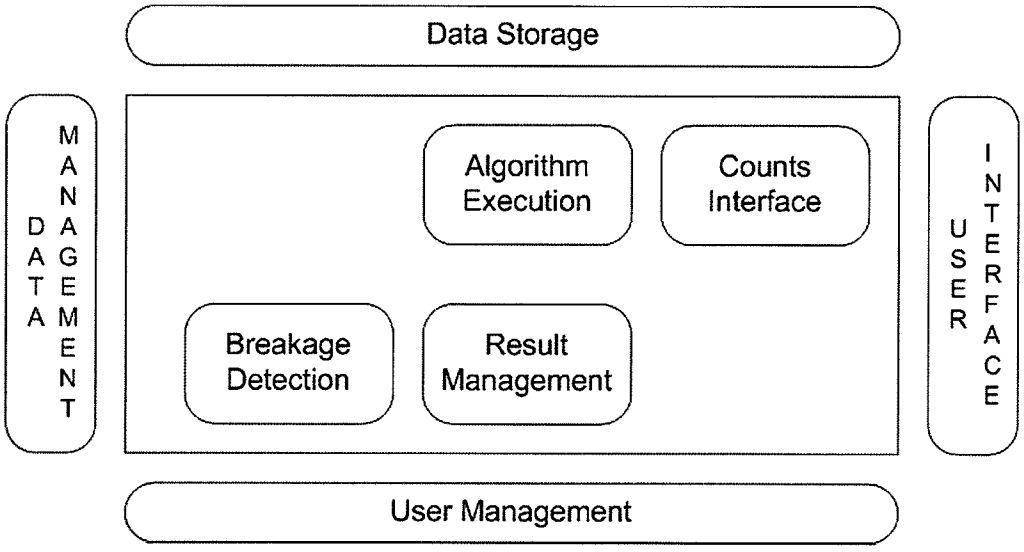

Figure 4.1: ARTIST Core Components

The model, referred to as the *architecture for real-time information standardization and transformation* (ARTIST), comprises of four fundamental blocks that address data and user management issues, and four supporting blocks that perform analysis and aggregation tasks. The fundamental blocks comprise of the following:

- *Data Management*: The primary purpose of this fundamental block is to perform all necessary data processing, preparing the data for analysis and presentation. Data Management is responsible for classifying the data based on standard codes - for example, SNOMED (Systematized Nomenclature of Medicine), UPC (Universal Product Code), ICD9-10 (International Classification of Disease), LOINC (Logical Observation Identifiers Names and Codes) - and aggregating data at various classifications (e.g., syndrome, province, health unit, city). Various technology and nomenclature standards need to be supported, including XML (eXtensible Markup Language), HL7 (Health Level Seven), SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language). This capacity allows interfacing with existing legacy systems, such as emergency room triage systems, case management systems, tele-health systems, and pharmacy databases.

- *Data Storage*: Once the data is received, it is stored in raw format for potential reverse linking if required. For example, someone with the proper authority may wish to review an emergency room chart associated with a particular record. Furthermore, the standardized and aggregated results also need to be stored to enable quick access for presentation of the information to the users. A structured query language (SQL) based database should be considered for this purpose.

- *User Management*: This functional block addresses user registration, authentication, and access control (role-based and target-based). It is responsible for controlling user access to data and information based on profile parameters, including data type (e.g., emergency room visits, tele-health, OTC), geo-level, organization, and function (e.g., mapping, charting, algorithm execution, data management). This ensures data confidentiality while allowing controlled access to multiple databases and applications. A typical identity management system could be used to handle this function.

- *User Interface*: This block manages the user interface of the system, how results are presented, and how anomaly alerts are distributed. Furthermore, it facilitates interactive access to counts and raw data to enable key functions including charting, mapping and reporting.

All of the four fundamental blocks discussed above may be fulfilled by off-the-shelf product(s) or custom solutions using standard technological platforms, such as Oracle and Java.

The **support function blocks** comprise of the following:

- *Breakage Detection*: This support block addresses a major challenge in an automated algorithm execution system, that is the ability to detect breakage in an incoming data feed. Lack of this ability could provide incorrect results generated from the automated analysis of an incomplete set of data.

- *Algorithm Execution*: This support block manages the execution of analytical anomaly detection algorithms on the aggregate data, together with the flagging of anomalies. This block must support both automated and manual algorithm execution.

- *Result Management*: The Result Management block manages results generated from potentially many algorithms and data sets, presenting results in an intuitive manner to the user. This includes the ability to portray the results both spatially (geographic) and temporally (time-based) in an integrated, accessible and navigation friendly view.

- *Counts Interface*: This block is responsible for negotiating between the user interface and data storage blocks. Once stored, data are aggregated based on system and user-defined parameters to facilitate anomaly detection and trend analysis at various geo-levels and classifications groupings. The Counts Interface block can be realized using basic software as it is primarily a generic data access layer for information presentation.

The remainder of this chapter will elaborate on three of the four support blocks: Breakage Detection, Algorithm Execution, and Results Management. But first it is necessary to introduce a novel data organization system, called the *source-classification-spatial* (SCS) hierarchy.

4.2 SCS Hierarchy

One of the most complex issues that a surveillance system must addressed is the ambiguous nature of the hierarchy of syndromes/categories and geographical areas. Some data sources facilitate clear definitions of syndromes, for example, emergency room data where syndromes are directly defined using chief complaints. Others present a challenge, for example, pharmaceutical over-the-counter sales data whereby specific drugs make up disease types which in turn

make up a syndrome. Further complication results from the fact that not all data sources provide data in a standard format nor do they provide data at the same granular level such as product UPC (universal product codes). An essential prerequisite to algorithm execution and results management are clear definitions of the mutli-level hierarchies of syndromes and geographical areas. A generic hierarchy framework that adapts to varying definitions of syndromes as well as to geography and the data provider is very important.

A novel data organization system called the *source-classification-spatial* (SCS) hierarchy provides a means to define a structure for enabling aggregation and analysis at various groups of **source**, **class**, **geolevel** triplets. The three variables of the triplet are defined as follows:

1. A *source* is the type of data being received by the system such as emergency room visits, laboratory results and pharmaceutical over-the-counter (OTC) sales. Each data source has unique characteristics which must be understood. For example, emergency room visits are based on patient arrival and triage times; patient counts are aggregated. Similarly, tele-triage data (nurse help line) are like emergency room visits where patient call time replaces patient arrival time. On the other hand, OTC data are based on specific drug sale counts for a given store. All data source types are typically interfaced into the system via a set of diverse communication pathways including push and pull file transfer protocol (FTP), Health Level 7 (HL7) standard, Electronic Data Interchange (EDI) format and Web services. Data receiving facilities are required to implement receivers and handlers to manage such diverse channels and formats.

2. *Classification* provides decomposition of data for aggregation at specific

disease grouping levels such as syndromes and chief complaints. This is typically the most confusing hierarchy and is dictated primarily by the granularity of the information being received. For example, for emergency room data it can be quite straightforward with chief complaints making up a syndrome. That is, there are two levels in the classification hierarchy. On the other hand, for OTC data as an example, classification can be complicated by an organization's willingness to disclose granular information. For example, one organization may provide data at the lowest level possible (drug UPC), while another may only provide drug category such as *Antinauseant* or *Antidiarrheal*. This diversity in granularity amongst various organizations makes it very difficult to handle analysis and thus, a clear definition of the classification hierarchy is paramount. An example of such a hierarchy for OTC data is a hierarchy with four levels including syndrome, drug category, age grouping (adult versus pediatric) and product (UPC).

3. The *spatial* (or geolevel) of the SCS hierarchy is the actual geographical decomposition of the area under consideration, which typically varies between countries or even within a country. A four-level hierarchy is proposed including province or territory, health unit, city or town and forward sortation area (FSA) or the first three digits of the postal code. Note that organizations such as hospitals, laboratories and pharmaceutical stores are typically considered to be part of an FSA. Although, this hierarchy seems straight forward, there are complications at the last two levels of city/town and FSA. There is no implied relationship that FSAs are contained within city or a town. Some FSAs span across boundaries. Thus, a three-level hierarchy can be used where level one is the city/town or FSA; level two is

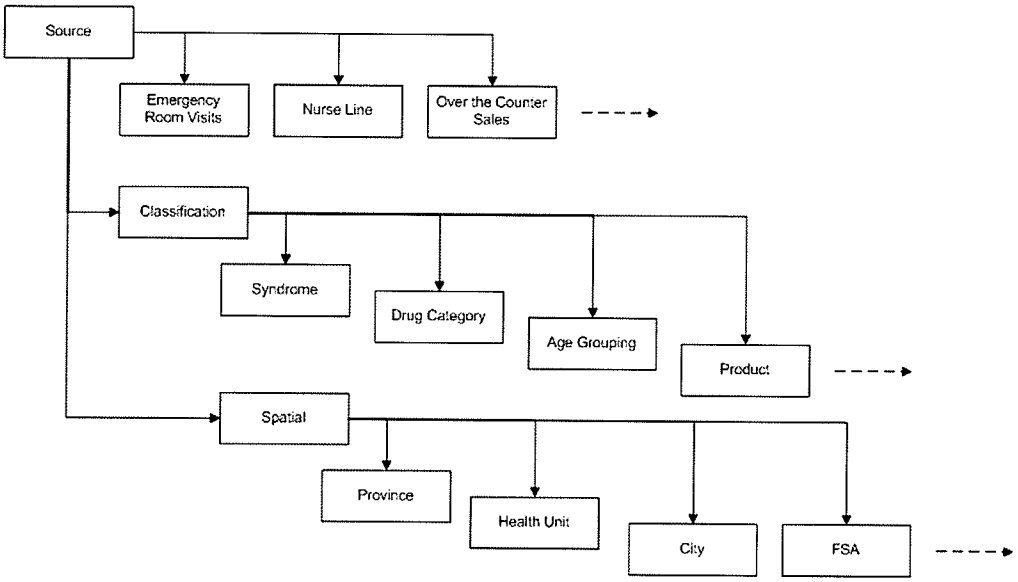

Figure 4.2: SCS Hierarchy

the health unit and level three is the province/territory.

Fig. 4.2 illustrates an example of an SCS hierarchy, where (OTC, Gastrointestinal (GI), Winnipeg) triplet would be an example of SCS hierarchy.

4.3 Support Block 1: Breakage Detection

One of the major challenges with automated algorithm execution system is the need to detect breakage in an incoming data feed (or a *data channel*). Lack of this ability could yield incorrect results due to automated analysis of an incomplete set of data. For instance, consider an emergency room visit data source that suddenly stops sending data. If the algorithm execution manager is set to run the algorithms once a day at 11PM, the system would analyze data on an incomplete set of data and thus would provide incorrect and potentially damaging result.

A novel algorithm used to detect breakage in a data channel using simple statistical methods is proposed. The algorithm, referred to as the statistical breakage detection algorithm, utilizes historical data on the received time for individual records or batch data based on individual input data sources.

4.3.0.1 Real Time Data Feeds

Systems that are capable of receiving data on close to real-time basis can make use of the received time (i.e. time the system receives the records) for breakage detection. The proposed algorithm makes use of mean (μ) and standard deviation (σ) based on time of the day. In order to add some seasonal variation, μ and σ are computed based on historical neighborhood scanning approach. That is, the values are based on 15 values centered on the specific time of day corresponding to current day from previous year as shown in Fig. 4.3.

As shown in Fig. 4.4, the algorithm generates a table based on the time of the day, in this case every hour. This table has five columns representing time

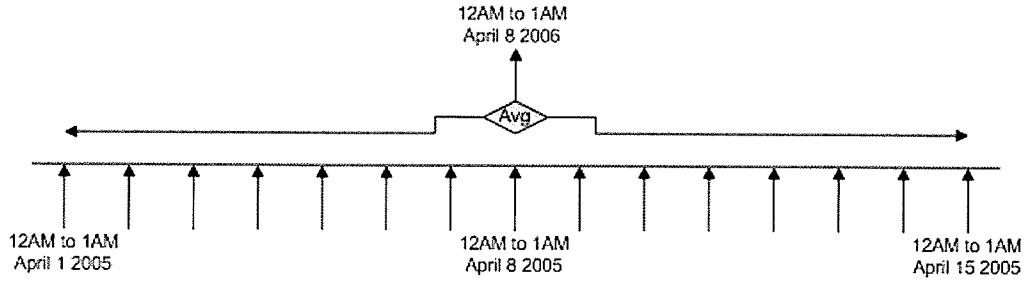

Figure 4.3: Statistical Breakage Detection Algorithm

Time of Day	Mean	Std Dev	Value	Outcome
1AM	5	1	4	0
2AM	5	1	4	0
3PM	50	4	60	+1
12AM	3	1	1	-1

Figure 4.4: Statistical Breakage Detection for Real Time Feeds

of day, μ , σ , actual count and the outcome. The outcome (or a decision) has three possible values: '0' or normal indicating that the value received during the time period is considered to be normal; '+1' or anomaly indicating that the value received is higher than what is expected (greater than $\mu + \sigma$) and thus can be used to flag an anomaly; and '-1' or breakage indicating that the value received is lower than what is expected (less than μ - σ) that can be used to flag a breakage.

This algorithm needs to be executed once a day, for instance at 12AM, to create a complete 24-hour table which can be used throughout the day based on scheduled execution at a specified interval, for example on hourly basis (see Fig. 4.4).

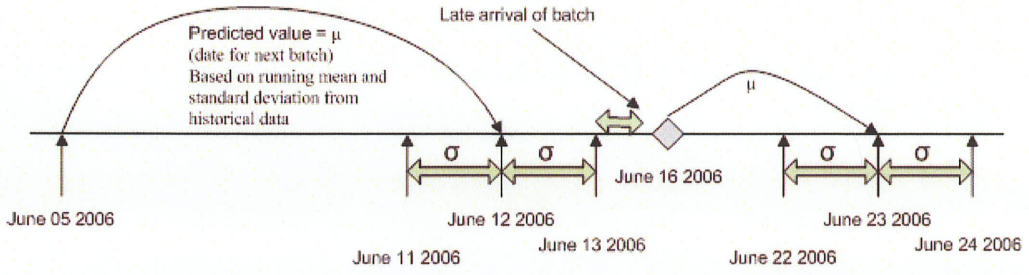

Figure 4.5: Statistical Breakage Detection for Batch Feeds

4.3.0.2 Batch Data Feeds

Some data providers do not have the ability to feed data in an automated fashion. Thus, it is necessary to define an algorithm that can handle bulk data (or batch data) on a routine basis. The algorithm proposed to deal with such routine data is based on expected arrival date instead of time of the day. Batch reception is not dependent on season but rather the infrastructure implemented by an organization. This makes the breakage detection quite straightforward in that breakages may be identified by simply looking at the historical mean and standard deviation for reception window, which is defined as the interval in number of days between two consecutive receptions. The reception window is data provider specific and needs to be updated every time a new batch is received at which time the predicted next batch arrival date is computed. A breakage is then defined as the absence of data on the expected day of batch arrival within computed standard deviation as illustrated in Fig. 4.5.

The next batch arrival date (or *predicted value*) is simply a running mean value based on historical reception windows for a given organization. In Fig. 4.5, predicted value based on batch arrival on June 05th is June 12th with standard deviation of one day. However, the next batch arrives on June 16th which is

after June 13th (predicted value + one standard deviation). This is a cause for a breakage and can be detected by the system using scheduled timers. Once the breakage has been fixed, new parameters are computed based on latest batch arrival date and the process repeats.

4.4 Support Block 2: Algorithm Execution

The SCS hierarchy can be very involved requiring numerous algorithms to be executed. This requires significant computational power as the algorithms need to be executed simultaneously and quickly in order to produce final results in a time efficient manner. This creates a need for an *execution manager* defined as a dynamic workflow manager to enable algorithm execution in a potentially multi-server environment. The *execution manager* is responsible for scheduling the algorithms and collating results. The basic concept is to execute the algorithms in a controlled manner so that the results can be stored in a format that enables seamless access to the data for presentation. Fig. 4.6 illustrates a sample flow chart for an execution manager based on three-level SCS hierarchy.

As illustrated, the process requires three steps to address the corresponding three levels in the SCS hierarchy. Level one addresses the data source by looping through all available data sources in the system. Level two takes care of classifications for each data source dealing with the classification hierarchy such as syndromes, category, age-grouping and product depending on the data source. Finally, level three addresses the geo-level hierarchy which can have multiple levels such as province, health unit, city and forward sortation area (FSA).

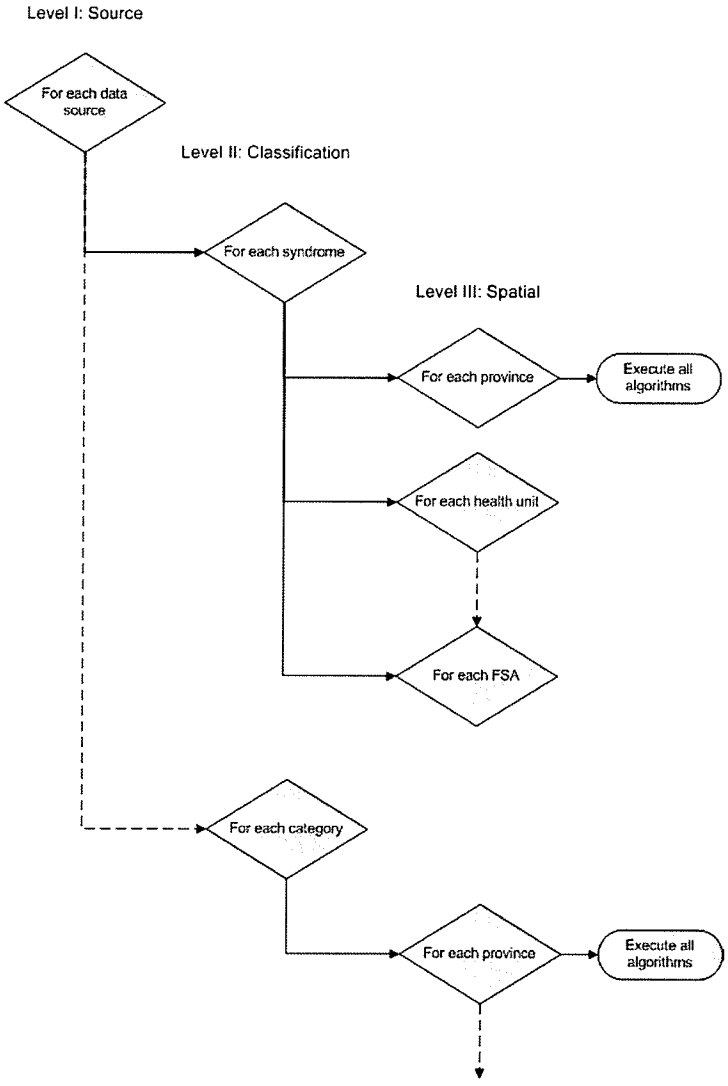

Figure 4.6: SCS Flowchart

In order to coordinate all of these algorithm executions, the execution manager needs to be setup as a queue to facilitate management of tasks. There are a few issues that need to be considered:

1. The ability to gracefully interject the queue and stop executions to enable server maintenance and reboots.

2. The system needs to be able to recover from abnormal power failures and be capable of completing any pending tasks that were interrupted.

4.5 Support Block 3: Result Management

As discussed in the previous section, there is a need for systems to execute multiple algorithms over multiple SCS hierarchy combinations (referred to as *groupings*) in order to identify potential anomalies. This creates a need for a structured system for managing the results so that they can be accessible for presentation in an effective manner. Some systems deal with this issue through the inline execution of algorithms. However, this approach does not address the automated identification of anomalies over a wide range of SCS hierarchy groupings which requires a large number of executions. That is, a user would be required to request a large number of inline executions to survey all possibilities; this would be computationally intensive as well as cumbersome. Thus, the ability for systems to store results over a specific time frame is critical in order to save computation power for repetitive executions and to allow for quick access for epidemiological analysis over a given time frame.

A discussion on result management is facilitated if we introduce a structure

referred to as a *result set.* The result set contains the outcomes of numerous analytical algorithms executed on a specific SCS group. This result set definition can get quite complex based on the nature of the geographical setup of the area under consideration. For example, the spatial hierarchy of Canada is based on a five level hierarchy comprising of country, jurisdiction, health unit, city/town and FSA. That is, Canada comprises of jurisdictions which can be a province or a territory. A jurisdiction is further divided into multiple health units, which are further divided into cities and towns. As can be quickly realized, the hierarchy can be quite involved, and thus the task of managing results can be very challenging.

Once the geographical setup of the area being investigated is well understood and the spatial hierarchy clearly defined, the next step is to implement a set of algorithms that will be used to analyze the incoming data. Each of the selected algorithms are executed over each possible SCS hierarchy for a given day.

