Outlier Detection : A Survey

VARUN CHANDOLA
University of Minnesota
ARINDAM BANERJEE
University of Minnesota
and
VIPIN KUMAR
University of Minnesota

Outlier detection has been a very important concept in the realm of data analysis. Recently, several application domains have realized the direct mapping between outliers in data and real world anomalies, that are of great interest to an analyst. Outlier detection has been researched within various application domains and knowledge disciplines. This survey provides a comprehensive overview of existing outlier detection techniques by classifying them along different dimensions.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms: Algorithms

Additional Key Words and Phrases: Outlier Detection, Anomaly Detection

1. INTRODUCTION

Outlier detection refers to the problem of finding patterns in data that do not conform to expected normal behavior. These anomalous patterns are often referred to as outliers, anomalies, discordant observations, exceptions, faults, defects, aberrations, noise, errors, damage, surprise, novelty, peculiarities or contaminants in different application domains. Outlier detection has been a widely researched problem and finds immense use in a wide variety of application domains such as credit card, insurance, tax fraud detection, intrusion detection for cyber security, fault detection in safety critical systems, military surveillance for enemy activities and many other areas.

The importance of outlier detection is due to the fact that outliers in data translate to significant (and often critical) information in a wide variety of application domains. For example, an anomalous traffic pattern in a computer network could mean that a hacked computer is sending out sensitive data to an unauthorized destination. In public health data, outlier detection techniques are widely used to detect anomalous patterns in patient medical records which could be symptoms of a new disease. Similarly, outliers in credit card transaction data could indicate credit card theft or misuse. Outliers can also translate to critical entities such as in military surveillance, where the presence of an unusual region in a satellite image of enemy area could indicate enemy troop movement. Or anomalous readings from a space craft would signify a fault in some component of the craft.

Outlier detection has been found to be directly applicable in a large number of domains. This has resulted in a huge and highly diverse literature of outlier detec-

tion techniques. A lot of these techniques have been developed to solve focussed problems pertaining to a particular application domain, while others have been developed in a more generic fashion. This survey aims at providing a structured and comprehensive overview of the research done in the field of outlier detection. We have identified the key characteristics of any outlier detection technique, and used these as dimensions to classify existing techniques into different categories. This survey aims at providing a better understanding of the different directions in which research has been done and also helps in identifying the potential areas for future research.

1.1 What are outliers?

Outliers, as defined earlier, are patterns in data that do not conform to a well defined notion of normal behavior, or conform to a well defined notion of outlying behavior, though it is typically easier to define the normal behavior. This survey discusses techniques which find such outliers in data.

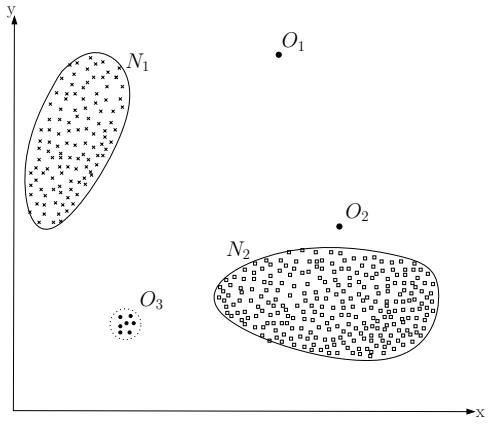


Fig. 1. A simple example of outliers in a 2-dimensional data set

Figure 1 illustrates outliers in a simple 2-dimensional data set. The data has two normal regions, N_1 and N_2. O_1 and O_2 are two outlying instances while O_3 is an outlying region. As mentioned earlier, the outlier instances are the ones that do not lie within the normal regions.

Outliers exist in almost every real data set. Some of the prominent causes for outliers are listed below

- *Malicious activity* – such as insurance or credit card or telecom fraud, a cyber intrusion, a terrorist activity
- *Instrumentation error* – such as defects in components of machines or wear and tear
- *Change in the environment* – such as a climate change, a new buying pattern among consumers, mutation in genes
- *Human error* – such as an automobile accident or a data reporting error

Outliers might be induced in the data for a variety of reasons, as discussed above, but all of the reasons have a common characteristic that they are *interesting* to the analyst. The "interestingness" or real life relevance of outliers is a key feature of outlier detection and distinguishes it from *noise removal* [Teng et al. 1990] or *noise accommodation* [Rousseeuw and Leroy 1987], which deal with unwanted *noise* in the data. Noise in data does not have a real significance by itself, but acts as a hindrance to data analysis. Noise removal is driven by the need to remove the unwanted objects before any data analysis is performed on the data. Noise accommodation refers to immunizing a statistical model estimation against outlying observations. Another related topic to outlier detection is *novelty detection* [Markou and Singh 2003a; 2003b; Saunders and Gero 2000] which aims at detecting unseen (*emergent, novel*) patterns in the data. The distinction between novel patterns and outliers is that the novel patterns are typically incorporated with the normal model after getting detected. It should be noted that the solutions for these related problems are often used for outlier detection and vice-versa, and hence are discussed in this review as well.