A key area of concern is that with over several dozen available and easily accessible algorithms, how do systems and individuals make decisions based on several often varying outcomes, especially when algorithms are executed in parallel on the same dataset. There is a need for an envelope or a meta-algorithm process and/or method to extract intelligence from outcomes of algorithms in order to increase the likelihood of providing decision makers with a valuable single outcome.

Consider a system with four algorithms including a three-day moving average (MA), an exponential weighted moving average (EWMA), cumulative sums (CUSUM) and weighted moving average (WMA). The first two algorithms MA and EWMA suggest that there is a RED (or high-level) alert while the other two

algorithms (CUSUM and WMA) suggest that there is an ORANGE (or medium-level) alert. In other words, the four algorithms used by the system split their decisions on two different alert codes and thus create a dilemma for overall alert code. How is a user to interpret these results without being provided with some indication of confidence in the result set? Is the alert RED with twenty percent confidence or is it ORANGE with ninety percent confidence? Moreover, even in situations where the algorithms do not split on the decision, how confident is the system in suggesting a specific level of alert code or start of an outbreak?

4.6 Summary

This chapter provided architecture for a generic surveillance system, described various functional blocks, identified some critical gaps and proposed some novel solutions to these gaps that may be considered when designing such a system. In addition, a critical gap regarding interpreting outcomes of algorithms to generate was identified.

Chapter 5

The Framework

5.1 Introduction

As emphasized in earlier chapters, recent advances in technology have made it possible to gather, integrate, and analyze large amounts of data in real-time or near real-time. These new technologies have touched off a renaissance in public health surveillance. For the most part, the traditional purposes of health surveillance have been to monitor long-term trends in disease ecology and to guide policy decisions. With the introduction of real-time capabilities, data exchange now holds the promise of facilitating early event detection and to assist in day-to-day disease management.

With the availability of dozens of different aberration detection algorithms, it is possible, if not probable, to get different results from different algorithms when executed on the same dataset. The results of the study in [67] suggest that commonly-used algorithms for disease surveillance often do not perform well

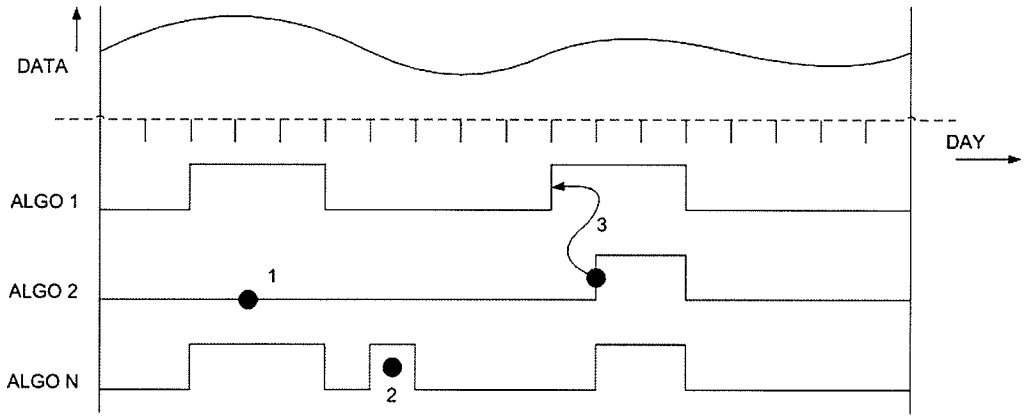

Figure 5.1: The Outbreak Detection Problem

in detecting aberrations other than large and rapid increases in daily counts relative to baseline levels. A new approach, denoted here as Confidence-based Aberration Interpretation Framework (CAIF), may help address this issue in disease surveillance by using a collective approach rather than algorithm specific approach.

5.2 The Problem Statement

Consider a system with multiple anomaly detection algorithms as illustrated in Fig 5.1. Due to differences in the implementation of the algorithms and parameters used (ex: thresholds, training periods and averaging windows), the outbreak decisions may vary significantly from one algorithm to another. On the other hand, there is also a possibility that these decisions are very similar for some set of algorithms. These two extremes create a dilemma for decision makers in that there could be a situation where most of the algorithms in a system suggest an outbreak, however, not knowing the relationships between these algorithms can result in a biased decision.

As illustrated, there are three main points of concern:

1. *False Negative:* Depending on the algorithm employed, there is a possibility of missing a real outbreak indicated as **1** in Fig 5.1. Obviously, this can be very damaging if the system were to make a decision based on that specific algorithm. False negatives can lead to potentially exponential damage within the general public due to delayed response to an outbreak.

2. *False Positive:* Some algorithms are susceptible to reporting false positives, that is, detect an anomaly during *peace* time. Most systems set their anomaly detection thresholds to be as sensitive as possible to minimize the risk of missing important events, producing frequent false alarms, which may determined to be false positives by subsequent investigation. These systems face inherent trade-offs among sensitivity, timeliness and number of false positives. False positives have a negative impact on public health surveillance because they can lead to expensive resource utilization for further investigation and can cause undue concern among the general public.

3. *Delayed Identification:* During initial stages of an outbreak, the number of cases are on the rise and hence detecting an outbreak at this point could be very effective and potentially aid in minimizing the impact of a potential bioterrorist attack. However, depending on the algorithm(s) employed, a system may end up with some algorithms detecting outbreaks well beyond the actual start day. This, once again, can be very costly to public health community and impact it negatively for obvious reasons.

These three concerns result in a trade-off situations between false positives, false negatives and detection time which are typically addressed by looking at

sensitivity, *specificity* and *time to detect* parameters as discussed later in the chapter.

In summary, a framework needs to be implemented that would enable a user/system to interpret the anomaly detection results with some indication of confidence. That is, is there a potential start of an outbreak with twenty percent confidence or is it ninety percent confidence? A framework that takes into account the relationships between algorithms and produces an unbiased confidence measure for identification of start of an outbreak (see Fig 1.1).

5.3 The Proposed Solution

The proposed anomaly interpretation framework aims to enhance surveillance decision-making by combining results of multiple aberration detection algorithms through the use of key result metrics. Fig 5.2 depicts the four steps of the proposed framework and the associated linkages between them.

5.3.1 Step 1: Specificity, Sensitivity and Time To Detect Evaluator

Traditionally, *specificity* and *sensitivity* have been used for comparing various algorithms and their performances. In this study, these two parameters are key in helping identify a subset of algorithms (referred to as **minimal set**) that would be sufficient to deduce an overall decision to detect start of an outbreak. The hypothesis is that the system may not require all candidate algorithms to come

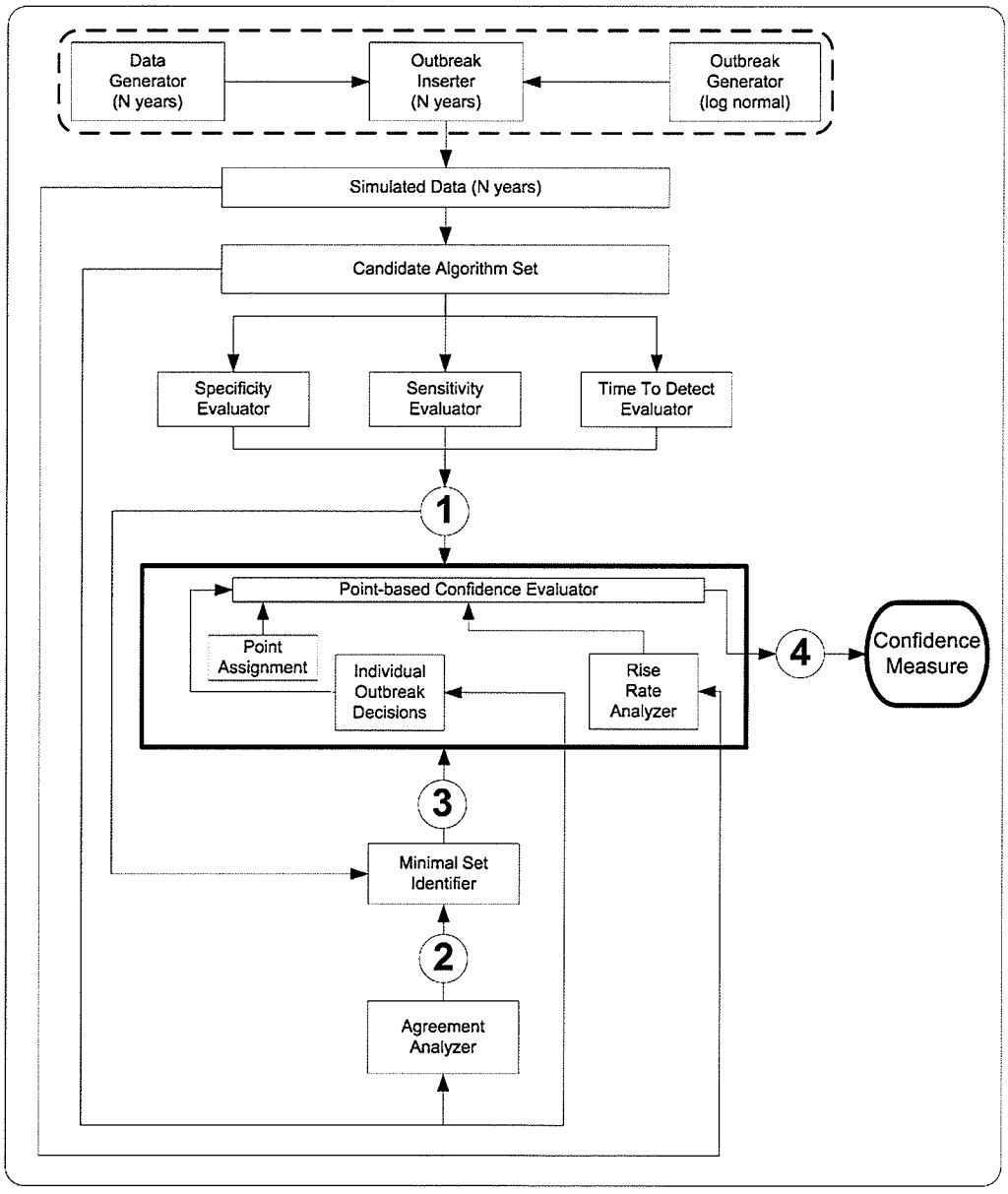

Figure 5.2: The Confidence-based Anomaly Interpretation Framework

up with a good decision as some of them may provide redundant information.

Sensitivity of an algorithm for a given dataset is defined as the total number of outbreaks during which the algorithm *flagged* (at least once per outbreak) divided by the total number of outbreak periods in the dataset[1]. Specificity of an algorithm for a given dataset, on the other hand, is defined as the total number of non-outbreak days on which the method **did not** flag divided by the total number of non-outbreak days in that dataset:

$$Sensitivity = (True\ Positive\ Count)/(Total\ Number\ of\ Outbreaks)$$

$$Specificity = (True\ Negative\ Count)/(Total\ Number\ of\ No\ Outbreak\ Days)$$

In addition to specificity and sensitivity, a third parameter called *time to detection* (TTD) defined as the average number of days from the first day of an outbreak until it was flagged by the algorithm, plays a vital role in the forthcoming analysis. This is a very important parameter as it aids in segregating a set of algorithms into various groups (or classes) and provides a very clear differentiation between set of algorithms based on its interpretation.

Fig 5.3 illustrates, in time, a progression of a sample outbreak over multiple days. Periods with no outbreaks are referred to as *peace-time*, while *outbreak-mode* refers to a time period with outbreak days.

The time to detect parameter, shown as TTD in Fig 5.3, is computed using a six step decision logic as illustrated in Fig 5.4. There are four variables of interest: i represents the day count within the simulated data set; $startIndex$ stores the

[1]A single outbreak usually lasts more than one day

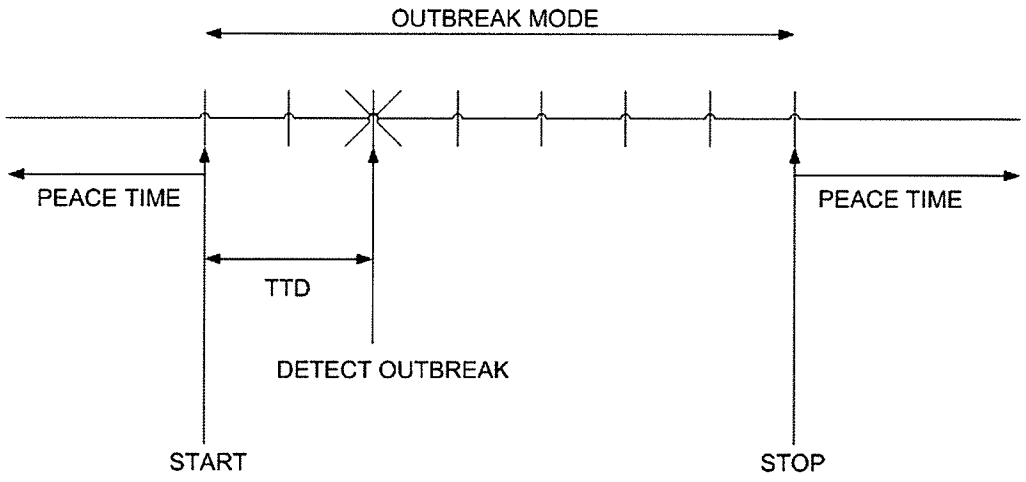

Figure 5.3: A Sample Outbreak

value of i that is the start of a specific outbreak period; *peaceTime* indicates whether the previous day was an outbreak (in which case the variable would be set to **false**); and *processed* which indicates whether the current outbreak period has already been counted as an outbreak to avoid duplicate counting of a given outbreak period. The flowchart comprises of six decision points:

1. *Decision Point 1*: Check the day counter i to decide whether all the days have been processed within the current data set, where each line of the comma delimited data set represents a day. If there are more days to be processed, then go to next step. Otherwise, exit.

2. *Decision Point 2*: Check whether the current day being processed has been flagged as an outbreak day. If not, then set *peaceTime* variable to **true** and go back to step 1. Otherwise, proceed to next step.

3. *Decision Point 3*: Check if the *peaceTime* variable is set to **true**. That is, are we currently in peace-time mode. If not then go back to step 1. Otherwise, set *startIndex* to current value of i (that is, current day as the

start of an outbreak). Also set *peaceTime* mode to **false** as we have entered an outbreak period. Furthermore, set *processed* flag to **false** to indicate that the current outbreak period has not been detected by the system yet and thus, has not contributed to the time to detect value. Proceed to next step.

4. *Decision Point 4*: Check whether the system (or an algorithm) has detected an outbreak. If not, then go back to step 1. Otherwise, proceed to next step.

5. *Decision Point 5*: Check whether the detected outbreak has been processed. That is, is the current outbreak part of a given outbreak period that has already been considered. If that is the case, then go back to step 1. Otherwise, proceed to next step.

6. *Decision Point 6*: Check whether this start of an outbreak is the very first one to be processed. If so, then initialize the time to detect value to $i - startIndex$ to get the difference between actual start of an outbreak and the day when it is identified. Otherwise, simply average the current time to detect value with $i - startIndex$ to compute a running average. At this point, set *processed* flag to **true** and go back to step 1.

The three parameters discussed in this section provide a wealth of insight into the goal of identifying a minimal sub set of algorithms sufficient for generating an overall confidence value for an anomaly indicator. Note that the duration of the outbreak is also a parameter that should be discussed. Since the focus is on identifying start of an outbreak, the duration does not really provide any insight that can be used in early identification. Thus, it will not be considered moving

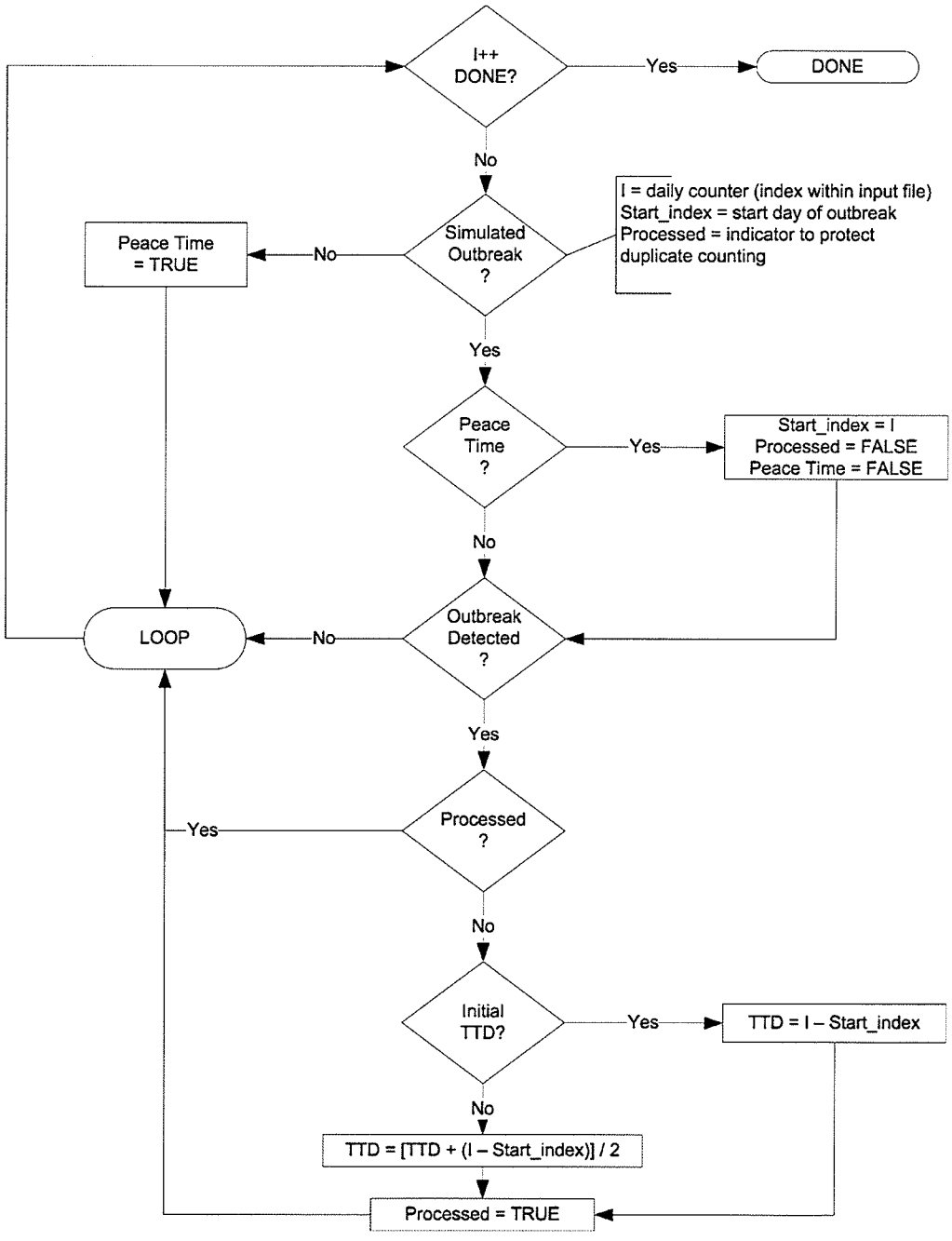

Figure 5.4: Computing Time To Detect parameter

forward.

5.3.2 Step 2: Agreement Analyzer

Agreement analyzer deals with quantifying the degree of agreement or relationship between any given two algorithms executed on the same data set. That is, are all candidate algorithms producing unique results? Or, is it that some algorithms yield similar results and thus provide no added value to the overall decision? This step of the framework exploits such relationship and/or agreement between any two algorithms using two quite different approaches: **Correlation** and **Kappa Coefficient**.

5.3.2.1 Correlation

Correlation is one of the most common and most useful statistics. A correlation, **r**, is a single number that describes the degree of linear relationship between two variables (also referred to as bivariate relationship). A positive relationship, in general terms, means that higher scores on one variable tend to be paired with higher scores on the other and that lower scores on one variable tend to be paired with lower scores on the other.

The correlation between two variables, in this case the two algorithm values or decisions, can be obtained using [71]:

$$r = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}}$$

where x and y are the time series for daily counts, N is the total number of days in the time series, $\sum xy$ is the sum of products of paired counts, $\sum x$ is the sum of counts from first algorithm in the pair, $\sum y$ is the sum of counts from second algorithm in the pair, $\sum x^2$ is the sum of squared x counts and $\sum y^2$ is the sum of squared y counts. $\rho_{correlation}$, the agreement matrix based on correlation, is obtained using the above formula as follows:

$$\rho_{correlation} = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix}$$

where r_{XY} is the correlation value for algorithm X against algorithm Y.

Once the correlation matrix has been computed, it is necessary to consider the significance of obtained correlation values between any pair of algorithms. That is, what is the probability that the correlation obtained in the sample came from a population where the correlation is 0? The *t-test* in the following equation can be used to answer this question.

$$t_{XY} = \frac{r_{XY}\sqrt{(S-2)}}{\sqrt{(1-r_{XY}^2)}}$$

where S is the sample size and (X, Y) are the algorithms being considered. This generates an NxN matrix of significant values for each algorithm pair. Based on the results and critical values of t, the minimum agreement threshold based on correlation $T_A^{correlation}$ can be deduced. This is the value that can be used in the next step of the framework to identify nearest neighbors for each algorithm based

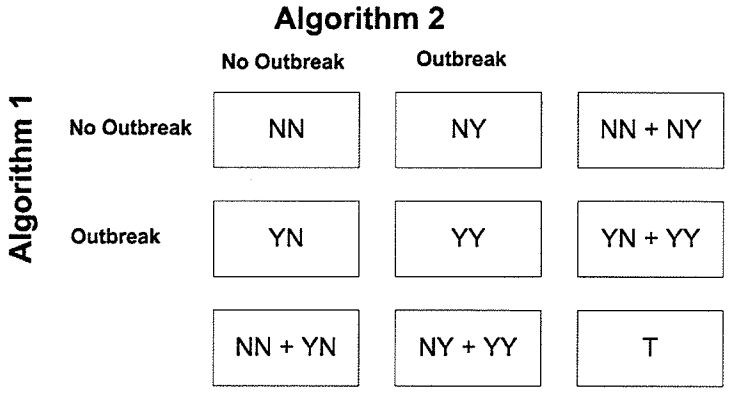

Figure 5.5: Kappa Coefficient: 2 by 2 Table

on the strength of the relationships.

5.3.2.2 Kappa Coefficient

An alternative approach to correlation matrix is the computation of *Kappa Coefficient*, which is an index that compares the agreement against that which might be expected by chance. Kappa can be thought of as the chance-corrected proportional agreement, where possible values range from +1 (perfect agreement) via 0 (no agreement above that expected by chance) to -1 (complete disagreement).

Next, lets consider a matrix generated using Cohen's kappa coefficient approach [72]. Consider a 2x2 table capturing decision outcomes by two different algorithms being compared as shown in Fig 5.5.

The following formula was used to compute the kappa coefficient between any two algorithms:

$$\kappa = \frac{(P_o - P_c)}{(1 - P_c)}$$

$$P_o = \frac{NN + YY}{T},$$

$$P_c = \frac{NN + NY}{T} * \frac{NN + YN}{T} + \frac{NY + YY}{T} * \frac{YN + YY}{T}$$

where P_o is the relative observed agreement and P_c is the probability that the agreement is due to chance. If we had two time series x_i and y_i of length $\mathbf{T}$ with $\mathbf{1}$ representing an outbreak and $\mathbf{0}$ representing no outbreak, then

$$NN = \sum_{i=0}^{T} (x_i = 0, y_i = 0)$$

$$YY = \sum_{i=0}^{T} (x_i = 1, y_i = 1)$$

$$YN = \sum_{i=0}^{T} (x_i = 1, y_i = 0)$$

$$NY = \sum_{i=0}^{T} (x_i = 0, y_i = 1)$$

ρ_{kappa}, the agreement matrix based on kappa coefficients, is obtained using the above formulas as follows:

$$\rho_{kappa} = \begin{pmatrix} \kappa_{11} & \kappa_{12} & \dots & \kappa_{1n} \\ \kappa_{21} & \kappa_{22} & \dots & \kappa_{2n} \\ \kappa_{n1} & \kappa_{n2} & \dots & \kappa_{nn} \end{pmatrix}$$

where κ_{XY} is the kappa coefficient for algorithm X against algorithm Y.

Once the kappa matrix has been computed, it is necessary to consider the significance of obtained agreement values between any pair of algorithms. Landis and Koch [73] give the following table for interpreting the significance of the κ value. Although inexact, this table provides a useful benchmark on the significance of the above matrix.

κ	Interpretation
Negative	Poor agreement
$0.0 \leq 0.20$	Slight agreement
$0.21 \leq 0.40$	Fair agreement
$0.41 \leq 0.60$	Moderate agreement
$0.61 \leq 0.80$	Substantial agreement
$0.81 \leq 1.00$	Almost perfect agreement

Based on the results and table above, the minimum agreement threshold based on kappa T_A^{kappa} can be deduced. This is the value that is used in the next step of the framework to identify nearest neighbors for each algorithm based on the strength of the relationships.

As it will be shown in the next chapter that discusses simulation results, either correlation or kappa coefficient may be used to deduce the strength of relationship between any two algorithm pairs within the candidate set. That is, $\rho = \rho_{correlation} \approx \rho_{kappa}$. If one opts to use kappa coefficient matrix as the agreement evaluator, then based on the table above, the minimum threshold (T_A) for a relationship to exist between any two algorithms may be set to 0.5.

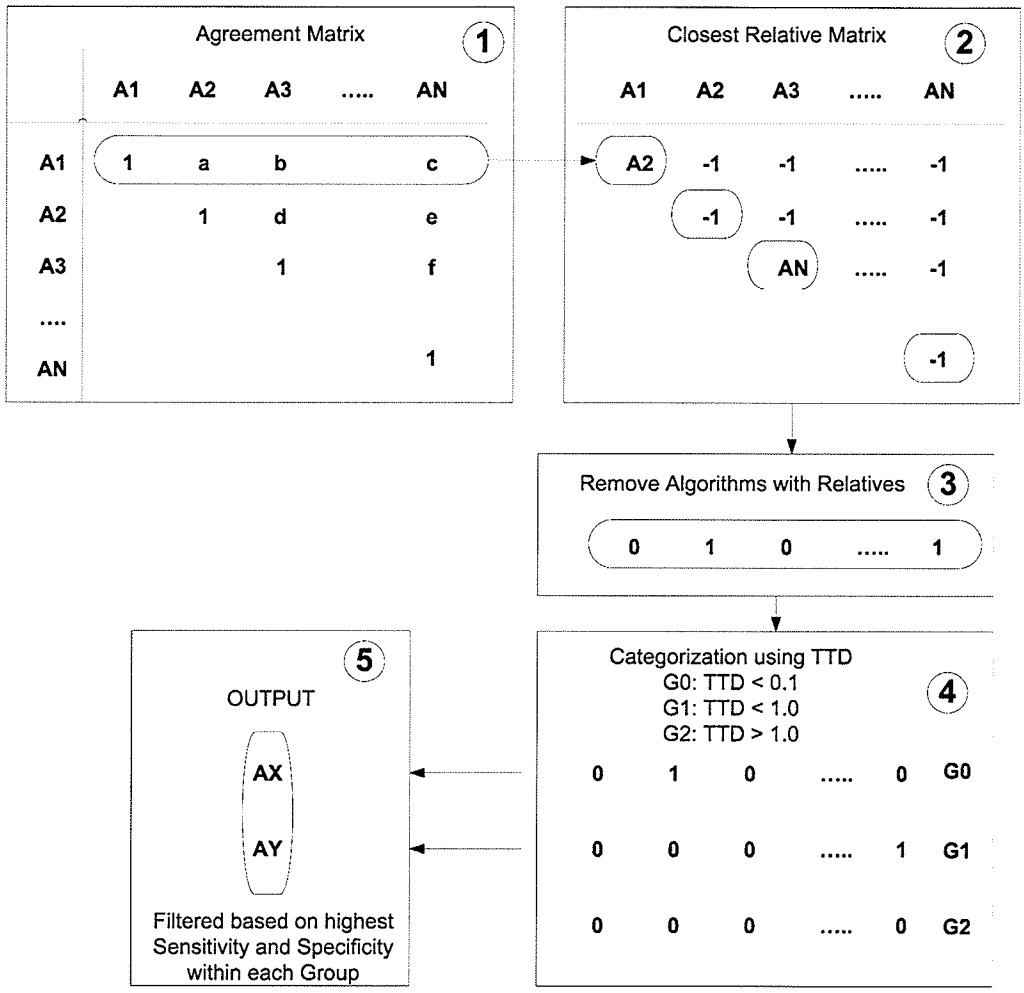

Figure 5.6: Minimal Set Identification Process

5.3.3 Step 3: Minimal Set Identifier

Once the sensitivity, specificity and time to detect parameters are well established for each algorithm and the agreement levels between every possible algorithm pair is known, a minimal set of algorithms can be identified that would be sufficient to produce quantifiable confidence value for the overall decision. Fig 5.6 illustrates a five-step process developed to identify this minimal set based on results from the previous two steps of the proposed framework.

- *Task 1*: This task is basically setting up the agreement matrix ρ generated from Step 2 of the framework. That is, initialize ρ with computed $\rho_{correlation}$ or ρ_{kappa} values. Note that only the upper triangle of the matrix needs to be analyzed to avoid any recursive relationships between two algorithms. That is, if A1 highly correlated to A2, then A2 is highly correlated with A1.

- *Task 2*: The next task deals with setting up the closest relative matrix. A closest relative to a specific algorithm X is algorithm Y that has an agreement value of at least some minimum agreement threshold (T_A) and has the highest agreement value with respect to X against all other algorithms within the set. The idea is that for each algorithm in the set, a corresponding algorithm with highest agreement value must be identified. It is entirely possible that a specific algorithm will not have a closest relative. In that case, the algorithm would be considered as an **independent** and thus needs to be included for next filtration task. For example, in the illustrated figure, A2 is closest relative to A1 as AN is to A3. However, algorithms A2 and AN are independent.

- *Task 3*: This task simply formalizes the algorithms that were selected in the previous task by removing all the algorithms from the closest relative matrix that have relatives identified, that is, the non-independent algorithms. This produces a **working set** of algorithms identified as **1** in the 1xN matrix.

- *Task 4*: The next task is to categorize the algorithms from the working set into three groups based on TTD value. The TTD was divided into three sets: close to zero days (TTD $\leq$ 0.1), less than one day (0.1 $\leq$ TTD $\leq$ 1.0) and greater than one day (TTD $\geq$ 1.0). This categorization makes

intuitive sense because typically one would be interested in TTD value of less than a day. Optimally, TTD should be as close to zero as possible, but realistically, public health individuals typically identify an outbreak more than a day later.

- *Task 5*: Once the groups have been identified, the final task deals with identifying the minimal set of algorithms through one more stage of filtration using specificity and sensitivity values obtained from step one of the framework. This task scans through each of the groups and attempts to flag algorithms that have both highest sensitivity and highest specificity when compared to other algorithms in the same group. If one algorithm has higher sensitivity but some other algorithm has higher specificity, then both the algorithms need to be considered.

This step of the framework yields a minimal subset of candidate algorithms that have minimal relation with each other and thus, form close to an independent minimal set that would be sufficient to deduce a confidence measure for an outbreak decision for a given day.

5.3.4 Step 4: Point-based Confidence Evaluator

The final step of the proposed framework deals with pulling together the findings from the first three steps and working out a scheme that produces a value that corresponds to overall confidence. There are three main parameters that need to be investigated.

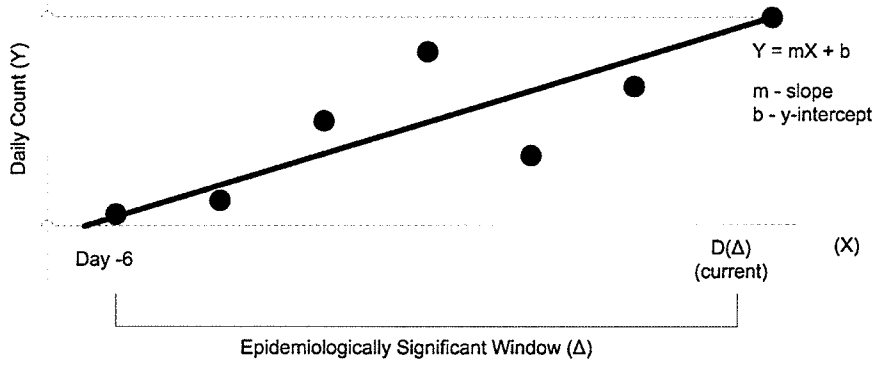

Figure 5.7: Rise Rate Analysis

5.3.4.1 Parameters of Interest: Rise Rate

The first parameter is the rate of change (referred to as **rise rate**) of actual daily count values over a specific time period, which provides some basic knowledge of the positive or negative trend over the last few days and also yields the speed with which the change is occurring.

Fig 5.7 illustrates a typical snapshot from daily counts data where the y-axis represents daily raw count and the x-axis represents the day with $D(\Delta)$ representing the current day. The rate of change (λ) is computed using basic linear regression method [74] to define a line that fits the daily count values in best possible manner:

$$\lambda = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

where n is the number of points being considered, x is the day and y is the count.

To be effective, the computation of rate is limited to a specific time frame

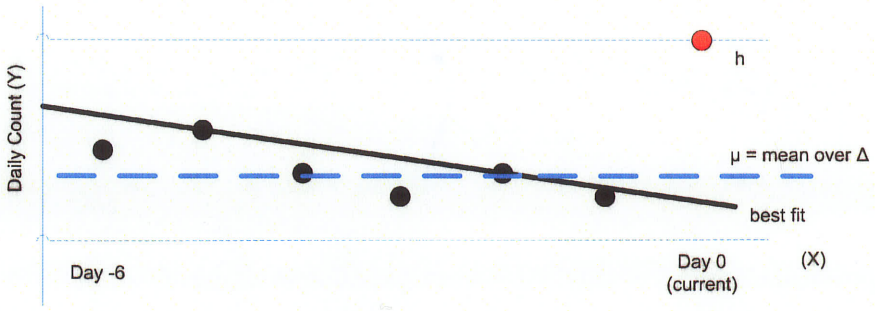

Figure 5.8: Count Delta

referred to as an epidemiologically significant window, Δ, which is defined in number of days.

5.3.4.2 Parameters of Interest: Count Delta

Next parameter of interest in analyzing the importance of the current day's count with respect to Δ. That is, does today's count follow a typical trend identified by the linear regression or is it drastically different and thus deserves special attention. As shown in Fig 5.8, there could be a scenario where past (Δ - 1) values yield a negative direction, however current day's value (h) is very high but cannot influence the linear regression formula to produce a positive slope which is more accurate in this case.

For such cases, the framework takes into account a second parameter of interest called **count delta** (ω). This value is simply the ratio between current day value, h, and the average value over Δ.

$$\omega = \frac{h}{\frac{1}{\Delta}\sum_{i=I-\Delta+1}^{i=I} X_i}$$

where I is the current day and X_i is the time series for daily counts.

5.3.4.3 Parameters of Interest: Outbreak Decisions

Based on the output of step three of the framework, the individual outbreak decision flags need to be considered. These provide the third parameter of interest, ϕ_i, where i refers to the algorithms in the minimal set. Each ϕ_i can have one of two values: **true** representing an outbreak has been detected by algorithm i and **false** representing no outbreak decision by algorithm i.

5.3.4.4 Point System: Rules

The overall objective of the framework is to produce a set of algorithms, that is as minimal as possible, to evaluate an aberration decision for any given day with some confidence value. Due to availability of multiple algorithms, a system that facilitates incremental confidence building based on contributions from various algorithms needs to be developed. A bimodal approach to confidence evaluation is proposed to address this issue as shown in Fig 5.9.

This bimodal approach is based on the concept of contribution to positive and negative confidence of a decision. The fundamental premise of the proposed scheme is a rule set, which is defined as the set of rules that collectively contribute to either positive or negative confidence. Positive confidence is a measure of collective strength of rules that contribute to a decision that supports identification of start of an outbreak. On the other hand, negative confidence is a measure of collective strength of rules that contribute to a decision that is against the decision of start of an outbreak. Rule sets are made of weighted combination of

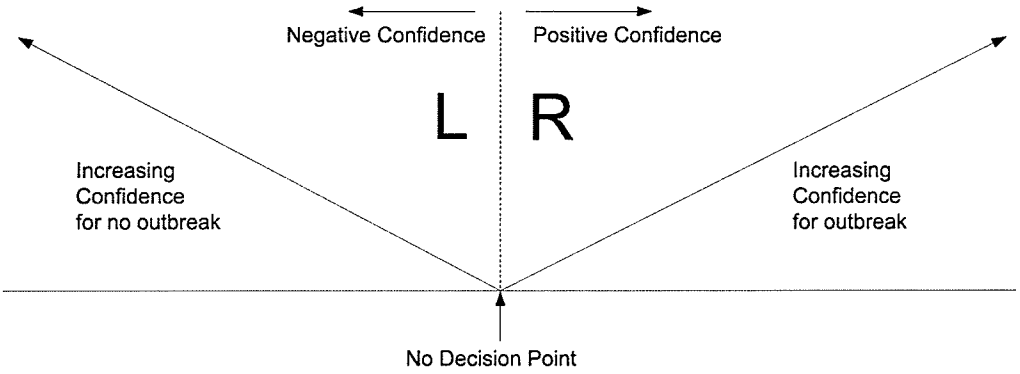

Figure 5.9: Point Assignment Scheme

identified parameters of interest. Further discussion on details of rule sets will follow shortly. Once the rule set has been identified, appropriate weights (or points) are assigned to the members of the rule set contributing to either side. A set of rules that contribute to positive confidence by collective summation of all of their respective points (p) are referred to as the **R** set. On the contrary, a set of rules that contribute to negative confidence by collective summation of all of their respective points (n) are referred to as the **L** set. That is, each side adds its collective contribution followed by $(p - n)$ to come up with overall confidence with **0** as the *no decision point.*

The following rules contribute to incremental positive confidence (**R** side rules):

$$\begin{bmatrix} \phi_i = true & \forall i \in K \\ \lambda_d > T_u * \lambda_{d-1} & \\ \omega_d > T_u * \omega_{d-1} & \end{bmatrix}$$

where d is the current day and K is the number of algorithms in the minimal set.

That is, there are $K + 2$ rules that contribute to positive confidence with each rule having a point magnitude of p_k, where k $\in$ (K+2).

The following rules contribute to incremental negative confidence (**L** side rules):

$$\begin{bmatrix} \phi_i = false \quad \forall i \in K \\ \lambda_d < T_d * \lambda_{d-1} \\ \omega_d < T_d * \omega_{d-1} \end{bmatrix}$$

where d is the current day and K is the number of algorithms in the minimal set. That is, there are $K + 2$ rules that contribute to positive confidence with each rule having a point magnitude of p_k, where k $\in$ (K+2).

The use of λ and ω requires introduction of some threshold value that defines the decision points in both the upward and downward directions. Thus, the scheme makes use of T_u parameter for the positive (or upside) threshold value and T_d for the negative (or downside) threshold value. Both of these values can be computed using sophisticated approaches like neural networks, however, a simple intuitive approach using hysteresis (Fig 5.10) was adopted. That is, λ and ω would contribute to positive confidence if the current day values were at least T_u times bigger than the previous day values. However, they would only contribute to negative confidence if the current day values were less than T_d times previous day values. This approach assists in identifying abrupt rises and falls in the count values with respect to immediate history. The proposed rule of thumb is to use $T_u \approx 3 * T_d$.

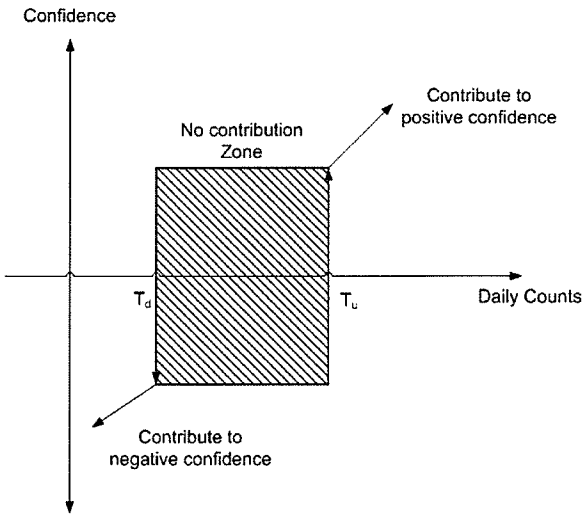

Figure 5.10: Threshold Hysteresis

To summarize, there are total of $Z = 2(K+2)$ rules that define a specific rule set ζ_i for a given point assignment i. In an attempt to simply the representation of rules and associated point assignments for **L** and **R** rules, a concise convention was designed as follows:

$$\zeta_i = \left\langle 1^{R_{p1}}_{L_{p1}}, 2^{R_{p2}}_{L_{p2}}, 3^{R_{p3}}_{L_{p3}}, 4^{R_{p4}}_{L_{p4}}, 5^{R_{p5}}_{L_{p5}}, \ldots, Z^{R_{pZ}}_{L_{pZ}} \right\rangle$$

where numbers 1 to Z represent the Z rules, L_{pZ} is the point assignment for the Zth Left rule and R_{pZ} is the point assignment for the Zth Right rule.