1.2 Challenges

A key challenge in outlier detection is that it involves exploring the unseen space. As mentioned earlier, at an abstract level, an outlier can be defined as a pattern that does not conform to expected normal behavior. A straightforward approach will be to define a region representing normal behavior and declare any observation in the data which does not belong to this normal region as an outlier. But several factors make this apparently simple approach very challenging.

- Defining a normal region which encompasses every possible normal behavior is very difficult.
- Often times normal behavior keeps evolving and an existing notion of normal behavior might not be sufficiently representative in the future.
- The boundary between normal and outlying behavior is often fuzzy. Thus an outlying observation which lies close to the boundary can be actually normal and vice versa.
- The exact notion of an outlier is different for different application domains. Every application domain imposes a set of requirements and constraints giving rise to a specific problem formulation for outlier detection.
- Availability of labeled data for training/validation is often a major issue while developing an outlier detection technique.
- In several cases in which outliers are the result of malicious actions, the malicious adversaries adapt themselves to make the outlying observations appear like normal, thereby making the task of defining normal behavior more difficult.
- Often the data contains noise which is similar to the actual outliers and hence is difficult to distinguish and remove.

In the presence of above listed challenges, a generalized formulation of the outlier detection problem based on the abstract definition of outliers is not easy to solve. In fact, most of the existing outlier detection techniques simplify the problem by

focussing on a specific formulation. The formulation is induced by various factors such as the nature of data, nature of outliers to be detected, representation of the normal, etc. In several cases, these factors are governed by the application domain in which the technique is to be applied. Thus, there are numerous different formulations of the outlier detection problem which have been explored in diverse disciplines such as *statistics*, *machine learning*, *data mining*, *information theory*, *spectral decomposition*.

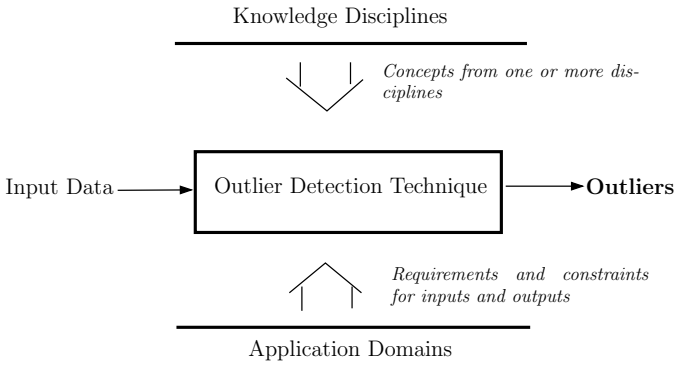


Fig. 2. A general design of an outlier detection technique

As illustrated in Figure 2, any outlier detection technique has following major ingredients —

1. Nature of data, nature of outliers, and other constraints and assumptions that collectively constitute the *problem formulation*.
2. *Application domain* in which the technique is applied. Some of the techniques are developed in a more generic fashion but are still feasible in one or more domains while others directly target a particular application domain.
3. The concept and ideas used from one or more *knowledge disciplines*.

1.3 Our Contributions

The contributions of this survey are listed below

1. We have identified the key dimensions associated with the problem of outlier detection.
2. We provide a multi-level taxonomy to categorize any outlier detection techniques along the various dimensions.
3. We present a comprehensive overview of the current outlier detection literature using the classification framework.
4. We distinguish between instance based outliers in data and more complex outliers that occur in sequential or spatial data sets. We present separate discussions on techniques that deal with such complex outliers.

The classification of outlier detection techniques based on the applied knowledge discipline provides an idea of the research done by different communities and also

highlights the unexplored research avenues for the outlier detection problem. One of the dimensions along which we have classified outlier detection techniques is the application domain in which they are used. Such classification allows anyone looking for a solution in a particular application domain to easily explore the existing research in that area.