With $\frac{Z}{2}$ possible rules on each side, the most obvious choice is a balanced system with the maximum number of points for negative confidence and the maximum number of points for positive confidence to equal multiple of $\frac{Z}{2}$. That is, if both sides matched in their outcomes, then the overall confidence value would equate to 0, an indecisive line. To facilitate wider base of different points and

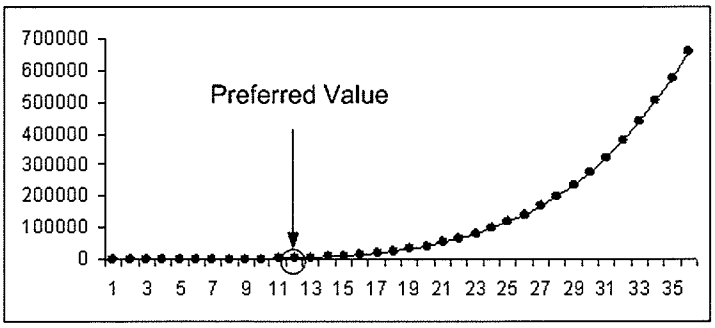

Figure 5.11: Maximum Number of Points

associated effects on overall decision, a system that exercises the point assignment with an unbiased (random) allocation of points is necessary. However, before such a system can be developed, the value of maximum points for each side (M) needs to be established. This can be achieved as follows:

$$\sum_{i=1}^{Z} p_i = M$$

where p_i represents point allocation for i^{th} rule.

In Fig 5.11, x-axis represents M and y-axis represents the total number of point assignment possibilities for $Z = 12$ (that is, $K = 6$). In this specific case, $M = 12$ seems reasonable as it is located at the knee of the rising curve and provides **6188** assignment possibilities, a number that is quite reasonable for simulation purposes.

Now that the rules and point assignment method have been designed, there is a need for devising a system that interprets outcomes of the application of identified rules and associated points and yields an optimal point assignment that produces desired outcome. The proposed approach is to group obtained

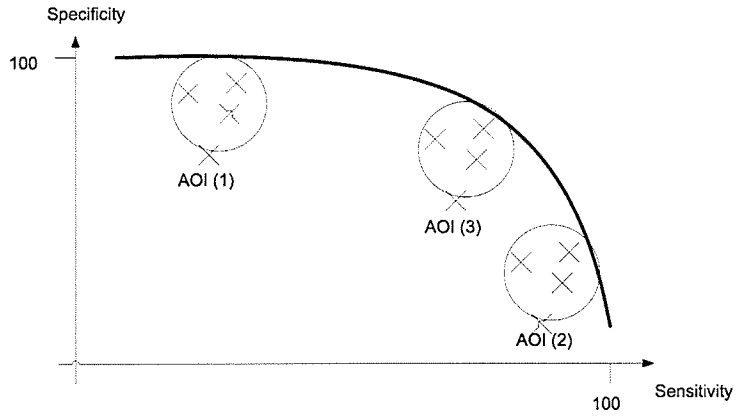

Figure 5.12: Clusters

points on the scatter sensitivity versus specificity plot into clusters of interest as shown in Fig 5.12. The idea is to identify specific areas of interest *(AOI)* on this scatter plot that produce outcome that is superior when compared to any single algorithm. That is, three AOIs are identified as follows: high specificity (left top); high sensitivity (bottom right) and maximum sensitivity/specificity (knee).

Any of the commonly used clustering techniques may be used to identify AOIs. The proposed approach utilizes *k-means* clustering [68] technique as it allows identification of initial centroids of desired clusters, which is attractive since, as discussed above, typically one would like to look at very specific clusters that provide, for instance, high specificity and high sensitivity - that is, AOI(3).

The objective of k-means approach is to minimize total intra-cluster variance, or, the squared error function:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

where there are k clusters $S_i(\forall i \in k)$, x_j is the sensitivity/specificity pair on the

scatter plot corresponding to ζ_i and μ_i is the centroid or mean point of all the points within cluster i.

Application of clustering methodology yields a multitude of rule sets ζ_i each of which produce a sensitivity/specificity pair ν_{ζ_i} yielding:

$$\psi_k = \{\nu_{\zeta_i}\}, \forall_{i \in k}$$

Once ψ_k has been figured out, the idea is to then pick an appropriate rule set in a given cluster k that falls in the desired AOI and use it for computing the overall confidence value. Note that one could develop an algorithm to identify an optimal point assignment within a cluster.

5.4 Nomenclature

In summary, the proposed CAIF framework utilizes multiple variables as follows:

- N is the number of algorithms in the candidate set.
- ρ is the agreement matrix between all pairs of algorithms within the candidate set.
- T_A is the minimum agreement threshold used to identify nearest neighbors.
- K is the number of algorithms in the minimal set.
- Z is the total number of positive and negative rules.

- M is the maximum number of points typically a multiple of $\frac{Z}{2}$.
- T_u is the positive (or upside) threshold value for point assignment scheme.
- T_d is the negative (or downside) threshold value for point assignment scheme.
- Δ is the epidemiologically significant window in days.
- λ is the rate of change of actual daily count values over a specific time period Δ.
- ω is the relation of the current day's count with respect to Δ.
- ϕ_j is the individual algorithm's outbreak decision flag based for a specific algorithm j within the minimal set.
- ζ_i is the rule set based on minimal set and specific point assignment i.
- ν_{ζ_i} is the sensitivity/specificity pair computed for a specific rule set i.
- ψ_k is a set of sensitivity/specificity pairs computed for all point assignments within a cluster k.

Based on this list, the following set, referred to as CAIF Parameters, needs to be populated using various steps of the framework:

$$CAIF\ Variables = \{N, \rho, T_A, K, Z, T_u, T_d, \Delta\}$$

with following parameters:

$$CAIF\ Parameters = \{\lambda, \omega, \phi_j\}$$

and following output values:

$$CAIF\ Outputs = \{\zeta_i, \nu_{\zeta_i}, \psi_k\}$$

Using the above nomenclature, the proposed four-step framework can be outlined as follows:

CAIF Framework
Step 1: (a) Identify outbreak data set(s) (b) Initialize candidate algorithm set Define N (c) Compute sensitivity, specificity and time-to-detect for each algorithm
Step 2: Compute agreement analyzer $\rho \leftarrow (\rho_{correlation}$ or $\rho_{kappa})$ Define ρ and T_A
Step 3: Execute Minimal set identification process Define K, Z and M
Step 4: (a) Setup inputs to point assignment scheme: Define T_u, T_d, Δ (b) Compute λ, ω and ϕ_j (c) Execute randomized strategy to obtain ζi Compute specificity/sensitivity pairs ν_{ζ_i} (d) Apply clustering technique(s) to generate ψ_k (e) Compute overall confidence value utilizing one of the rules sets in ψ_k

5.5 Summary

A novel aberration interpretation framework has been proposed for producing a confidence based system decision focusing on high confidence values at the start of an outbreak. The framework comprises of multiple steps to allow identification of a subset of algorithms as well as a dynamic point assignment scheme for computing a balanced decision.

The proposed framework provides a multitude of benefits:

1. Savings in the computation effort by identifying only a smaller subset of algorithms that are necessary and sufficient for a sound system decision.
2. Provides a mechanism to derive confidence value based on dynamic point assignment system.
3. Produces a superior overall system decision within desired AOI when compared to any single algorithm.
4. Provides a framework for future research to investigate optimal point allocation systems as well as analysis of new algorithms and their effects on the overall decision.

The next chapter presents an application of proposed scheme along with results obtained from a simulator developed during the course of this research. The method is also adaptable or extensible. The framework introduced here captures the essential elements of a confidence based decision process.

Chapter 6

Simulation Results

6.1 Environment

A simulation environment was setup using Borland C++ Builder Professional Version 6.0 development tools running on Windows XP platform. A graphical user interface was designed using the available components within the development suite. Details on the actual simulator have been provided later in this chapter.

In addition to a custom simulator, an open source statistical package **R** was used to perform some of the required statistical analysis. This package is available online at [69].

6.1.1 Simulator

A graphical user interface based simulator was custom developed for the purposes of this study. The simulator was developed in stages reflecting the four steps of the proposed framework. Generic routines to process input data files were developed to enable automated processing of large number of files using inherent C++ file and string manipulating functions.

Fig 6.1 illustrates a screenshot from the CAIF Simulator developed during the course of this research. A graphical interface window to allow setting of various parameters was also developed that would simply modification of the key parameters without requiring code change and recompilation. Specifically, the number of data sets to use in Step 1 and Step 2 via two digit $F1$ and $F2$ parameters with values ranging from 00 to 99. The simulator composes filenames for the datasets using these parameters ($sF1\_F2.csv$) and thus, requires the data sets to be present in the directory serving as the repository for the data sets. This allows for automated file processing of up to 10,000 files.

6.2 CAIF Framework Steps

Details of the specific parameters within the simulator will be touched upon while discussing each of the steps separately. The following sections provide results and detailed discussions corresponding to each of the steps of the proposed CAIF framework.

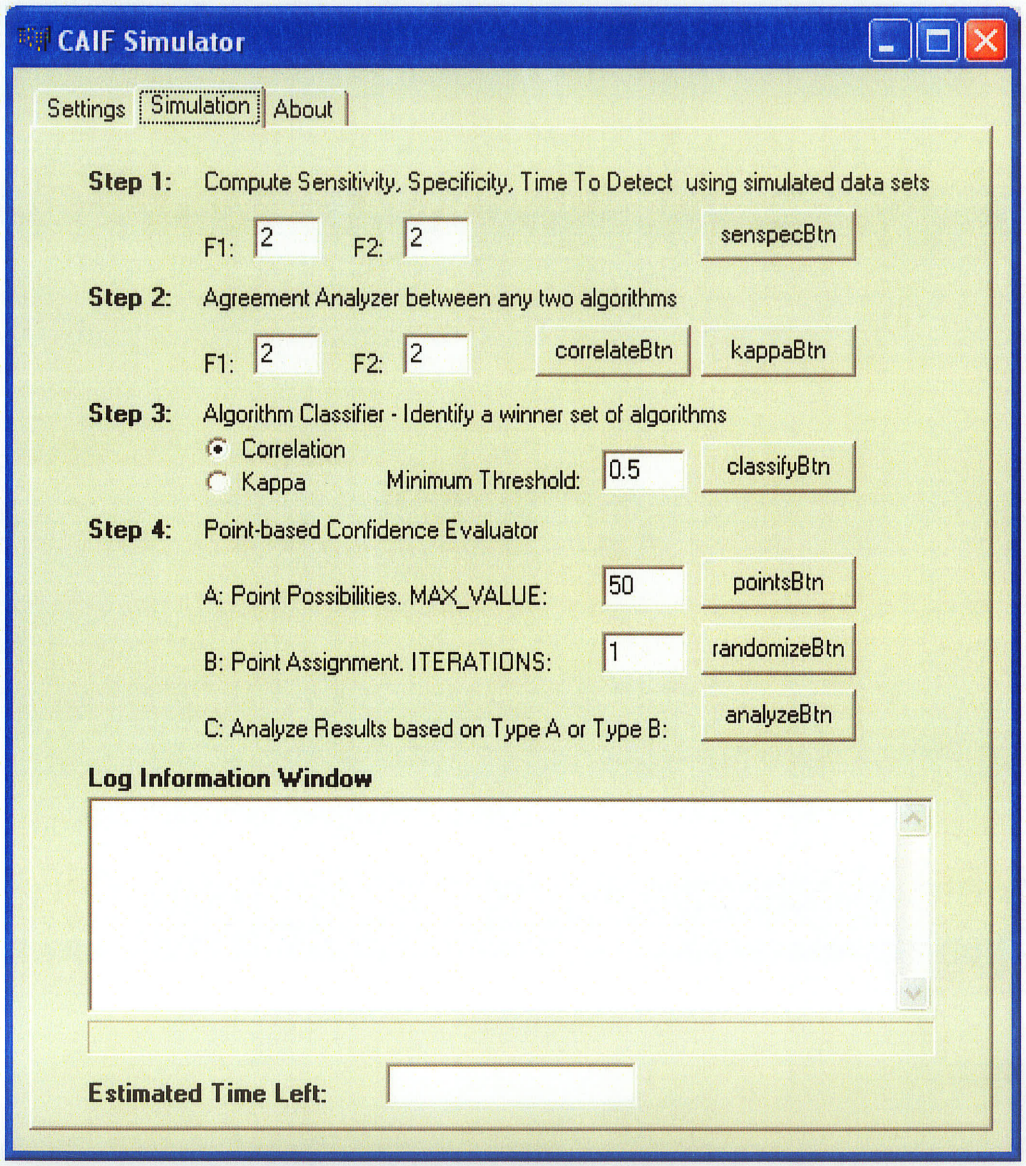

Figure 6.1: CAIF Simulator

6.2.1 Step 1a: Simulated Data Sets

In order to enable the computation of various key metrics defined within the framework (sensitivity, specificity, and time to detection), there was a need to either generate or obtain simulated outbreak data as real-life outbreak data is in limited supply. Fortunately, CDC [70] provides a large number of simulated data files each containing 100 iterations of 6 years of daily data, 1994-1999, using a negative binomial distribution with superimposed outbreaks. Outbreaks of various shapes and sizes are randomly placed throughout the data including 1-day spikes and log normal distribution based outbreaks.

6.2.2 Step 1b: Identify Candidate Algorithms

The following nine algorithms were implemented within the CAIF simulator as a candidate algorithm set.

- Three-day Moving Average (3MA): A simple three day moving average with period equal to three and x_D as count for Dth day with D = 1 as the current day. An outbreak is flagged when 3MA sum exceeds a threshold value of $\mu + 2\sigma$, where μ and σ are computed based on historical data.

$$3MA = (x_1 + x_2 + x_3)/3$$

- Five-day Moving Average (5MA): A simple five day moving average with period equal to five and x_D as count for Dth day with D = 1 as the current day. An outbreak is flagged when 5MA sum exceeds a threshold value of

$\mu + 2\sigma$, where μ and σ are computed based on historical data.

$$5MA = (x_1 + x_2 + x_3 + x_4 + x_5)/5$$

- Seven-day Moving Average (7MA): A simple seven day moving average with period equal to seven and x_D as count for Dth day with D = 1 as the current day. An outbreak is flagged when 7MA sum exceeds a threshold value of $\mu + 2\sigma$, where μ and σ are computed based on historical data.

$$7MA = (x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)/7$$

- Weighted Moving Average (WMA): A variant of the simple moving average with each day weighed differently, with the current day having the largest weight. The period was set to three days and x_D as count for Dth day and w_D as the weight for Dth day with D = 1 as the current day. An outbreak is flagged when WMA sum exceeds a threshold value of $\mu + 2\sigma$, where μ and σ are computed based on historical data.

$$WMA = (w_1x_1 + w_2x_2 + w_3x_3)/3$$

$$w_1 = 1/2, w_2 = 1/3 \; and \; w_3 = 1/6$$

- Exponentially Weighted Moving Average(EWMA): A variant of the simple moving average with the current day having most of the weight and an exponential decay of weights for the remaining days. The period was set to three days and x_D as count for Dth day with D = 1 as the current day and

weight, w, set to 0.5. An outbreak is flagged when EWMA sum exceeds a threshold value of $\mu + 2\sigma$, where μ and σ are computed based on historical data.

$$EWMA_{today} = wx_D + (1 - w)EWMA_{yesterday}$$

- Cumulative Sums (CUSUM): Commonly used syndromic surveillance algorithm implemented with K-parameter set to 1; D = 1 is the current day; μ and σ are calculated from the entire data set (that is, all available data). An outbreak is flagged when the CUSUM exceeds the H-parameter (decision interval). In this case H was set to 3.

$$CUSUM_{today} = max(0, CUSUM_{yesterday} + (z - K)),$$

$$Z = (x_D - \mu)/\sigma$$

- Early Aberration Reporting System (EARS) - C1: CDC based syndromic surveillance algorithm implemented with K-parameter set to 0.1; D = 1 is the current day; μ and σ are calculated from the previous 7 days, D_{-1} to D_{-7}. An outbreak is flagged when C1 exceeds the H-parameter (decision interval). In this case H was set to 2.

$$C1 = max(0, (x_D - \mu - K\sigma)/\sigma)$$

- Early Aberration Reporting System (EARS) - C2: A variant of C1 implemented with K-parameter set to 0.1; D = 1 is the current day; μ and σ

are calculated from 9 previous days with a two day lag, D_{-3} to D_{-9}. An outbreak is flagged when C2 exceeds the H-parameter (decision interval). In this case H was set to 2.

$$C2 = max(0, (x_D - \mu - K\sigma)/\sigma)$$

- Early Aberration Reporting System (EARS) - C3: A variant of C1 which includes the previous days sum; implemented with K-parameter set to 0.1; D = 1 is the current day; μ and σ are calculated from 7 previous days with a two day lag, D_{-3} to D_{-9}. An outbreak is flagged when C2 exceeds the H-parameter (decision interval). In this case H was set to 2.

$$C3_{today} = max(0, C3_{yesterday} + (x_D - \mu - K\sigma)/\sigma)$$

Note that a parameters tab had to be developed within the simulator to enable setting of the parameters discussed above for each of the algorithms in the candidate set.

6.3 Step 1c: Specificity, Sensitivity and Time To Detect Evaluation

The sensitivity, specificity and time to detect parameters were computed using formulas discussed in the previous chapter. The final values for each algorithm were obtained by averaging the results from ten input files each containing over 200,000 days of simulation outbreak data.

The following table lists mean values of sensitivity, specificity and time to detect parameters for each of the algorithms executed over simulated data sets.

Algorithm	Time To Detect (days)	Specificity (%)	Sensitivity (%)
3MA	1.87	99.35	48.07
5MA	3.35	99.53	34.32
7MA	4.30	99.49	23.02
WMA	1.33	99.17	52.12
EWMA	2.87	98.98	45.77
CUSUM	0.83	88.52	53.98
EARS C1	1.27	92.67	86.25
EARS C2	0.79	92.90	88.36
EARS C3	0.76	81.62	96.13

Visualizing these values in a form of a line chart (Fig 6.2), we can clearly identify the difference between moving average algorithms (3MA, 5MA, 7MA, WMA, EWMA) and cumulative sum based algorithms (CUSUM, C1, C2, C3). That is, moving average algorithms seem to have very high specificity but also take longer to detect true positives and thus might miss an entire outbreak if it is over a short span. On the other hand, CUSUM based algorithms compromise specificity to yield better sensitivities and thus produce comparable specificities and sensitivities.

This is an expected observation since moving average algorithms are designed to detect shifts in counts based on immediate past and thus depending on the averaging window size they might miss short-lived outbreaks yielding poor sensitivities. That is, they will detect most of the true negatives (high specificity)

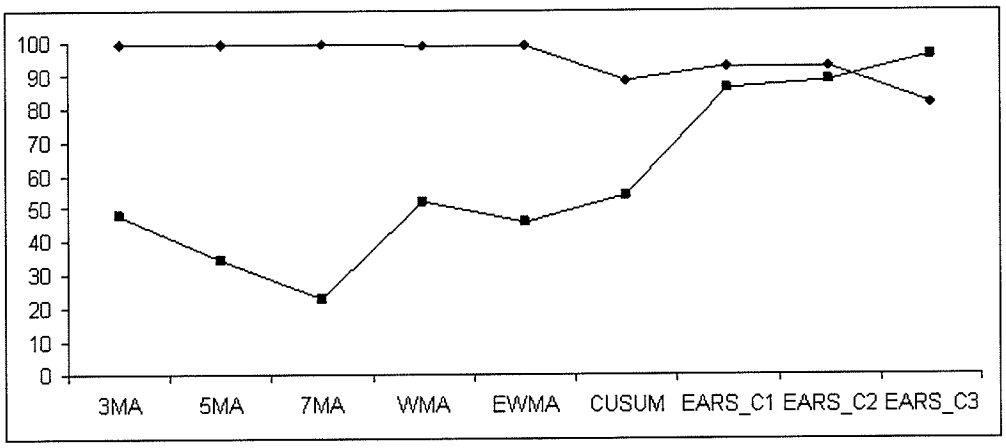

Figure 6.2: Mean Sensitivity and Specificity Values

as well as they will yield a high number of false negatives (low sensitivity). On the other hand, CUSUM based algorithms provide a trade off between the two parameters due to their inherent implementation of analysis based on the differential values of the mean. It is worthwhile to note that the lower sensitivity for CUSUM algorithm is probably a result of setting the H parameter to a value of 3. Furthermore, higher specificity for C1, C2 and C3 algorithms can probably be attributed to the fact that the outbreak trigger is governed by a much shorter time period (between seven to nine days) as compared to moving average algorithms which base their trigger points (mean and standard deviation) on the entire data set.