1.4 Organization

This survey is organized into three major sections which discuss the above three ingredients of an outlier detection technique. In Section 2 we identify the various aspects that constitute an exact formulation of the problem. This section brings forward the richness and complexity of the problem domain. In Section 3 we describe the different application domains where outlier detection has been applied. We identify the unique characteristics of each domain and the different techniques which have been used for outlier detection in these domains. In Section 4, we categorize different outlier detection techniques based on the knowledge discipline they have been adopted from.

1.5 Related Work

As mentioned earlier, outlier detection techniques can be classified along several dimensions. The most extensive effort in this direction has been done by Hodge and Austin [2004]. But they have only focused on outlier detection techniques developed in machine learning and statistical domains. Most of the other reviews on outlier detection techniques have chosen to focus on a particular sub-area of the existing research. A short review of outlier detection algorithms using data mining techniques was presented by Petrovskiy [2003]. Markou and Singh presented an extensive review of novelty detection techniques using neural networks [Markou and Singh 2003a] and statistical approaches [Markou and Singh 2003b]. A review of selected outlier detection techniques, used for network intrusion detection, was presented by Lazarevic et al. [2003]. Outlier detection techniques developed specifically for system call intrusion detection have been reviewed by Forrest et al. [1999], and later by Snyder [2001] and Dasgupta and Nino [2000]. A substantial amount of research on outlier detection has been done in statistics and has been reviewed in several books [Rousseeuw and Leroy 1987; Barnett and Lewis 1994] as well as other reviews [Beckman and Cook 1983; Hawkins 1980]. Tang et al. [2006] provide a unification of several distance based outlier detection techniques. These related efforts have either provided a coarser classification of research done in this area or have focussed on a subset of the gamut of existing techniques. To the extent of our knowledge, our survey is the first attempt to provide a structured and a comprehensive overview of outlier detection techniques.

1.6 Terminology

Outlier detection and related concepts have been referred to as different entities in different areas. For the sake of better understandability, we will follow a uniform terminology in this survey. An *outlier detection problem* refers to the task of finding anomalous patterns in given data according to a particular definition of anomalous behavior. An *outlier* will refer to these anomalous patterns in the data. An *outlier detection technique* is a specific solution to an outlier detection problem. A *normal*

pattern refers to a pattern in the data which is not an outlier. The output of an outlier detection technique could be labeled patterns (outlier or normal). Some of the outlier detection techniques also assign a score to a pattern based on the degree to which the pattern is considered an outlier. Such a score is referred to as *outlier score*.

2. DIFFERENT ASPECTS OF AN OUTLIER DETECTION PROBLEM

This section identifies and discusses the different aspects of outlier detection. As mentioned earlier, a specific formulation of the problem is determined by several different factors such as the input data, the availability (or unavailability) of other resources as well as the constraints and requirements induced by the application domain. This section brings forth the richness in the problem domain and motivates the need for so many diverse techniques.

2.1 Input Data

A key component of any outlier detection technique is the input data in which it has to detect the outliers. Input is generally treated as a collection of **data objects** or **data instances** (also referred to as *record, point, vector, pattern, event, case, sample, observation, or entity*) [Tan et al. 2005a] . Each data instance can be described using a set of **attributes** (also referred to as *variable, characteristic, feature, field, or dimension*). The data instances can be of different types such as *binary*, *categorical* or *continuous*. Each data instance might consist of only one attribute (*univariate*) or multiple attributes (*multivariate*). In the case of multivariate data instances, all attributes might be of same type or might be a mixture of different data types.

One important observation here is that the features used by any outlier detection technique do not necessarily refer to the observable features in the given data set. Several techniques use preprocessing schemes like feature extraction [Addison et al. 1999], or construct more complex features from the observed features [Ertoz et al. 2004], and work with a set of features which are most likely to discriminate between the normal and outlying behaviors in the data. A key challenge for any outlier detection technique is to identify a best set of features which can allow the algorithm to give the best results in terms of accuracy as well as computational efficiency.