6.3.1 Step 2: Agreement Analysis

Now that the sensitivity, specificity and time to detect parameters were computed for each algorithm, the next task was to identify a relationship evaluator matrix. Both the correlation and kappa coefficient approaches were implemented within the simulator.

First, a correlation matrix was generated using 600 years of simulated data. The diagonal of this matrix always consists of ones. This is because these are the correlations between each variable and itself (and a variable is always perfectly correlated with itself). A correlation matrix is always a symmetric matrix.

	3MA	5MA	7MA	WMA	EWMA	CUSUM	C1	C2	C3
3MA	1.00								
5MA	0.92	1.00							
7MA	0.83	0.95	1.00						
WMA	0.98	0.93	0.84	1.00					
EWMA	0.17	0.33	0.48	0.21	1.00				
CUSUM	0.41	0.33	0.25	0.36	-0.12	1.00			
C1	-0.22	-0.20	-0.03	-0.22	0.32	-0.16	1.00		
C2	-0.27	-0.29	-0.31	-0.28	0.31	-0.14	-0.03	1.00	
C3	-0.34	-0.35	-0.34	-0.35	0.31	-0.18	-0.01	0.96	1.00

In order to get an appreciation of types of relationship that exist between different algorithm pairs, three specific examples were plotted as shown in Fig 6.3. For example, the correlation between 3MA and EWMA algorithms, although positive, was quite weak. Detailed analysis showed that most of the agreement between these algorithms was confined to the lower half of the parametric values. On the other hand the relationship between 5MA and 7MA was very strong and seemed to match throughout the spectrum. Finally, an example of a poor relationship was that between 7MA and C3 algorithms. This was expected because in contrast to moving average algorithms, CUSUM algorithms are based on very different metrics.

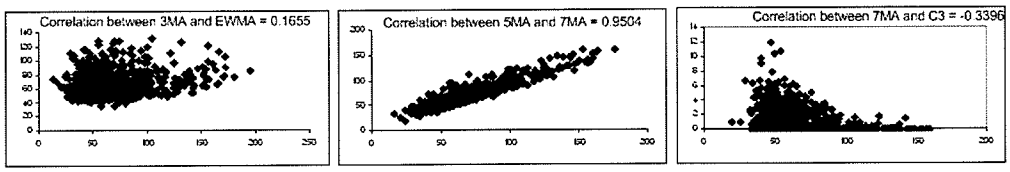

Figure 6.3: Correlation Examples

Next, significance values for each of the pairs had to be computed. Using the t-test formula from the previous chapter, any value greater than **0.16** in the correlation matrix clears the two-tailed critical value (1.644) for probability of less than *0.05*. This suggests that if $\rho_{correlation}$ was used as the agreement analyzer, then one could use $T_A = T_A^{correlation} = 0.16$ as the threshold value.

Next, considering the results for kappa coefficients. Similar to the previous coefficient matrix, the diagonal of this matrix always consists of ones. This is because these are the coefficients between each variable and itself (and a variable is always perfectly correlated with itself). A kappa matrix is also a symmetric matrix.

	3MA	5MA	7MA	WMA	EWMA	CUSUM	C1	C2	C3
3MA	1.00								
5MA	0.74	1.00							
7MA	0.58	0.79	1.00						
WMA	0.95	0.74	0.58	1.00					
EWMA	0.07	0.21	0.36	0.07	1.00				
CUSUM	0.21	0.23	0.23	0.21	-0.06	1.00			
C1	-0.19	-0.18	-0.05	-0.19	0.23	-0.09	1.00		
C2	-0.20	-0.19	-0.19	-0.20	0.27	-0.10	0.07	1.00	
C3	-0.28	-0.26	-0.15	-0.28	0.29	-0.12	0.10	0.63	1.00

An interesting thing to note is that the kappa matrix is very similar to the correlation matrix ($\rho_{correlation} \approx \rho_{kappa}$). That is, it produces similar level of agreement and disagreement between a set of algorithms as does the correlation matrix. This raises the comfort level significantly as the kappa matrix is based on actual decisions of the algorithms (that is, binary values), while the correlation matrix is based on actual counts generated by each algorithm before a decision is made. In the following steps, ρ is set to ρ_{kappa} with $T_A = T_A^{kappa} = 0.5$ based on the agreement guide provided by Landis and Koch [73].

So far, the following values for the CAIF variable list have been identified:

$$\{N = 9, \rho = \rho_{kappa}, T_A = 0.5\}$$

6.3.2 Step 3: Minimal Set Identification

The following list guides through each task of the third step of the proposed framework. Recall that the objective of this step is to generate the minimal set from candidate set of algorithms.

- *Task 1*: The candidate algorithm set was set to **[3MA, 5MA, 7MA, WMA, EWMA, CUSUM, C1, C2, C3]**.
- *Task 2*: Based on ρ, the CAIF Simulator produced a closest relative matrix of **[WMA, 7MA, WMA, -1, -1, -1, -1, C3, -1]**, where -1 represents an independent algorithm.
- *Task 3*: Based on the closest relative matrix, a working set of **[0, 0, 0, 1, 1, 1, 1, 0, 1]** was produced, where '1' implies that the corresponding algorithms were considered in the grouping task (Task 4).
- *Task 4*: Based on the working set obtained in the previous task, the CAIF Simulator yielded the following three groups in order: G1 = **[0, 0, 0, 0, 0, 0, 0, 0, 0]**, G2 = **[0, 0, 0, 0, 0, 1, 0, 0, 1]** and G3 = **[0, 0, 0, 1, 0, 0, 1, 0, 0]**. That is, recalling the definitions of groups G1, G2 and G3 from previous chapter, none of the algorithms in the candidate set had TTD value of close to zero ; CUSUM and C3 as algorithms with TTD value of less that one day; and WMA and C1 as algorithms with TTD value greater than one day.
- *Task 5*: Based the three groups produced in the previous task, the CAIF Simulator produced **[WMA, CUSUM, C1, C3]** as the minimal set. That is, the first group was empty set with no algorithms; the second group was

set to **[0, 0, 0, 0, 0, 1, 0, 0, 1]** because CUSUM and C3 complement each other in terms of sensitivity and specificity values; and finally the third group was set to **[0, 0, 0, 1, 0, 0, 1, 0, 0]** because WMA and C1 complement each other as well.

In summary, the final minimal set to be passed to next step was identified as **[WMA, CUSUM, C1, C3]**. It is quite interesting to note that all but one of the CUSUM based algorithms from the initial algorithm set were included in the final minimal set but only one out of five moving average based algorithms was present. This is due to the fact that all four CUSUM based algorithms are designed in slightly unique manner in terms of their use of baseline data, time lags and trigger points.

With this result, the CAIF variable list was updated to:

$$\{N = 9, \rho = \rho_{kappa}, T_A = 0.5, K = 4\}$$

6.3.3 Step 4: Point-based Confidence Evaluator

Based on the identified minimal set, the rule set (ζ_i) was defined as follows. The right-side positive confidence contributing rules:

$$\begin{bmatrix} 1^R : \phi_{CUSUM} = true \\ 2^R : \phi_{C3} = true \\ 3^R : \phi_{WMA} = true \\ 4^R : \phi_{C1} = true \\ 5^R : \lambda_d > T_u * \lambda_{d-1} \\ 6^R : \omega_d > T_u * \omega_{d-1} \end{bmatrix}$$

and the left-side negative confidence contributing rules:

$$\begin{bmatrix} 1_L : \phi_{CUSUM} = false \\ 2_L : \phi_{C3} = false \\ 3_L : \phi_{WMA} = false \\ 4_L : \phi_{C1} = false \\ 5_L : \lambda_d < T_d * \lambda_{d-1} \\ 6_L : \omega_d < T_d * \omega_{d-1} \end{bmatrix}$$

Note that there are 6 rules on each side, i.e. $Z = 6$. The obvious choice for $M = 12$ as discussed in the previous chapter as this value is at the knee of the point assignment possibilities curve and provides sufficient possibilities for a random point assignment strategy. This means that the sum of all points contributed by each rule on each side adds up to 12 (100% possitive confidence) or -12 (100% negative confidence).

Finally, the values for the two thresholds T_u and T_d needed to be defined. Based on some preliminary work on the simulated data sets, the values for T_u

and T_d were set to 1.15 and 0.5 respectively. That is, on average during peace time, both λ and ω stayed within fifteen percent between current day and previous day, thus value of T_u was set to $(1 + 0.15)$. Since the proposed value for T_d is approximately three time T_u, its value was set to 0.5. One could analyze these values and perform in-depth analysis for their optimal values using wide variety of techniques. This was not done here as the objective was to illustrate the overall framework as opposed to determining optimal values of parameters such as T_u and T_d. In the event the framework is considered to be of value, it would then be desirable to attempt to optimize its various components.

The CAIF variable list was now complete with the following values:

$$\{N = 9, \rho = \rho_{kappa}, T_A = 0.5, K = 4, Z = 6, T_u = 1.15, T_d = 0.5, M = 12, \Delta = 7\}$$

Once all the parameters were computed, the CAIF simulator was setup to perform many iterations to produced a large variety of point assignment using randomized point assignment strategy where only unique combinations of points for each set were allowed. This produced a scatter plot of specificity against sensitivity as shown in Fig 6.4. As expected, the points on the scatter plot, independent of the number of point assignments used, seem to cluster in three areas: top left, bottom right and knee of the curve.

Once the scatter plot was produced, as per the framework, using the R tool [69], k-means clustering was applied to identify points that lie within the desired AOIs. Numerous iterations of k-means were generated on 2500 point assignments with the number of centroids equaling 3, 10, 15 and 20 to produce clusters as shown in Fig 6.5 (a-d).

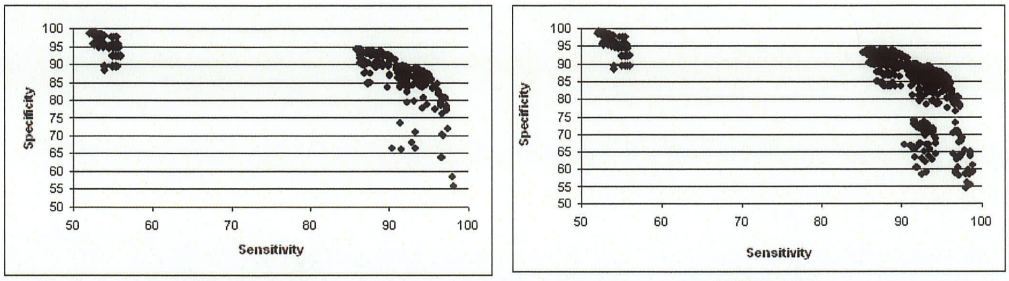

Figure 6.4: Minimal Set Scatter Plot

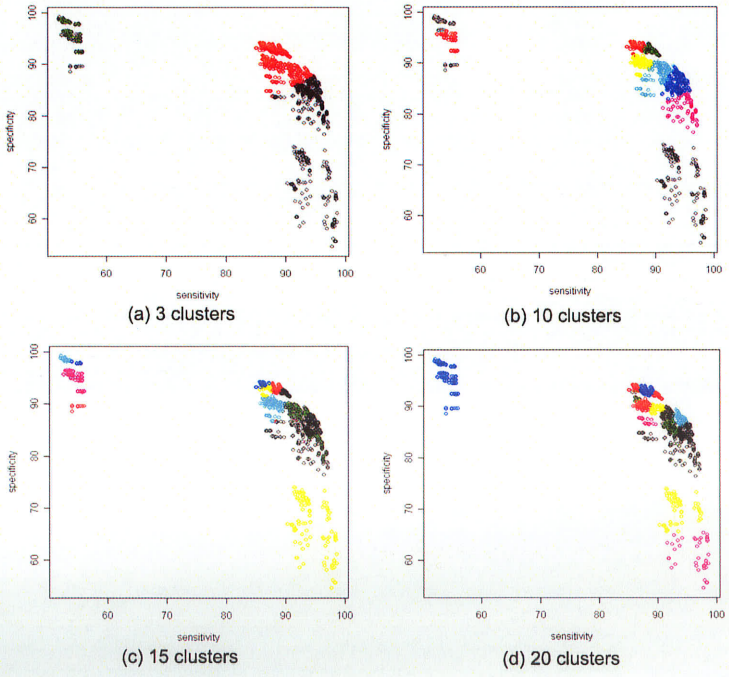

Figure 6.5: K-Means Clustering

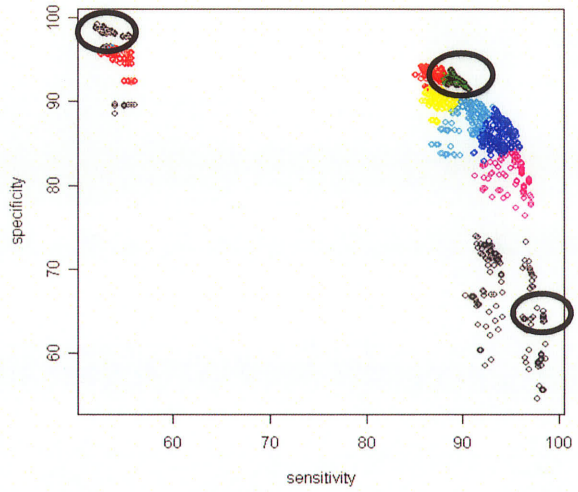

Figure 6.6: Identified Areas Of Interest

As illustrated in Fig 6.5 (a), there are clearly three areas of interest in agreement with what was discussed in the previous chapter. Adding a few more clusters as shown in Fig 6.5 (b), the separation within the larger three clusters becomes evident and the AOIs become more clear. Further adding more clusters does not seem to add any more visible differentiation.

Fig 6.6 illustrates three AOIs on a scatter plot as identified by using ten clusters. The centers for these ten clusters are listed in the following table.

Cluster	Specificity (%)	Sensitivity (%)
1	92.94	88.15
2	98.35	53.42
3	84.93	92.50
4	90.15	87.38
5	66.50	94.63
6	88.28	90.78
7	94.52	54.74
8	89.10	54.39
9	81.46	95.92
10	86.89	94.41

From the above list, the three clusters of interest representing the AOIs were **2**, **5** and **10** with the following centroids (98.35, 53.42), (66.50, 94.63) and (86.89, 94.41). For AOI(1), none of the point assignments provided a better result than simply running **WMA** algorithm which yielded (99.17, 52.12) as the specificity and sensitivity values. Thus, the conclusion was that the proposed framework does not provide any benefit in cases when highest possible specificity is desired. For AOI(2), the identified centroid of (66.50, 94.63) provided a cluster with about 125 point assignments. For example, the following rule set, obtained by randomly selecting a point assignment from a set of 125 possibilities within AOI(2), yielded sensitivity of 98.64% with specificity of 61.06%, a result that is not achievable by any one algorithm independently:

$$\left\langle 1_0^4, 2_3^4, 3_0^0, 4_0^0, 5_1^2, 6_8^2 \right\rangle$$

which translates to positive confidence associated with the following rules,

$$\begin{bmatrix} 1^R : \phi_{CUSUM} = true \rightarrow 4\, points \\ 2^R : \phi_{C3} = true \rightarrow 4\, points \\ 3^R : NA \rightarrow 0\, points \\ 4^R : NA \rightarrow 0\, points \\ 5^R : \lambda_d > T_u * \lambda_{d-1} \rightarrow 2\, points \\ 6^R : \omega_d > T_d * \omega_{d-1} \rightarrow 2\, points \end{bmatrix}$$

Negative confidence associated with the following rules,

$$\begin{bmatrix} 1_L : NA \rightarrow 0\, points \\ 2_L : \phi_{C3} = false \rightarrow 3\, points \\ 3_L : NA \rightarrow 0\, points \\ 4_L : NA \rightarrow 0\, points \\ 5_L : \lambda_d < T_u * \lambda_{d-1} \rightarrow 1\, point \\ 6_L : \omega_d < T_d * \omega_{d-1} \rightarrow 8\, points \end{bmatrix}$$

Note that each side of the rule set contributes a maximum of $M = 12$ points providing an overall confidence measure ranging from -12 (100% negative confidence) to +12 (100% positive confidence).

For AOI(3), the identified centroid of (86.89, 94.41) is quite close to the result produced by **EARS C3** algorithm. However, this cluster has over 200 point assignments some of which yield higher sensitivity and specificity values than **EARS C3** which provides the best pair from all algorithms in the candidate set.

For example, the following rule set yields (86.39, 95.50):

$$\left\langle 1_1^0, 2_6^0, 3_1^2, 4_0^3, 5_3^5, 6_1^2 \right\rangle$$

which translates to positive confidence associated with the following rules,

$$\begin{bmatrix} 1^R : NA \rightarrow 0\, points \\ 2^R : NA \rightarrow 0\, points \\ 3^R : \phi_{WMA} = true \rightarrow 2\, points \\ 4^R : \phi_{C1} = true \rightarrow 3\, points \\ 5^R : \lambda_d > T_u * \lambda_{d-1} \rightarrow 5\, points \\ 6^R : \omega_d > T_d * \omega_{d-1} \rightarrow 2\, points \end{bmatrix}$$

Negative confidence points associated with the following rules,

$$\begin{bmatrix} 1_L : \phi_{CUSUM} = false \rightarrow 1\, point \\ 2_L : \phi_{C3} = false \rightarrow 6\, points \\ 3_L : \phi_{WMA} = false \rightarrow 1\, point \\ 4_L : NA \rightarrow 0\, points \\ 5_L : \lambda_d < T_u * \lambda_{d-1} \rightarrow 3\, points \\ 6_L : \omega_d < T_d * \omega_{d-1} \rightarrow 1\, point \end{bmatrix}$$

Next, it was time to apply the computed parameters and one of the rule sets from desired AOI cluster (in this case AOI(3)) to a sample outbreak within the

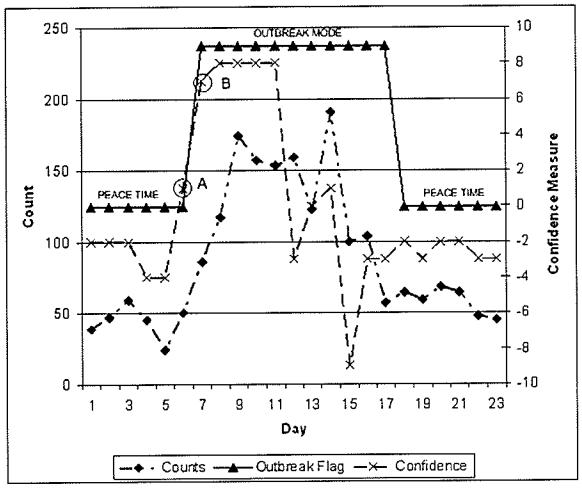

Figure 6.7: Simulated Outbreak Analysis

simulated data sets and confirm its effectiveness. (Fig 6.7) illustrates a snapshot that superimposes daily counts during outbreak mode along with computed confidence measure using the above rule set.

As shown, the framework suggests an outbreak day with confidence measure of $+\mathbf{1}$ ($\frac{1}{12}$ or 8.33% positive confidence) on day 6, a day before an outbreak is going to start (point A). Although a false positive decision, it is a weak false positive that aids in planning for the following day which will have a strong positive confidence measure of $+\mathbf{7}$ translating to $\frac{7}{12}$ or 58.3% positive confidence (point B). This is exactly what the aim of this framework was set to be, that is, identify start of an outbreak with some level of confidence measure. Further to note, as the outbreak progresses, the confidence seems to drop to negative values. This is because the framework is intended to monitor initial start of an outbreak. As the values stabilize during an outbreak, the confidence measure of start of an outbreak will diminish as expected.

6.3.4 Application of CAIF to a Real Scenario

Next, the same rule set from AOI(3) was applied to real emergency room visit data for the city of Winnipeg obtained from the Canadian Early Warning System [19, 58].

As shown in Fig 6.8, one of the key observations is that the indication that an outbreak is going to occur in the next few days was identified by a higher confidence value on Day 8, which was most likely the first day of an outbreak curve with peak on Day 11. Further, the confidence measure was computed based on an minimal set identified by the proposed framework and not the entire set of nine algorithms. That is, the minimal set identified by the proposed framework was sufficient to detect the start of an event a few days earlier than it was actually detected.

It is worthwhile to spend a little bit of time analyzing some of the days plotted on the chart.

- **Day 8**: Three of nine algorithms suggest an outbreak out of which two are from the identified minimal set. Looking at this at face value would produce a biased decision that we had no signs of start of an outbreak on day 8. However, considering only the minimal set, there is a split decision, and using the proposed point assignment system a confidence measure of **+5** translating to $\frac{5}{12}$ or 41.7% positive confidence is produced. Thus, there were clear signs for start of an outbreak on that day as suggested by a strong confidence value.