Input data can also be categorized based on the structure present among the data instances. Most of the existing outlier detection algorithms deal with data in which no structure is assumed among the data instances. We refer to such data as *point data*. Typical algorithms dealing with such data sets are found in network intrusion detection domain [Ertoz et al. 2004] or in medical records outlier detection domain [Laurikkala et al. 2000]. Data can also have a *spatial*, *sequential* or both type of structures. For *sequential data*, the data instances have an ordering defined such that every data instance occurs sequentially in the entire data set. Time-series data is the most popular example for this case and has been extensively analyzed with respect to outlier detection in statistics [Abraham and Chuang 1989; Abraham and Box 1979]. Recently, biological data domains such as genome sequences and protein sequences [Eisen et al. 1998; Teng 2003] have been explored for outlier detection. For *spatial data*, the data instances have a well defined spatial structure such that the location of a data instance with respect to others is significant and is

typically well-defined. Spatial data is popular in traffic analysis domain [Shekhar et al. 2001] and ecological and census studies [Kou et al. 2006]. Often, the data instances might also have a temporal (sequential) component giving rise to another category of *spatio-temporal* data, which is widely prevalent in climate data analysis [Blender et al. 1997]. Later in this section we will discuss the situations where the structure in data becomes relevant for outlier detection.

2.2 Type of Supervision

Besides the input data (or observations), an outlier detection algorithm might also have some additional information at its disposal. A labeled training data set is one such information which has been used extensively (primarily by outlier detection techniques based on concepts from machine learning [Mitchell 1997] and statistical learning theory [Vapnik 1995]). A training data set is required by techniques which involve building an explicit predictive model. The labels associated with a data instance denote if that instance is *normal* or *outlier*[1]. Based on the extent to which these labels are utilized, outlier detection techniques can be divided into three categories

2.2.1 *Supervised outlier detection techniques.* Such techniques assume the availability of a training data set which has labeled instances for normal as well as outlier class. Typical approach in such case is to build predictive models for both normal and outlier classes. Any unseen data instance is compared against the two models to determine which class it belongs to. Supervised outlier detection techniques have an explicit notion of the normal and outlier behavior and hence accurate models can be built. One drawback here is that accurately labeled training data might be prohibitively expensive to obtain. Labeling is often done manually by a human expert and hence requires a lot of effort to obtain the labeled training data set. Certain techniques inject artificial outliers in a normal data set to obtain a fully labeled training data set and then apply supervised outlier detection techniques to detect outliers in test data [Abe et al. 2006].

2.2.2 *Semi-Supervised outlier detection techniques.* Such techniques assume the availability of labeled instances for only one class. it is often difficult to collect labels for other class. For example, in space craft fault detection, an outlier scenario would signify an accident, which is not easy to model. The typical approach of such techniques is to model only the available class and decare any test instance which does not fit this model to belong to the other class.

Techniques that assume availability of only the outlier instances for training are not very popular. The primary reason for their limited popularity is that it is difficult to obtain a training data set which covers every possible outlying behavior that can occur in the data. The behaviors which do not exist in the training data will be harder to detect as outliers. Dasgupta et al [2000; 2002] have used only outlier instances for training. Similar semi-supervised techniques have also been applied for system call intrusion detection [Forrest et al. 1996].

On the other hand, techniques which model only the normal instances during training are more popular. Normal instances are relatively easy to obtain. More-

[1]Also referred to as normal and outlier classes

over, normal behavior is typically well-defined and hence it is easier to construct representative models for normal behavior from the training data. This setting is very similar to as **novelty detection** [Markou and Singh 2003a; 2003b] and is extensively used in damage and fault detection.

2.2.3 *Unsupervised outlier detection techniques.* The third category of techniques do not make any assumption about the availability of labeled training data. Thus these techniques are most widely applicable. The techniques in this category make other assumptions about the data. For example, parametric statistical techniques, assume a parametric distribution of one or both classes of instances. Similarly, several techniques make the basic assumption that normal instances are far more frequent than outliers. Thus a frequently occurring pattern is typically considered normal while a rare occurrence is an outlier. The unsupervised techniques typically suffer from higher false alarm rate, because often times the underlying assumptions do not hold true.

Availability of labels govern the above choice of operating modes for any technique. Typically, semi-supervised detection and unsupervised modes have been adopted more. Generally speaking, techniques which assume availability of outlier instances in training are not very popular. One of the reasons is that getting a labeled set of outlying data instances which cover all possible type of outlying behavior is difficult. Moreover, the outlying behavior is often dynamic in nature (for e.g - new types of outliers might arise, for which there is no labeled training data). In certain cases, such as air traffic safety, outlying instances would translate to airline accidents and hence will be very rare. Hence in such domains unsupervised or semi-supervised techniques with normal labels for training are preferred.

2.3 Type of Outlier

An important input to an outlier detection technique is the definition of the *desired outlier* which needs to be detected by the technique. Outliers can be classified into three categories based on its composition and its relation to rest of the data.