- **Day 9**: The confidence value drops drastically to just above the **0** or no

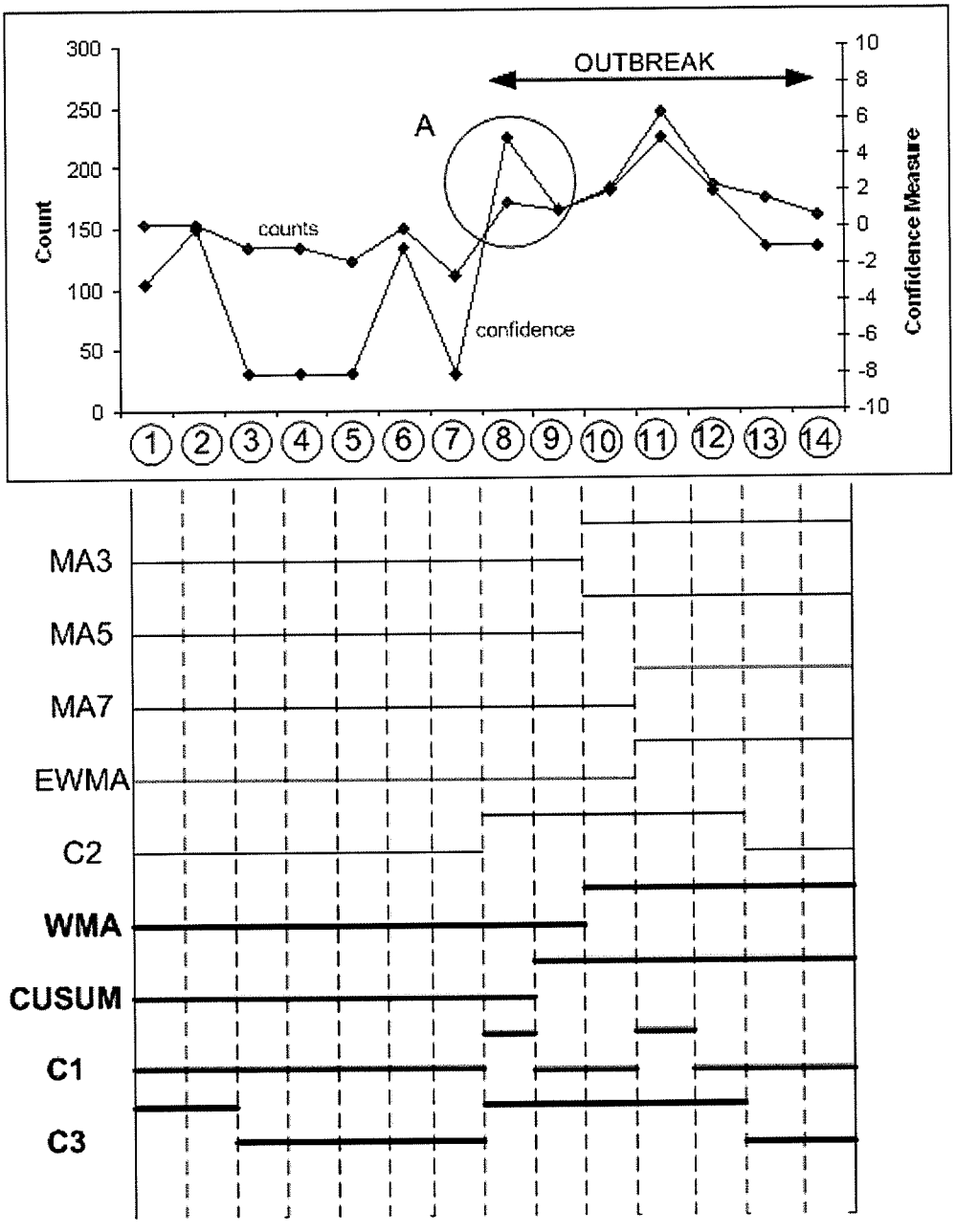

Figure 6.8: An Application of CAIF

decision line. This is due to the actual count staying at similar level as the count for previous day thus the δ and ω values did not change much and thus did not contribute to the overall confidence value as strongly as they did on the previous day. However, the confidence value still stayed above zero point indicating some level of activity.

- **Day 11**: This is the day when the counts of cases during an outbreak are the highest. All four algorithms of the minimal set declare an outbreak, however, the framework produces confidence measure of only $+5$. This is because the framework is monitoring start of an outbreak and not necessarily the peak. At the peak, both λ and ω do not contribute their portion to the overall confidence measure since neither the recent most count nor the count delta satisfy the rules as defined in the positive set.

Using the proposed framework, the identification with significant confidence would have been detected on Day 8 and initial start of some activity instead of delayed identification which occurred on Day 11.

6.4 Summary

In this chapter, the process of obtaining values of all CAIF parameters as well as results of application of CAIF approach using the computed parameters to both simulated and real outbreak scenarios were discussed at length. A simulation environment was setup that comprised of custom simulator for some aspects of the proposed approach as well as commercial and open source packages to compute various statistical and epidemiological parameters used in the proposed approach.

The data for simulation were obtained from CDC.

Nine candidate algorithms were selected based on literature review of most commonly used aberration detection algorithms, details of which were presented including the values of various parameters used by each of the algorithms. The epidemiological parameters (sensitivity, specificity and time to detect) were computed using the simulation environment. A discussion on the obtained values for these parameters in relation to each candidate algorithm was provided.

Two approaches for measuring the agreement between various candidate algorithms were considered: *correlation* and *kappa coefficients*. Both approaches were simulated and the resulting coefficient matrices were discussed. Both approaches provided similar outcomes and thus, it was a matter of selecting one approach. The decision was made to make use of kappa coefficients due to its acceptance within the epidemiological community.

Based on the agreement matrix and epidemiological parameters, the minimal set was computed. The five-step process was explained and the final set of four algorithms **(WMA, CUSUM, C1, C2)** was identified. Using this minimal set along with λ, ω and ϕ, the rule sets corresponding to three areas of interest (AOI) were identified. It was concluded that the proposed approach did not provide any gains in case of AOI(1) as **WMA** provided better result as a standalone algorithm. However, the approach provided significant gain when considering AOI(2) and AOI(3).

Finally, the proposed CAIF approach was applied to a real scenario generating very promising results for identifying start of an outbreak with high confidence measure thus, providing significant heads-up for a potential start of an outbreak.

It is important to note that the nine algorithms used as candidate set have various parameters that may be tuned to obtain better performance for particular types of outbreaks depending on incubation and type of infection mechanism. That is, the proposed framework is not intended to discourage optimization of existing algorithms but instead provide a platform to (1) evaluate algorithms against each other, (2) extract the complementary strengths of algorithms and (3) provide collective prediction for start of an outbreak.

Chapter 7

Alternate Methods

This chapter provides description of some of the commonly used techniques for data mining, pattern recognition, classification and prediction and includes results of their application to the problem of minimal set identification, that is, Step 3 of the proposed CAIF approach.

It is vital to note that these approaches are not typically applied to solve such a problem, however, careful definition of inputs and outputs along with proper interpretation of associated results provides excellent alternatives for minimal set identification process that can be compared against the proposed CAIF approach.

Please note that most of the descriptive text (excluding the results and the application of techniques to the problem) in this chapter related to existing techniques has been adapted from existing literature as they provide better explanation of these techniques. Effort has been made to present the description in a more applicable manner to this research, however, it was difficult to completely rewrite the text.

7.1 Bayesian Networks

A Bayesian network [75] is a model that encodes probabilistic relationships among variables of interest. It can be represented as a network of variables where the connection points (or arcs) between them are drawn from cause to effect.

Keeping in mind our objective of developing a meta-algorithm (or an envelope algorithm) that utilizes the strengths of multiple anomaly detection algorithms and not of devising a new stand-alone anomaly detection algorithm, it is quite difficult to identify variables of interest and their cause-effect relationships for building a bayesian networks. However, such approach is quite appropriate for a stand alone algorithm such as PANDA [66] described in one of the earlier chapters.

For example, within PANDA, a dynamic probabilistic causal model using a Bayesian network representation is created for each patient. Such a patient-specific causal model includes variables that represent risk factors (e.g., infectious disease exposures of various types), disease states, and patient symptoms. Some of the variables in each patient-specific network are linked (via arcs) to population variables, such as a variable that represents spatio-temporal information about the release of an infectious agent. Also, if the disease state of a patient P is potentially influenced by another patient Q (e.g., if Q has a contagious disease and P was exposed to Q), then the causal model for P would have arcs into it from the model for Q. The inter-linked patient-specific causal models form a large causal Bayesian network that represents the entire population being modeled.

As can be deduced from this example, it is difficult to identify such cause-effect relationships between algorithms and thus, bayesian networks were not investigated any further for the identification of minimal set.

7.2 Decision Trees

Next, the application of decision trees to identifying a minimal set was considered. Decision tree generation process can be viewed as a process of identifying those input variables that play a significant role in classifying most of the data correctly given values of input variables and expected output. The following description for decision trees has been extracted from [76, 77].

A decision tree is a logical model represented as a binary tree that shows how the value of a *target* variable can be predicted by using the values of a set of *predictor* variables. A *target* variable is the variable whose values are to be modeled and predicted by other variables. A *predictor* variable is a variable whose values will be used to predict the value of the target variable.

At a fundamental level, a decision tree comprises of **nodes** (Fig 7.1). Each node represents a set of records from the original dataset. Nodes that have child nodes are called **interior** nodes. Nodes that do not have child nodes are called **terminal** or **leaf** nodes. The topmost node is called the **root** node. A decision tree is constructed by a binary split that divides the rows in a node into two groups (child nodes). The same procedure is then used to split the child groups. This process is called *recursive partitioning.* The split is selected to construct a tree that can be used to predict the value of the target variable.

Some of the key features of decision trees include:

1. Easy to build, understand and interpret.

2. Handles both continuous and categorical variables.

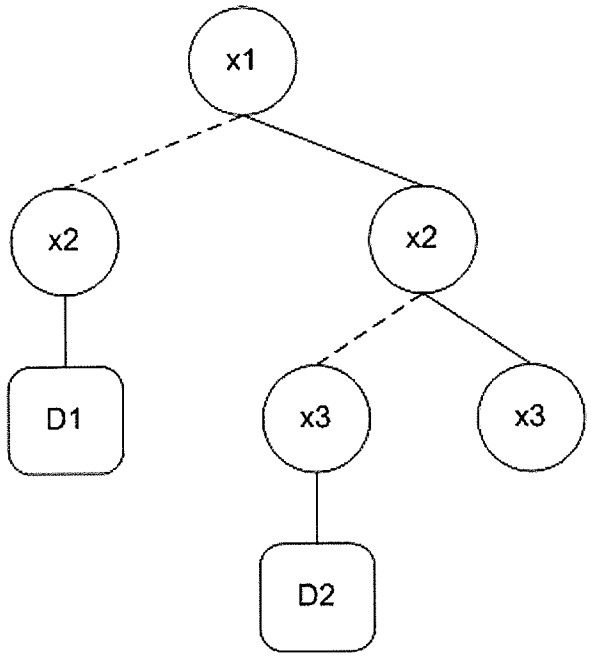

Figure 7.1: An Example of a Decision Tree

3. Performs classification as well as regression.

4. Automatically handles interactions between predictor variables.

5. Identifies importance of each predictor variables (in this case, the algorithms).

One of the most accurate modeling technique based on decision trees is called **TreeBoosting**, which is a method for improving the accuracy of a predictive function by applying the function repeatedly in a series and combining the output of each function with weighting so that the total error of the prediction is minimized. In many cases, the predictive accuracy of such a series greatly exceeds the accuracy of the base function used alone. In a TreeBoost model, the first tree is fitted to the simulated data. The residuals (error values) from the first tree are then fed into the second tree which attempts to reduce the error. This process is

repeated through a chain of successive trees. The final predicted value is formed by adding the weighted contribution of each tree. Usually, the individual trees are fairly small, but the full TreeBoost additive series may consist of hundreds of these small trees.

Using a commercially available software DTREG [77], a TreeBoost was generated based on 600 years of simulated data with each day comprising of all nine outbreak decisions and a flag indicating actual decision. All nine algorithms were used as predictor variables and the simulated outbreak flag was used as a categorical (**0** or **1**) target variable. Simulation was setup to build 20 trees with a maximum of 10 levels within each tree. The following overall importance of variable list was produced:

Algorithm	Importance (%)
WMA	100.00
C2	22.70
EWMA	16.64
MA5	4.86
C3	4.65
CUSUM	2.72
C1	2.47
MA7	1.14
MA3	0.00

As the table shows, only WMA, C2 and EWMA contributed significantly to the process. This is an expected result as 89% of the days in the simulated data set have a decision of 0. WMA, C2 and EWMA being algorithms with high

specificity, one would expect them to play a vital role in such biased dataset with large number of *no outbreak* days. On the other hand, when the input data set was modified to have 500 days with 59% of the days as *no outbreak* days (thus, producing a close to balanced data set), then the result was drastically different with following list as the variables of interest:

Algorithm	Importance (%)
C3	100.00
CUSUM	83.65
C2	83.59
EWMA	46.211
WMA	15.86
MA7	9.54
MA5	5.62
C1	3.53
MA3	0.00

In this case, C3, CUSUM, C2, EWMA and WMA play a vital role. This result is once again expected as collectively this set contributed to high sensitivity. When the two results are put together, the same set of algorithms appear on both side, which is very encouraging as the identified algorithm closely matches the minimal set identified by CAIF.

7.3 Significance of Contribution Graph

A novel approach that is very similar to decision trees was developed and implemented referred to as *significance of contribution (SOC) graph.* It analyzes each predictor variable (an algorithm) individually at each level and identifies one with highest contribution before proceeding to next level, where it considers all other variables and repeat the recursive process until no benefit is realized by adding more levels. The major difference with decision tree approach is that in decision trees the next level is a binary split using next predictor variable if classification is not complete.

An SOC graph is merely an OR network that exploits collective strength of multiple algorithms and identifies a sequence of algorithms that maximize the coverage of an outbreak decision, that is, maximum probability of correct decision. Fig 7.2 illustrates the SOC graph created using the nine candidate algorithms represented as nodes and the maximum significance nodes in bold and located in the center of the graph. Each link between two nodes provides a *coverage strength* value that corresponds to probability of picking up correct decision using the nodes in the chain with OR logic.

Analysing this further, the graph illustrates that a combination of four algorithms (**C3, CUSUM, C1, WMA**) provide maximum contribution for detecting days with an outbreak. Adding any more nodes does not increase the overall coverage strength of the system.

The graph is built one level at a time where a node with maximum coverage strength becomes the root node for second level. This significantly reduces the complexity by eliminating exploration of insignificant nodes. For instance, if one

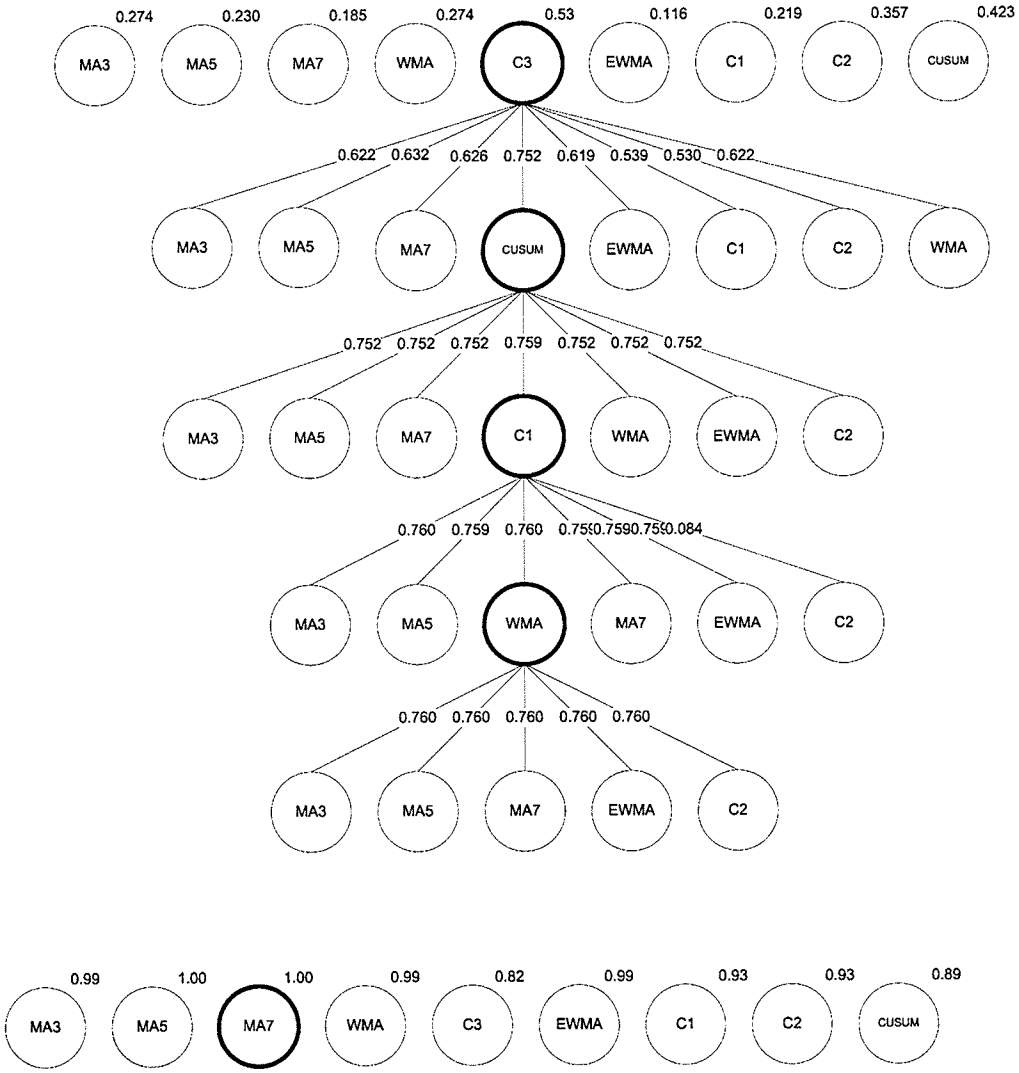

Figure 7.2: The Significance of Contribution Graph

was to build an SOC graph in its entirety, then there would be:

$$Possibilities = \sum_{1 \leq r \leq 9} \binom{9}{r}$$

which equates to **511** possibilities. A rather large number for a realistic presentable graph.

In summary, the SOC graph results in the same selection of minimal set as the Step 3 of the CAIF framework. Although this approach is based on purely simulated data without any fundamentals such as correlation and epidemiological parameters (sensitivity, specificity), it increases the confidence and comfort level of the result produced by the proposed CAIF approach.

7.4 Linear Discriminant Analysis

The following description and example images for linear discriminant analysis (LDA) have been extracted from [78].

Originally developed in 1936 by R.A. Fisher, *discriminant analysis* is a classic method of classification that has stood the test of time. It often produces models whose accuracy approaches (and occasionally exceeds) more complex modern methods.

In discriminant analysis, the dependent variable (Y) is the group and the independent variables (X) are the object features that might describe the group. The dependent variable is always a categorical variable while the independent

variables can be any measurement scale.

The LDA technique can be used if the assumption is that the groups being analyzed are linearly separable, which suggests that the groups can be separated by a linear combination of features that describe the objects. If there are only two features, the separators between objects group become lines. If there are three or more features, then the separators become hyper-planes.

Using the DTREG Simulator, linear discriminant analysis was performed on the balanced data set of 500 days which generated the following importance of variable table:

Algorithm	Importance (%)
C2	100
C3	35
EWMA	12
CUSUM	11
MA7	6
MA3	2
WMA	2
C1	1
MA3	0

Once again, the algorithms deemed to play a vital role include C2, C3, EWMA and CUSUM. Interesting observation is that WMA is not considered as one of the important algorithms. In any case, the output minimal set supports that proposed by CAIF.

7.5 Support Vector Machines

The following description and images for support vector machines have been extracted from [77, 79, 80].

One final technique that was considered is called *support vector machine* approach. A Support Vector Machine (SVM) performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories.

Within SVM model, a predictor variable is called an *attribute*, and a transformed attribute that is used to define the hyperplane is called a *feature*. The task of choosing the most suitable representation is known as feature selection. A set of features that describes one case (that is, a row of predictor values) is called a vector. So the goal of SVM modeling is to find the optimal hyperplane that separates clusters of vectors in such a way that points with one category of the target variable are on one side of the plane and points with the other category are on the other size of the plane. The vectors near the hyperplane are called the *support vectors.*

The simplest way to divide two groups is with a straight line, flat plane or an N-dimensional hyperplane. But what if the points are separated by a nonlinear region? In that case we need a nonlinear dividing line. Rather than fitting nonlinear curves to the data, SVM handles this by using a kernel function to map the data into a different space where a hyperplane can be used to do the separation (see Fig 7.3).