2.3.1 `Type I` *Outliers.* In a given set of data instances, an individual outlying instance is termed as a `Type I` outlier. This is the simplest type of outliers and is the focus of majority of existing outlier detection schemes. A data instance is an outlier due to its attribute values which are inconsistent with values taken by normal instances. Techniques that detect `Type I` outliers analyze the relation of an individual instance with respect to rest of the data instances (either in the training data or in the test data).

For example, in credit card fraud detection, each data instance typically represents a credit card transaction. For the sake of simplicity, let us assume that the data is defined using only two features – *time of the day* and *amount spent*. Figure 3 shows a sample plot of the 2-dimensional data instances. The curved surface represents the normal region for the data instances. The three transactions, o_1, o_2 and o_3 lie outside the boundary of the normal regions and hence are `Type I` outliers.

Similar example of this type can be found in medical records data [Laurikkala et al. 2000] where each data record corresponds to a patient. A single outlying record will be a `Type I` outlier and would be interesting as it would indicate some problem with a patient's health.

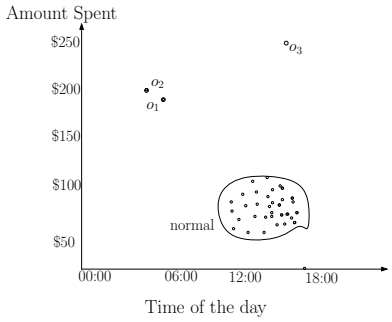


Fig. 3. `Type I` outliers o_1, o_2 and o_3 in a 2-dimensional credit card transaction data set. The normal transactions are for this data are typically during the day, between 11:00 AM and 6:00 PM and range between \$10 to \$100. Outliers o_1 and o_2 are fraudulent transactions which are outliers because they occur at an abnormal time and the amount is abnormally large. Outlier o_3 has unusually high amount spent, even though the time of transaction is normal.

2.3.2 `Type II` *Outliers.* These outliers are caused due to the occurrence of an individual data instance in a specific context in the given data. Like `Type I` outliers, these outliers are also individual data instances. The difference is that a `Type II` outlier might not be an outlier in a different context. Thus `Type II` outliers are defined with respect to a context. The notion of a context is induced by the structure in the data set and has to be specified as a part of the problem formulation. A context defines the neighborhood of a particular data instance.

`Type II` outliers satisfy two properties

1. The underlying data has a spatial/sequential nature: each data instance is defined using two sets of attributes, *viz. contextual attributes* and *behavioral attributes*. The contextual attributes define the position of an instance and are used to determine the context (or neighborhood) for that instance. For example, in spatial data sets, the longitude and latitude of a location are the contextual attributes. Or in a time-series data, time is a contextual attribute which determines the position of an instance on the entire sequence. The behavioral attributes define the non-contextual characteristics of an instance. For example, in a spatial data set describing the average rainfall of the entire world, the amount of rainfall at any location is a behavioral attribute.
2. The outlying behavior is determined using the values for the behavioral attributes within a specific context. A data instance might be a `Type II` outlier in a given context, but an identical data instance (in terms of behavioral attributes) could be considered normal in a different context.

`Type II` outliers have been most popularly explored in time-series data [Weigend et al. 1995; Salvador and Chan 2003] and spatial data [Kou et al. 2006; Shekhar et al. 2001]. Figure 4 shows one such example for a temperature time series which shows the monthly temperature of an area over last few years. A temperature of 35F might be normal during the winter (at time t_1) at that place, but the same value during summer (at time t_2) would be an outlier. Similar example can be found in credit card fraud detection domain. Let us extend the data described for `Type I` outliers by adding another attribute – *store name*, where the purchase was

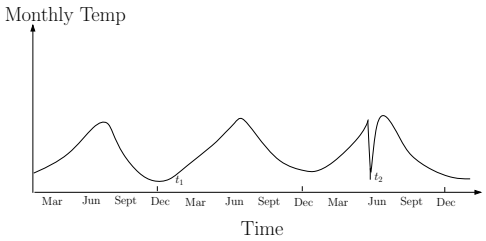


Fig. 4. `Type II` outlier t_2 in a temperature time series. Note that the temperature at time t_1 is same as that at time t_2 but occurs in a different context and hence is not considered as an outlier.

made. The individual might be spending around $10 at a gas station while she might be usually spending around $100 at a jewelery store. A new transaction of $100 at the gas station will be considered as a `Type II` outlier, since it does not conform to the normal behavior of the individual in the context of gas station (even though the same amount spent in the jewelery store will be considered normal).

2.3.3 `Type III` *Outliers.* These outliers occur because a subset of data instances are outlying with respect to the entire data set. The individual data instances in a `Type III` outlier are not outliers by themselves, but their occurrence together as a substructure is anomalous. `Type III` outliers are meaningful only when the data has spatial or sequential nature. These outliers are either anomalous subgraphs or subsequences occurring in the data. Figure 5 illustrates an example which shows a human electrocardiogram output [Keogh et al. 2002]. Note that the extended flat line denotes an outlier because the same low value exists for an abnormally long time.

Fig. 5. `Type III` outlier in an human electrocardiogram output. Note that the low value in the flatline also occurs in normal regions of the sequence.

We revisit the credit card fraud detection example for an illustration of `Type III` outliers. Let us assume that an individual normally makes purchases at a gas station, followed by a nearby grocery store and then at a nearby convenience store. A new sequence of credit card transactions which involve purchase at a gas station, followed by three more similar purchases at the same gas station that day, would indicate a potential card theft. This sequence of transactions is a `Type III` outlier. It should be observed that the individual transactions at the gas station would not be considered as a `Type I` outlier.

`Type III` outlier detection problem has been widely explored for sequential data such as operating system call data and genome sequences. For system call data, a particular sequence of operating system calls is treated as an outlier. Similarly, outlier detection techniques dealing with images detect regions in the image which are anomalous (`Type III` outliers).

It should be noted that `Type I` outliers may be detected in any type of data. `Type II` and `Type III` outliers require the presence of sequential or spatial structure in the data. But the nature of outliers actually required to be detected by a particular outlier detection algorithm is determined by the specific problem formulation. In certain cases `Type I` outliers are interesting. But in certain other cases, `Type II` or `Type III` outliers are more meaningful.

This classification of outliers is motivated from the domain of time-series outlier detection, where four different types of disturbances are studies, *viz.* — observational outliers, innovational outliers, mean (level) shifts, and temporary changes. Fox [1972] first proposed the idea of these different outliers for univariate time series. Tsay et al. [2000] generalized these outliers to multivariate time series. The different types of outliers refer to the different ways in which a disturbance can be modeled in a time series. An *observational outlier* is an outlying observation in the time series. An *innovational outlier* is an observation which affects the subsequent observations to be outlier as well. A *mean or level shift* occurs when the mean of the underlying generative process changes significantly. A *temporary change* occurs when the mean of the underlying generative process changes for a finite time and then returns to its normal value.

This classification is limited to statistical time-series modeling and outlier detection techniques dealing with time-series. The purpose of this classification is to identify how the outliers are induced in the data with respect to the underlying generative process. Our `Type I`, `Type II` and `Type III` classification is more general and covers all outlier detection techniques. Thus in our terminology, the above four outliers will still be `Type II` outliers occurring because of different reasons.

2.4 Output of Outlier Detection

The nature of outliers discussed above impose a requirement on the structure of the outlying patterns detected by the technique. Another requirement for any outlier detection technique is the manner in which the outliers are reported. Typically, outlier detection techniques fall into one of the following two categories

2.4.1 *Labeling Techniques.* The techniques in this category assign a label (*normal* or *outlier*) to each test instance. Thus they behave like a classification algorithm in this respect. If the test input is a set of instances, the technique provides a set of outliers and a set of normal instances.

The benefit of such techniques is that they provide an exact set of outliers for the analysts. The drawback of these techniques is that they do not differentiate between different outliers; no ranking among the outliers is provided. Often times, there is a confidence associated with a pattern being an outlier, and in such cases a zero-one decision is not feasible. This motivates the need for the scoring type of techniques discussed below.

2.4.2 *Scoring Techniques.* These techniques assign an outlier score to each pattern depending on the degree to which that pattern is considered an outlier. Thus the output of such techniques is a ranked list of outliers. An analyst may choose to either analyze top few outliers or use a cut-off threshold to select the outliers.

The drawback of a ranked list of outliers is the choice of the threshold to select a set of outliers. Often times choosing this threshold is not straightforward and has

to be arbitrarily fixed.

Besides defining the nature of data and outliers, the application domain can also impose certain constraints such as desired degree of accuracy and computational efficiency. In domains such as safety-critical systems, the accuracy of the algorithm is a foremost requirement. On the other hand, online systems such network intrusion detection systems require the algorithms to be efficient and scalable. Often times, the techniques have to maintain a balance between the accuracy and efficiency aspects of the solution. Recently privacy preservation of data has also become an important constraint in several domains and outlier detection algorithms have to address this problem [Vaidya and Clifton 2004].