There are many different types of kernels that can be used, however, the

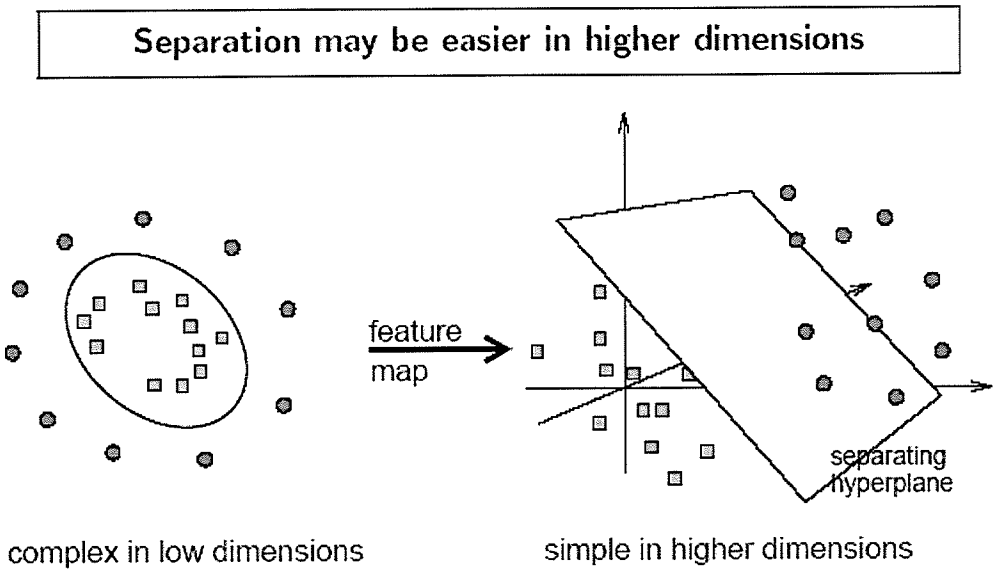

Figure 7.3: SVM Kernel [77]

following is a list of most commonly used [80]:

$$\phi = \left\{ \begin{array}{ll} x_i * x_j & \rightarrow Linear \\ (\gamma x_i x_j + coefficient)^{degree} & \rightarrow Polynomial \\ e^{-\gamma|x_i - x_j|^2} & \rightarrow Radial\ Basis\ Function \\ tanh(\gamma x_i x_j) + coefficient & \rightarrow Sigmoid \end{array} \right\}$$

Once again, using DTREG simulator, the following list was generated as the importance of variable table using a balanced set of 500 days for each of the four kernels:

Algorithm	RBF (%)	Linear (%)	2^{nd} order Polynomial (%)	Sigmoid
C2	100.00	100.00	100.00	100.00
EWMA	36.23	40.00	10.00	7.76
WMA	34.78	4.61	5.00	2.59
MA3	34.78	4.61	5.00	2.59
MA5	28.99	0.00	3.00	1.72
MA7	26.09	27.69	6.00	3.45
C1	2.89	0.00	1.00	0.00
C3	2.89	0.00	19.00	60.35
CUSUM	0.00	0.00	12.00	7.76

Its interesting to see that different kernels provide different results. However, most appropriate result that corresponds to other approaches is generated by sigmoid kernel with minimal set of C2, C3, CUSUM and EWMA.

7.6 Summary

In summary, all four approaches provide similar minimal set as the proposed CAIF approach. The following table summarizes the findings and provides comments for each of the approaches.

Approach	Minimal Set	Comments
CAIF	WMA, CUSUM, C1, C3	Based on epidemiological parameters (sensitivity, specificity and time-to-detect) and statistical agreement (kappa coefficient or correlation).
TreeBoost	WMA, CUSUM, C3, C2, EWMA	Based on decision trees and TreeBoosting technique.
SOC Graph	WMA, CUSUM, C1, C3	Based on novel recursive tree building approach based on significance of contribution by each algorithm.
Linear DA	CUSUM, C2, C3, EWMA	Based on discriminant analysis.
SVM	CUSUM, C2, C3, EWMA	Based on support vector machines and associated sigmoid kernel.

As can be seen, the proposed CAIF approach produces results that are confirmed by various different approaches. The analysis of all the discussed approaches are based on actual number of outbreak days in the input dataset, which is in complete contrast with proposed CAIF approach that uses sensitivity and specificity measures which depend on identifying an outbreak at least once during an outbreak period.

The advantage of CAIF over the alternative approaches discussed in this chapter is its simplicity of implementation and logical process of identifying the minimal set. The stepwise approach is both clear and easy of follow when compared to complicated mathematical models such as support vector machines with their kernels. This makes CAIF attractive from both implementation and further research perspective.

Chapter 8

Conclusion

As emphasized throughout this document, recent advances in technology have made it possible to gather, integrate, and analyze large amounts of data in real time or near-real time. These new technologies have touched off a renaissance in public health surveillance. For the most part, the traditional purposes of health surveillance have been to monitor long-term trends in disease ecology and to guide policy decisions. With the introduction of real-time capabilities, surveillance now holds the promise of facilitating early event detection and to assist in day-to-day disease management.

For prompt response and mitigation of serious outbreaks, disease events must be detected early and monitored in as near real time as possible. Early detection provides the opportunity not only to implement strategies to reduce exposure and limit disease, but also to mobilize front line response personnel and resources to meet primary care needs. A surveillance system that allows for the early detection of a disease episode (whether intentional or natural) would enable an

earlier, more rapid response. The detection problem is currently a challenging one for a variety of reasons, including continuous changes in the environment, host, and biologic agents, which change how a disease may present and evolve. The challenge is further complicated by the lack of integrated systems to collect, store, and analyze relevant surveillance data.

Once detected, disease events must be monitored and assessed accurately and in real time. Ongoing information on the prevalence, incidence, characterization, severity, and location of cases will provide health professionals with the information necessary to mobilize and allocate resources, monitor progression, and plan next steps. Mass numbers of individuals presenting to one emergency room will require a different type of response than modest numbers presenting to several emergency rooms in the same city. Individuals phoning into a tele-health service with similar symptoms from throughout an entire region would likely indicate the need for greater resource mobilization than the case where calls originate only from one neighborhood.

A detailed discussion on various epidemics and threats of influenza has been provided in the first chapter. It covers the historical perspective of influenza since the spanish flu. The chapter goes on to discuss the Canadian public health infostructure and presents some key findings outlined in the Naylor report. A justification of real time surveillance to potentially combat natural (accidental) and bioterrorist (intentional) biological crisis has been provided. In summary, the chapter states that an inter-disciplinary *health* surveillance approach needs to be embraced, where health officials, assisted by automated acquisition of data and generation of statistical alarms, monitor disease indicators continually (real-time) or at minimum daily (aggregate) to detect outbreaks of diseases earlier and

more completely than would otherwise be possible with traditional public health methods. Moreover, due to hierarchical and disparate nature of the health system, various types of health-care facilities will have to interact and participate and thus generate massive amounts of data and result sets that need to be analyzed for anomalies and presented to the public health individuals for investigation and decision making.

The second chapter provides a summary of some of the major existing syndromic surveillance systems around the globe in terms of their objectives, data collection and analysis methods. Real-time public health data surveillance systems are becoming key tools necessary to detect epidemics and initiate timely responses to outbreaks. An effective surveillance system would be able to detect the onset of any bio-terrorism related or naturally occurring disease outbreak at an early stage. Such early warning can aid in effective mobilization of resources during an outbreak. There are many such systems in place today in various parts of the world. An important point to note is that most of these systems collect data from remote locations to a centralized repository and then perform data analysis with various types of front-ends for data presentation. As discussed, this centralized data collection approach results in large amounts of data. With declining cost of memory storage devices and ease of getting access to aggregate data, such systems can play a vital role in identifying multi-disciplinary health threats. However, such systems must comprise of efficient methods and processes to deal with large amounts of data and anomaly interpretations.

The third chapter provides a detailed scan of analytical anomaly detection algorithms, which includes some of the most commonly used analytical algorithms

by the existing systems. With recent development in technology, real-time statistical disease monitoring has become an important issue. Biological terrorism and warfare has motivated development of computer systems for automatically detecting abrupt and epidemiologically significant changes in public health surveillance data. Such systems use automated anomaly detection algorithms to detect and alert anomalous findings. Two main categories of algorithms were discussed: *moving average* and *cusum* based.

This fourth chapter proposes a conceptual architecture, referred to as the *architecture for real-time information standardization and transformation* (ARTIST), for a real time surveillance system and identifies fundamental and support components required to deal with various functions related to data collection, analysis and presentation. The discussion is intended to provide an understanding of various functional components of a typical real time surveillance system including a critical surveillance gap relating to interpreting anomaly detection outcomes produced by individual aberration detection algorithms. .

The fifth chapter provides the details of the proposed confidence-based anomaly interpretation framework to quantitatively measure the anomaly outcomes generated by various algorithms by exploiting relationships between them. A detailed discussion has been presented including various algorithms and processes that facilitate confidence based interpretation to identify the start of an outbreak early on. This framework may be used to further the research in the following areas: (1) implement a larger set of algorithms within the framework and study their effects on the overall decision; (2) investigate optimal point allocation systems that potentially are computationally efficient when compared to proposed

random assignment scheme; and (3) explore learning systems such as neural networks within the framework to devise a system that potentially adapts to different minimal algorithm sets based on seasonality and variation in the data.

In the sixth chapter, the process of obtaining values of all CAIF parameters as well as results of application of CAIF approach using the computed parameters to both simulated and real outbreak scenarios were discussed at length. A simulation environment was setup that comprised of custom simulator for some aspects of the proposed approach as well as commercial and open source packages to compute various statistical and epidemiological parameters used in the proposed approach. The data for simulation were obtained from CDC. Nine candidate algorithms were selected based on literature review of most commonly used aberration detection algorithms, details of which were presented including the values of various parameters used by each of the algorithms. The epidemiological parameters (sensitivity, specificity and time to detect) were computed using the simulation environment. A discussion on the obtained values for these parameters in relation to each candidate algorithm was provided. Two approaches of measuring the agreement between various candidate algorithms were considered; *correlation* and *kappa coefficients*. Both approaches were simulated and the resulting coefficient matrices were discussed. Based on the agreement matrix and epidemiological parameters, the minimal set was computed. A five-step process was explained and the final set of four algorithms **(WMA, CUSUM, C1, C2)** was identified. Using this minimal set along with λ, ω and ϕ, the rule sets corresponding to three areas of interest (AOI) were identified. It was concluded that the proposed approach did not provide any gains in case of AOI(1) as **WMA** provided better result as a standalone algorithm. However, the approach provided significant gain when considering AOI(2) and AOI(3). Finally, the proposed CAIF approach was

applied to a real scenario generating very promising results of identifying start of an outbreak with high confidence measure thus, providing significant heads-up for a potential start of an outbreak.

The seventh chapter provides description of some of the commonly used techniques for data mining, pattern recognition, classification and prediction and includes results of their application to the problem of minimal set identification, that is, Step 3 of the proposed CAIF approach. Although the identified approaches are not typically applied to solve such a problem, however, careful definition of inputs and outputs along with proper interpretation of associated results provides excellent alternatives for minimal set identification process that can be compared against the proposed CAIF approach. The results indicated that all four approaches provided similar minimal sets as the proposed CAIF approach. The advantage of CAIF over other discussed approaches is its simplicity of implementation and logical process of identifying the minimal set. The stepwise approach is both clear and easy of follow when compared to complicated mathematical models such as support vector machines with their kernels. This makes CAIF attractive from both implementation and further research perspective.

In conclusion, the proposed framework provides a multitude of benefits:

1. Removal of algorithm redundancy by identifying only a smaller subset of algorithms that are necessary and sufficient for a sound system decision.

2. A mechanism to derive confidence value based on dynamic point assignment system.

3. A superior overall system decision within desired AOI when compared to any single algorithm. However, there is obviously the potential that a more

sophisticated technique will be more powerful, in which case it would be additive to the framework and likely more easily accepted by the community.

4. A framework for future research to investigate optimal point allocation systems as well as analysis of new algorithms and their effects on the overall decision.

Finally, the following is a list of potential areas for future research based on the proposed framework:

1. *Further Generalization*: Implementation of other versions of exponential smoothing schemes including seasonality corrected approach and its application to the overall framework.

2. *Time Effect*: Taking into account time of day, day of week, week of month and month of year within the framework and use it to deduce further redundancy between various algorithms.

3. *Identification of Optimal Rule*: Use of more sophisticated clustering techniques as well as optimal point identification systems to come up with best rule to use within a given area of interest.

4. *Data Labeling*: A feedback mechanism for public health specialists to close the loop for labeling outbreaks and no-outbreak decisions. This will extend the framework to allow for other techniques for evaluation purposes.

5. *Invariant Minimal Set*: There is no question that some algorithms are better than others when looking at different types of outbreaks. Applying a variety of outbreak types to the data (beyond log normal, daily spikes,

etc) will help in figuring out if the minimal set produced by the framework is invariant.

Appendices

The following two appendices arose out of informal discussions with the members of the examining committee.

The first attempts to outline some of the challenges facing implementation and deployment of health surveillance systems from within a Canadian context.

The second is an attempt to add credibility to the work undertaken by extending the framework to include more complex or aggressive algorithms than typical in health surveillance systems.

Appendix A

Current Issues

This appendix is dedicated to identifying some of the major issues that biosurveillance community faces with increasing need for sophisticated electronic systems and evolutionary data access abilities.

An assessment of the Canadian public health surveillance environment indicates that improvements are needed along the entire surveillance life cycle from data, to information, to intelligence, to communication (sharing of intelligence). Intelligence sharing is important because public health surveillance not only includes data exchange and analysis, but also the dissemination and sharing of information. Communication among and between stakeholders is often overlooked as a piece of the surveillance puzzle. To the contrary, communications is probably the most critical component of an effective public health surveillance system. This understanding of health surveillance is in line with Naylor's definition in his 2003 report in which he reflects upon the 2003 SARS episode in Canada [15]:

> Health Surveillance is] the tracking and forecasting of any health

event or health determinant through the continuous collection of high-quality data, the integration, analysis and interpretation of those data into surveillance products (for example reports, advisories, alerts, and warnings), and the dissemination of those surveillance products to those who need to know.

Over the past several years, a number of widely publicized health events have sparked much reflection, debate, and subsequent action relating to the state of public health surveillance in Canada. Two large waterborne outbreaks (an E. coli outbreak in Walkerton, Ontario, in 2000 and a cryptosporidiosis outbreak in North Battleford, Saskatchewan, in 2001 [81]), the 2003 SARS episode in Toronto, Ontario, and fears of bioterrorism since September 2001 have all prompted local, Provincial/Territorial, and Federal health authorities to critically assess their surveillance capacities, particularly with respect to infectious diseases.

1. Web-based and wireless technologies, data extraction tools, and advanced modeling and forecasting methods all have surveillance enabling roles to play. However, along with potentially facilitating public health surveillance in Canada, information technology also has the potential to further complicate an already complex public health arena. Over the past several years, there has been an explosion in technologies that are potentially applicable to public health surveillance. The challenge now is not whether processes are technically possible, but which technologies to harness and for what purposes, and how to implement and integrate them with existing systems and business processes. Given the seemingly infinite supply of new technologies, it is critical that in the development and implementation of new

surveillance systems, the ultimate purpose for harnessing new technologies be clearly defined and understood.

2. The growing divide in technical knowledge between the ultimate users of surveillance systems (i.e., public health stakeholders) and information technology professionals could result in surveillance systems and tools that do not fully meet-or do not efficiently meet- their intended goals. A fully closed-loop system that provides analytical tools to the public health stakeholders and expects interpretations from stakeholders is much needed. This closed-loop approach will facilitate *labeling* of outbreaks which is paramount to development of future algorithms.

3. With the exception of some very recent advances in syndromic surveillance and other close to real-time surveillance approaches, public health surveillance has not fundamentally evolved much this past century. Data on specific diseases are gathered, aggregated at a central location, analyzed, and summarized. For the most part, public health surveillance is still very much disease- or program-specific. How data are gathered, aggregated, analyzed, and summarized may be quite different for influenza than for enteric illnesses, and different again for sexually transmitted diseases. Many disease-specific surveillance systems still rely on the manual collection of data from the providers, mail and fax (and now email) to deliver data to a centralized location, manual execution of basic statistical procedures to summarize the data, and mail and fax (and email) to distribute the results. This poses challenges with data collection.

4. It is well recognized that a comprehensive early warning system must identify as many individuals as possible early in the disease process when they

have nonspecific symptoms, such as cough or diarrhea, and then use statistical algorithms to find any interesting patterns among the sick individuals that suggest that an unusual event is occurring. Such a system needs to receive data directly and within an acceptable time frame. Candidate sources of early warning data are many and include emergency departments, laboratories, pharmacies, and tele-triage systems. Such diverse nature of data providers combined with their own unique systems creates a need for a dynamic and flexible architecture for data collection and collation.

5. Despite their promise for facilitating early event detection and real-time disease monitoring, real-time surveillance systems, particularly syndromic surveillance systems, have not been a main priority in the health care community. The primary reason is that syndromic surveillance is still an unproven concept, and the expectations of what can be accomplished through syndromic surveillance vary widely. Many public health stakeholders are afraid of data and alert overload. Furthermore, for many, the technologies and methods employed (particularly complex statistical algorithms) are foreign. An evaluation of the effectiveness of syndromic surveillance is urgently needed so that policy makers can determine if, when, and how these methods should be applied. To be effective, a screening tool such as syndromic surveillance must detect outbreaks early and lead to more effective intervention than if the outbreaks were detected through traditional means. The proportion of relevant outbreaks that syndromic surveillance detects, the extent to which relevant outbreaks are detected early, and the rate of false alarms remain largely unknown, as are the benefits in terms of changes in the course of illness and community health.

6. The framework proposed in this document aids in identification of minimal set of algorithms from a pool of algorithms that would be sufficient for identification of start of a potential event. However, careful note must be taken that implementing such a framework alone is not sufficient moving forward. New algorithms needs to be developed. Existing algorithms need to be improved. The framework may then be used to compare the newly developed algorithms against a growing pool of algorithms.

7. The framework proposes a distributed data collection environment using the proposed ARTIST architecture. The fact that data comes from geographically distributed locations, algorithms that consider spatial scans need more dedicated research focus, although some algorithms do exist already, nonetheless, the area needs more research as various parameters may provide wealth of knowledge into the outbreak identification issue, such as population, demographics and vicinity.

8. Due to the nature of health care system and seamless human mobility, there is a need for systems that put emphasis on automated patient linkages between various systems within a jurisdiction and between jurisdictions. Issues such as changing residency, treatment across jurisdictions and cross-border treatments need to be investigated.

A thorough description of statistical challenges in modern biosurveillance has also been provided in [82].

Appendix B

CAIF Generalization

The proposed confidence based aberration interpretation framework (CAIF) is well suited for evaluation of new algorithms in relation to a set of candidate algorithms pool. Indicators such as uniqueness of an algorithm in terms of collective contribution as a set may be deduced by applying the algorithm through the framework steps and monitoring the identified minimal set. This appendix is focused at introducing a new algorithm in to the candidate set, an algorithm that has never been considered duing the development of the proposed framework.

The application of new algorithm to the framework provides support for:

1. *Scalability*: ability of the framework to deal with introduction of new algorithms in terms of number of candidate algorithms.

2. *Performance*: does the framework perform as expected when introducing an efficient algorithm.

3. *Flexibility*: ability of the framework to analyze a different type of algorithm

than the ones implemented during the simulation study and development of the framework.

Keeping the above in mind, an effective and parsimonious approach referred to as exponential smoothing [83] was investigated. A version of this technique that takes trending into account was implemented. Exponential smoothing techniques are simple and intuitive yet very robust and effective approaches to forecasting. Although widely used in business arena, the techniques may be used effectively for aberration detection based on their forecasting abilities.

The following set of equations were used to implement the daily trend corrected exponential smoothing (ES) algorithm [84]:

$$T_i = (1 - \beta)T_{i-1} + \beta(F_i - F_{i-1})$$

where T_i is the smoothed trend for current day i, β is the trend smoothing constant (set to **0.7**), T_{i-1} is the smoothed trend for previous day $i - 1$, F_i is the simple exponential forecast for current day i and F_{i-1} is the simple exponential forecast for previous day $(i - 1)$ and,

$$F_i = F_{i-1} + \alpha(C_{i-1} - F_{i-1})$$

where C_{i-1} is the actual daily count for previous day $(i - 1)$ and α is the exponential smoothing constant (set to **0.5**).