3. APPLICATIONS OF OUTLIER DETECTION

The ability to detect outliers is a highly desirable feature in application domains for a variety of reasons. In this section we discuss some of the popular applications of outlier detection. For each application domain we discuss the significance of outliers, the challenges unique to that domain and the popular approaches for outlier detection that have been adopted in that domain.

3.1 Intrusion Detection

Intrusion detection refers to detection of malicious activity (break-ins, penetrations, and other forms of computer abuse) in a computer related system [Phoha 2002]. These malicious activities or *intrusions* are very interesting from a computer security perspective. An intrusion is different from the normal behavior of the system. This property allows a direct formulation of the problem as an outlier detection problem. Outlier detection techniques have been widely applied for intrusion detection.

Key challenge for outlier detection in this domain is the huge volume of data. The outlier detection schemes need to be computationally efficient to handle these large sized inputs. Moreover the data typically comes in a streaming fashion, thereby requiring on-line analysis. Another issue which arises because of the large sized input is the false alarm rate. Since the data amounts to millions of data object, a few percent of false alarms can make analysis overwhelming for an analyst. Typically, labeled data corresponding to normal behavior is readily available, while labels for intrusions are harder to obtain for training. Thus semi-supervised and unsupervised outlier detection techniques have been more favored in this domain.

Denning [1987] classifies intrusion detection systems into **host based** and **network based** intrusion detection systems.

3.1.1 *Host Based Intrusion Detection Systems.* These systems (also referred to as system call intrusion detection systems) deal with operating system call traces. These calls could be generated by programs [Hofmeyr et al. 1998] or by users [Lane and Brodley 1999]. The data is sequential in nature and the alphabet consists of individual system calls like The alphabet is usually small (183 for SunOS 4.1x Operating System). Different programs execute these system calls in different sequences. The length of the sequence for each program is varying. Figure 6 illustrates a sample set of operating system call sequences.

open,	read,	mmap,	mmap,	open,	read,	mmap	...
open,	mmap,	mmap,	read,	open,	close	...	
open,	close,	open,	close,	open,	mmap,	close	...

Fig. 6. A sample data set comprising of three operating system call traces.

Typically the intrusions are in the form of outlying subsequences (`Type III` outliers) of these traces which occur due to malicious programs, unauthorized behavior and policy violations. Thus all subsequences have events belonging to the same alphabet, but the co-occurrence of events is the key factor in differentiating between normal and outlying behavior. An important feature of the data in this domain is that the data can be typically profiled at different levels such as program level or user level.

One important aspect of outlier detection in this domain is to handle the sequential nature of data. This involves sequential data modeling as well as finding similarity between sequences and is further discussed in Section 13. After obtaining a representation of the sequential data, different outlier detection techniques may be used to obtain the outlying subsequences.

Technique Used	Section	References
Statistical Profiling using Histograms	Section 8.2.1	Forrest et al [1996; 2004; 1996; 1994; 1999],Hofmeyr et al. [1998],Kosoresow and Hofmeyr [1997],Jagadish et al. [1999],Cabrera et al. [2001],Gonzalez and Dasgupta [2003],Dasgupta et al [2000; 2002], Ghosh et al [1999a; 1998; 1999b], Debar et al. [1998],E. Eskin and Stolfo [2001],Marceau [2000],Endler [1998], Lane et al [1999; 1997b; 1997a]
Mixture of Models	Section 8.1.3	Eskin [2000]
Finite State Machines	Section 8.2.2	Ilgun et al. [1995],Michael and Ghosh [2000],Sekar et al. [2002]
Hidden Markov Models	Section 8.1.4	Gao et al. [2002]
Neural Networks	Section 5.1	Ghosh et al. [1998]
Support Vector Machines	Section 5.3	Hu et al. [2003]
Rule-based Systems	Section 5.4	Lee et al[1997; 1998; 2000]

Table 1. A categorization of different host based intrusion detection systems based on the outlier detection technique being used. These techniques are further discussed in Section 13.

A survey of different techniques used for this problem is presented by Snyder [2001]. The popular outlier detection techniques used in this domain are shown in Table 1.

3.1.2 *Network Intrusion Detection Systems.* These systems deal with detecting intrusions in network data. A typical setting is a large network of computers which is connected to the rest of the world via the Internet.