This algorithm along with the other nine candidate algorithms **(3MA, 5MA,**

7MA, WMA, EWMA, CUSUM, C1, C2, C3) were applied to the framework using a data set of simulated outbreaks over 200000 days (the same set used to simulate the nine candidate algorithms in the chapter that discusses simulation results). As expected the **ES** algorithm produced a very high specificity (95.8%) and much higher sensitivity values (71.7%) when compared to the other five moving average variants. This signals that the eventual minimal set should include **ES** as one of the algorithms. The issue, however, is whether it would replace any of the existing four minimal set algorithms **(WMA, CUSUM, C1, C3)** or complement them. That is, how does **ES** relate to the others in the pool of candidate algorithms?

Running through step 2 of the framework (the agreement analyzer), it was interesting to note that **ES** algorithm did not agree (at least strongly) with any of the other candidate algorithms. That is, the kappa value against all other algorithms was less that T_A^{kappa} (which was set to 0.5). This means that **ES** algorithm produces slightly different decisions than any of the existing algorithms. Specifically, the values were: **(0.486, 0.448, 0.424, 0.393, 0.323, 0.129, -0.041, -0.015, -0.036)** corresponding to the other nine algorithms identified above. As expected the algorithm has stronger relationship with variants of moving average algorithms than the cusum based algorithms. However, the relationship is not strong because the issue with the moving average techniques is that they often fail to respond quickly to trends in the data. The application of trend adjustment along with exponential smoothing formulates a forecast that includes the continuously learned trend adjustment. The trend adjustment also uses a smoothing constant to enable dynamic weight assignment to more recent changes in the trend based on its value.

Finally, going through the third step of the framework, the identified minimal set was **(WMA, ES, CUSUM, C1, C3)**. That is, **ES** was considered to be an independent algorithm and did not replace any of the previously identified minimal set algorithms. Specifically, the following list guides through each task of the third step of the framework.

- *Task 1*: The candidate algorithm set was initialized to **[3MA, 5MA, 7MA, WMA, EWMA, CUSUM, C1, C2, C3, ES]**.
- *Task 2*: Based on ρ_{kappa}, the CAIF Simulator produced a closest relative matrix of **[WMA, 7MA, WMA, -1, -1, -1, -1, C3, -1, -1]**, where -1 represents an independent algorithm.
- *Task 3*: Based on the closest relative matrix, a working set of **[0, 0, 0, 1, 1, 1, 1, 0, 1, 1]** was produced, where '1' implies that the corresponding algorithms were considered in the grouping task (Task 4).
- *Task 4*: Based on the working set obtained in the previous task, the CAIF Simulator yielded the following three groups in order: G1 = **[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]**, G2 = **[0, 0, 0, 0, 0, 1, 0, 0, 1, 0]** and G3 = **[0, 0, 0, 1, 0, 0, 1, 0, 0, 1]**. That is, recalling the definitions of groups G1, G2 and G3 from previous chapter, none of the algorithms in the candidate set had TTD value of close to zero ; CUSUM and C3 as algorithms with TTD value of less that one day; and WMA, C1 and ES as algorithms with TTD value greater than one day.
- *Task 5*: Based the three groups produced in the previous task, the CAIF Simulator produced **[WMA, CUSUM, C1, C3, ES]** as the minimal set. That is, the first group was empty set with no algorithms; the second group

was set to **[0, 0, 0, 0, 0, 1, 0, 0, 1, 0]** because CUSUM and C3 complement each other in terms of sensitivity and specificity values; and finally the third group was set to **[0, 0, 0, 1, 0, 0, 1, 0, 0, 1]** because WMA, C1 and ES complement each other as well.

This brief study demonstrates that the proposed framework may be scaled to include new algorithms and identify appropriate minimal set based on the strengths of the new algorithm when compared to algorithms in the candidate pool. Furthermore, it also emphasizes the need to seriously investigate all forms of exponential smoothing technique including seasonality [85, 86] and level [83] and their respective application to aberration detection.

Bibliography

[1] Madjid M, Lillibridge S, Parsa M *et al.* Influenza as a Bioweapon. *Journal of the Royal Society of Medicine.* Vol. 96, No. 7, pp. 345-346. 2003.

[2] Stanford University. Pandemic of 1918. June 1997. Available from: http://www.stanford.edu/group/virus/uda. Accessed 03 April 2007.

[3] Pandemic of 1918. History In Depth. July 2003. Available from: http://www.rbls.lib.il.us/dpl/ref/hist/hid/histhidflu.htm.

[4] The great influenza pandemic: Could it happen again? Available from: http://www.influenza.com/Index.cfm?FA=Science_History_4. Accessed 03 April 2007.

[5] Department of Health and Human Services (DHHS). Pandemics and Pandemic Scares in the 20th Century. National Vaccine Program Office. February 2004. Available from: http://www.hhs.gov/nvpo/pandemics/flu3.htm. Accessed 03 April 2007.

[6] Snacken R, Kendal A, Haaheim L and Wood J. The Next Influenza Pandemic: Lessons from Hong Kong, 1997. *Emerging Infectious Diseases.* Vol. 5, No. 2, pp. 195-203. March-April 1999.

[7] Pandemics and Pandemic Threats since 1900. Available at: http://www.pandemicflu.gov/general/historicaloverview.html. Accessed 17 May 2007.

[8] Li R, Leung K, Sun F and Samaranayake L. Severe Acute Respiratory Syndrome (SARS) and the GDP. Part I: Epidemiology, virology, pathology and general health issues. *British Dental Journal.* Vol. 197, No. 2. July 2004.

[9] Heymann D and Rodier G. Global Surveillance, National Surveillance, and SARS. *Emerging Infectious Diseases.* Vol. 10, No, 2, pp.173-175. 2004.

[10] World Health Organization. Severe Acute Respiratory Syndrome (SARS): Status of the Outbreak and Lessons for the Immediate Future. SARS technical briefing. WHA 56. 20 May 2003.

[11] Gensheimer K, Meltzer M, Postema A and Strikas R. Influenza Pandemic Prepardness. *Emerging Infectious Diseases.* Vol. 9, No. 12, pp. 1645-1648. April 2004.

[12] Center for Infectious Disease Research and Policy (CIDRAP). Influenza virus seen as possible bioterrorism weapon. Available from: http://www.cidrap.umn.edu/cidrap/content/bt/bioprep/news/july0803flu .html. Accessed 04 April 2007.

[13] Centre for Disease Control (CDC). Smallpox Home. Available from: http://www.bt.cdc.gov/agent/smallpox/. Accessed 03 April 2007.

[14] Centre for Disease Control (CDC). Bioterrorism Agents and Diseases. November 2004. Available from: http://www.bt.cdc.gov/agent/agentlist-category.asp. Accessed 03 April 2007.

[15] Health Canada. Learning from SARS: Renewal of Public Health in Canada. A report of the national advisory committee on SARS and public health. October 2003.

[16] National Enteric Surveillance Program (NESP). Available at: http://www.nml-lnm.gc.ca/english/NESP.htm. Accessed on 16 May 2007.

[17] Gesteland P, Gardner R, Tsui F, Espino J, Rolfs R, James B, Chapman W, Moore A and Wagner M. Automated Syndromic Surveillance for the 2002 Winter Olympics. *Journal of American Medical Informatics Association.* Vol. 10, No. 6, pp. 547-554, November 2003.

[18] Wagner M, Robinson J, Tsui F-C, Espino J, and Hogan W. Design of a National Retail Data Monitor for Public Health Surveillance. *Journal of American Medical Informatics Association.* Vol. 10, No. 5, pp. 409-418. September 2003.

[19] McDonald L, Aramini J, Mukhi S, Edge V, and Kabani A. The Winnipeg Regional Health Authority Early Event Detection Project: A System Overview. *Syndromic Surveillance Conference.* November 2005.

[20] Espino J, Hogan W and Wagner M. Telephone Triage: A Timely Data Source for Surveillance of Influenza-like Diseases. *Proceedings of American Medical Informatics Association (AMIA) Annual Symposium.* pp. 215-219. 2003

[21] Buehler J, Berkelman R, Hartley D, Peters C. Syndromic surveillance and bioterrorism-related epidemics. *Emerging Infectious Diseases.* Vol. 9, No 10, pp. 1197-204. October 2003.

[22] Goldenberg A, Shmueli G, Caruana R and Fienberg S. Early statistical detection of anthrax outbreaks by tracking over-the-counter medication sales. *Proceedings of National Academy of Sciences of the U.S.A.* Vol. 99, No. 8. April 2002.

[23] Mostashari F and Hartman J. Syndromic surveillance: a local perspective. *Journal of Urban Health: Bulletin of the New York Academy of Medicine.* Vol. 80 No. 2, Supplement 1. pp. i1-i7, 2003.

[24] Lombardo J, Burkom H, Elbert E, Magruder S, Lewis S, Loschen W, Sari J, Sniegoski C, Wojcik R and Pavlin J. A Systems Overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). *Journal of Urban Health: Bulletin of the New York Academy of Medicine.* Vol. 80, No. 2. 2003.

[25] Stoto M, Schonlau M and Mariano L. Syndromic Surveillance: Is It Worth the Effort? *Chance.* Vol. 17, No. 1, pp. 19-24. 2004.

[26] Reingold A. If Syndromic Surveillance Is the Answer, What Is the Question? *Biosecurity and Bioterrorism: Biodefense Stratgey, Practise and Science.* Vol. 1, No. 2. 2003.

[27] Sosin D. Syndromic surveillance: the case for skillful investment. *Biosecurity and Bioterrorism.* Vol. 1, No. 4, pp. 247-53. 2003.

[28] Lewis M, Pavlin J, Mansfield J, O'Brien S, Boomsma L, Elbert Y and Kelle W. Disease outbreak detection system using syndromic data in the greater Washington DC area. *American Journal of Preventive Medicine.* Vol. 23, No. 3, pp. 180-186. October 2002.

[29] Hutwagner L, Thompson W, Seeman G and Treadwell T. 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. 2, Supplement 1, pp. i89-i96. 2003.

[30] Kuldroff M. and Nagarwalla N. Spatial disease clusters: detection and inference. *Statistics Medicine.* Vol. 14, pp. 799-810. 1995.

[31] Lombardo J, Burkom H and Pavlin J. ESSENSE II and the Framework for Evaluating Syndromic Surveillance Systems. *Morbidity and Mortality Weekly Report (MMWR).* Supplement 53, pp. 159-165. 2004.

[32] Wagner M. Testimony on building an early warning public health surveillance system. November 2001.

[33] Lober W, Trigg L, Karras B, Bliss D, Ciliberti J, Stewart L and Duchin J. Syndromic surveillance using automated collection of computerized discharge diagnoses. *Journal of Urban Health: Bulletin of the New York Academy of Medicine.* Vol. 80 No. 2, Supplement 1. pp. i97-i106. 2003.

[34] Buckeridge D, Graham J, O'Connor M, Choy M, Tu S and Musen M. Knowledge-based bioterrorism surveillance. *Proceedings of American Medical Informatics Association (AMIA) Annual Symposium.* pp. 76-80. 2002.

[35] Vennergrund D. The Application of Data Mining to Bioterrorism. SRA International. February 2004.

[36] Reichard G, Demitry P and Catalino J. COHORT: An Integrated Information Approach to Decision Support for Military Subpopulation Health Care. *Defense Technical Information Center (DTIC).* Report No. ADA461819. 2004.

[37] Widdowson M-A, Bosman A, Straten E, Tinga M, Chaves S, Eerden L, and Pelt W. Automated, Laboratory-based System Using the Internet for Disease Outbreak Detection - the Netherlands. *Emerging Infectious Diseases.* Vol. 9, No. 9. September 2003.

[38] McDonald L, McDonald K, Edge V, Mukhi S, and Aramini J. Detection Abilities of Several Commonly Used Algorithms as Determined by Simulation Analysis. *Syndromic Surveillance Conference.* 2006.

[39] McDonald L, Guthrie G, Edge V, Mukhi S, and Aramini J. Evaluation of a Systematic Emergency Department Chief Complaint System for Near Real-Time Public Health Surveillance. *Syndromic Surveillance Conference.* 2006.

[40] McDonald L, Aramini J, Edge V, Mukhi S, and Ferguson J. Evaluation of the Canadian Early Warning System (CEWS): An Early Event Detection Application. *1st Annual Public Health Agency of Canada Forum.* March 2006.

[41] Stroup D, Williamson G, Herndon J and Karon J. Detection of aberrations in the occurrence of notifiable diseases surveillance data. *Statistics Medicine.* Vol. 8, pp. 323-329. 1989.

[42] Farrington C, Beale A, Andrews N and Catchpole M. A statistical algorithm for the early detection of outbreaks of infectious disease. *Journal of the Royal Statistical Society. Series A (Statistics in Society).* Vol. 159, No. 3, pp. 547-563. 1996.

[43] Stern L and Lightfoot D. Automated outbreak detection: a quantitative retrospective analysis. *Epidemiology Infection.* Vol. 22, No. 1, pp. 103-10. 1999.

[44] Bradley C, Rolka H, Walker D and Loonsk J. BioSense: implementation of a national early event detection and situational awareness system. *Morbidity and Mortality Weekly Report (MMWR).* Supplement 54, pp. 11-19. 2005.

[45] Klienman K, Lazarus R and Platt R. A Generalized Linear Mized Models Approach for Detecting Incident Clusters of Diseases in Small Areas, with an Application to Biological Terrorism. *American Journal of Epidemiology.* Vol. 159, No. 3, pp. 217-224. 2004.

[46] Greenko J, Mostashari F, Fine A and Layton M. Clinical Evaluation of the Emergency Medical Services (EMS) Ambulance Dispatch-Based Syndromic Surveillance System,New York City. *Journal of Urban Health.* Vol. 80, No. 2, Supplement 1, pp i50-56. 2003.

[47] Hashimoto S, Murakami Y, Taniguchi K and Nagai M. Detection of epidemics in their early stage through infectious disease surveillance. *International Journal of Epidemiology.* pp 905-910. 2000.

[48] Hogan W, Tsui F-C, Ivanov O, Gesteland P, Grannis S, Overhage J, Robinson J and Wagner M. Detection of Pediatric Respiratory and Diarrheal Outbreaks from Sales of Over-the-counter Electrolyte Products. *Journal of the American Medical Informatics Association.* Vol. 10, No. 6. November-December 2003.

[49] Foldy S, Biedrzycki P, Barthell E, *et al.* The Public Health Dashboard: A Surveillance Model for Bioterrorism Preparedness. *Journal of Public Health Management and Practice.* Vol. 10, No. 3, pp. 234-240. May-June 2004.

[50] Buckeridge D, O'Connor M, Xu H and Musen M. A Knowledge-Based Framework for Deploying Surveillance Problem Solvers. *Proceedings of American*

Medical Informatics Association (AMIA) Annual Symposium. pp. 76-80. 2002

[51] Lober W, Karras B, Wagner M, *et al.* Roundatable on Bioterrorism Detection. *Journal of the American Medical Informatics Association.* Vol. 9, No. 2. Mar-Apr 2002.

[52] Reis B and Mandl K. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making.* Vol. 3, No. 2. January 2003.

[53] Reis B, Pagano M, and Mandl K. Using temporal context to improve biosurveillance. *Proceedings of National Academy of Sciences of the U.S.A.* Vol. 100, No. 4, pp. 1961-1965. February 2003.

[54] Williamson G and Weatherby H. A monitoring system for detecting aberrations in public health surveillance reports. *Statistics Medicine.* Vol. 18, No. 23, pp. 3283-3298. December, 1999.

[55] Rolfhamre P. Outbreak Detection of Communicable Diseases -Design, Analysis and Evaluation of Three Models for Statistically Detecting Outbreaks in Epidemiological Data of Communicable Diseases. Master's Thesis in Computer Science. Stockholm University. 2003.

[56] Centre for Disease Control (CDC). Updated Guidelines for Evaluating Public Health Surveillance Systems. Available from: http://www.cdc.gov/mmwr/preview/mmwrhtml/rr5013a1.htm. Accessed 03 April 2007.

[57] NIST/SEMATECH e-Handbook of Statistical Methods: Single Exponential Smoothing at the National Institute of Standards and Technology. Available from: http://www.itl.nist.gov/div898/handbook/. Accessed 04 April 2007.

[58] The Canadian Early Warning System. Available from https://www.cnphi-rcrsp.ca.

[59] Morton A, Whitby M, Mclaws M, *et al.* The application of statistical process control charts to the detection and monitoring of hospital-acquired infections. *Journal of Quality In Clinical Practice.* Vol. 21, No. 4, pp. 112117. 2001.

[60] Brown S, Benneyan J, Theobald D, Sands K, Hahn M, Potter-Bynoe G, Stelling J, O Brien T and Goldmann D. Binary cumulative sums and moving averages in nosocomial infection cluster detection. *Emerging Infectious Diseases.* Vol. 8, No. 12, pp. 1426-32. 2002.

[61] Hutwagner L, Maloney E, Bean N, Slutsker L and Martin S. Using laboratory-based surveillance data for prevention:an algorithm for detecting Salmonella outbreaks. *Emerging Infectious Diseases.* Vol. 3, No. 3, pp. 395-400. 1997.

[62] O Brien S. and Christie P. Do CuSums have a role in routine communicable disease surveillance? *Public Health.* Vol. 111, No. 4, pp. 255-8. July 1997.

[63] Wong W-K, Moore A, Cooper G and Wagner M. WSARE: What 's Strange About Recent Events? *Journal of Urban Health: Bulletin of the New York Academy of Medicine.* Vol.80, No.2, Supplement 1. 2003.

[64] Wong W-K, Moore A, Cooper G and Wagner M. Bayesian network anomaly pattern detection for disease outbreaks. *Proceedings of the Twentieth International Conference on Machine Learning.* pp. 808-813. 2003.

[65] Wong W-K, Moore A, Cooper G and Wagner M. Rule-based Anomaly Pattern Detection for Detecting Disease Outbreaks. *Proceedings of the 18th National Conference on Artificial Intelligence.* MIT Press. 2002.

[66] Moore A, Cooper G, Tsui R and Wagner M. Summary of Biosurveillance-relevant statistical and data mining technologies. Unpublished report. February 2002.

[67] Jackson M, Baer A, Painter I and Duchin J. A simulation study comparing aberration detection algorithms for syndromic surveillance. *BMC Medical Informatics and Decision Making.* Vol. 7. 2007.

[68] Wikipedia. K-Means Algorithm. Available from: http://en.wikipedia.org/wiki/K-means_algorithm. Accessed 01 April 2007.

[69] The R Package for Statistical Computing. Available from: http://www.r-project.org/. Accessed 01 April 2007.

[70] Centre for Disease Control (CDC). Data Sets. Available from: http://www.bt.cdc.gov/surveillance/ears/datasets.asp. Accessed 03 April 2007.

[71] Trochim W. Correlation. Available from: http://www.social researchmethods.net/kb/statcorr.htm. Accessed 03 April 2007.

[72] Jacob C. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement.* Vol. 20, pp. 3746. 1960.

[73] Landis J and Koch G. The measurement of observer agreement for categorical data. *Biometrics.* Vol. 33, No. 1, pp. 159-174. 1977.

[74] Waner S and Costenoble S. Linear Regression. Available from: http://people.hofstra.edu/faculty/stefan_waner/realworld/tutorialsf0/frames1_5.html. Accessed 03 April 2007.

[75] Heckerman D. A Tutorial on Learning with Bayesian Networks. *Microsoft Research Advanced Technology Division.* Technical Report MSR-TR-95-06. March 1995.

[76] Gehrke J and Loh W-Y. Advances in Decision Tree Construction. Cornell University. 2001.

[77] Software for Predictive Modeling and Forecasting. Available from: http://www.dtreg.com/. Accessed 01 April 2007.

[78] Balakrishnama S and Ganapathiraju A. Linear Discriminant Analysis - A Brief Tutorial. *Institute of Signal and Information Processing.* Mississipi State University.

[79] Shawe-Taylor J and Cristianini N. Support Vector Machines and other kernel-based learning methods. Cambridge University Press. 2000.

[80] Support Vector Machine (SVM). Available from: http:// www.statsoft.com/textbook/stsvm.html. Accessed 01 April 2007.

[81] Stirling R, Aramini J, Ellis A, Lim G, Meyers R, Fleury M. Waterborne cryptosporidiosis outbreak, North Battleford, Saskatchewan, Spring 2001. Ottawa, Ontario: Health Canada 2000.

[82] Shmeuli G and Burkom H. Statistical Challanges in Modern Biosurveillance.

[83] Billah B, King M, Snyder R and Koehler A. Exponential smoothing model selection for forecasting. *International Journal of Forecasting.* Vol. 22, pp. 239-247. 2006.

[84] Department of Computer Science - University of Saskatchewan. Exponential smoothing with trend adjustment. Available at http://bosna.usask.ca/resources/tutorials/csconcepts/.

[85] Winters, P. Forecasting sales by exponentially weighted moving averages. Management Science. Vol. 6, 324342. 1960.

[86] Gardner E and McKenzie E. Seasonal exponential smoothing with damped trends. Management Science. Vol. 35, No. 3, pp. 372-372. 1989.