These systems work with data at different levels of granularity. This includes packet level traces, CISCO net-flows data etc. The data has a temporal aspect associated with it but most of the applications typically do not handle the sequential aspect explicitly. The data is high dimensional with typically a mix of categorical as well as continuous attributes. One major issue with this domain is that availability

Technique Used	Section	References
Statistical Profiling using Histograms	Section 8.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003], Sargor [1998]
Parametric Statistical Modeling	Section 8.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 8.2.3	Chow and Yeung [2002]
Finite State Machines	Section 8.2.2	Netstat [Vigna and Kemmerer 1999], Sekar et al [2002; 1999], Ray [2004]
Markov Models	Section 8.1.4	Ye [2004]
Bayesian Classification	Section 5.2	Siaterlis and Maglaris [2004], Sebyala et al. [2002], Valdes and Skinner [2000]
Neural Networks	Section 5.1	HIDE [Zhang et al. 2001], NSOM [Labib and Vemuri 2002], Smith et al. [2002], Hawkins et al. [2002], Kruegel et al. [2003], Manikopoulos and Papavassiliou [2002], Ramadas et al. [2003]
Support Vector Machines	Section 5.3	Eskin et al. [2002]
Rule-based Systems	Section 5.4	ADAM [Barbara et al. 2001a; Barbara et al. 2003; Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Clustering	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based Approaches	Section 7	MINDS [Ertoz et al. 2004], Eskin et al. [2002]
Spectral Decomposition	Section 10	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theory	Section 9	Lee and Xiang [2001],Noble and Cook [2003]

Table 2. A categorization of different network intrusion detection systems based on the outlier detection technique being used.

of accurately labeled training data is usually not possible. Another issue here is that the nature of outliers keeps on changing over time as the intruders adapt their network attacks to evade the existing intrusion detection solutions.

The intrusions typically occur as outlying patterns (`Type I` outliers) though certain schemes model the data in a sequential fashion and detect outlying subsequences (`Type III` outliers) [Gwadera et al. 2005b; 2004]. The primary reason for these outliers are due to the attacks launched by outside hackers who want gain unauthorized access to the network for information theft or to disrupt the network.

A brief comparison of the popular outlier detection techniques for network intrusion detection is presented by Lazarevic et al. [2003]. The popular outlier detection techniques in this domain are shown in Table 2.

3.2 Fraud Detection

Fraud detection refers to detection of criminal activities occurring in commercial organizations such as banks, credit card companies, insurance agencies, cell phone companies, stock market etc. The malicious users might be the actual customers

of the organization or might be posing as a customer (also known as *identity theft*). The fraud occurs when these users consume the resources provided by the organization in an unauthorized way. The organizations are interested in immediate detection of such frauds to prevent economic losses.

Fawcett and Provost [1999] introduce the term *activity monitoring* as a general approach to fraud detection in these domains. The typical approach of outlier detection techniques is to maintain a usage profile for each of the customer and monitor them to detect any deviations. Some of the specific applications of fraud detection are discussed below.

3.2.1 *Credit Card Fraud Detection.* This is a very important domain where credit card companies are interested in either detecting fraudulent credit card applications or fraudulent credit card usage (associated with credit card thefts). The first type of problem is similar to insurance fraud detection [Ghosh and Reilly 1994]. The second problem of detecting unauthorized credit card access is unique in the sense that it requires online detection of fraud as soon as the fraudulent transaction takes place. Outlier detection techniques have been applied in two different ways to address this problem. The first one is known as *by-owner* in which each credit card user is profiled based on his/her credit card usage history. Any new transaction is compared to the user's profile and flagged as an outlier if it does not match the profile. This approach is typically expensive since it requires querying a central data repository, every time a user makes a transaction. Another approach known as *by-operation* detects outliers from among transactions taking place at a specific location. Thus in this approach the algorithm is executed locally and hence is less expensive.

Technique Used	Section	References
Neural Networks	Section 5.1	CARDWATCH [Aleskerov et al. 1997], Ghosh and Reilly [1994],Brause et al. [1999],Dorronsoro et al. [1997]
Rule-based Systems	Section 5.4	Brause et al. [1999]
Clustering	Section 6	Bolton and Hand [1999]

Table 3. A categorization of different credit card fraud detection systems based on the outlier detection technique being used.

The data typically comprises of records defined over several dimensions such as the user ID, amount spent, time between consecutive card usage etc. The frauds are typically reflected in transactional records (`Type I` outliers) and correspond to high payments, purchase of items never purchased by the user before, high rate of purchase etc. The credit companies have complete data available and also have labeled records. Moreover, the data falls into distinct profiles based on the credit card user. Hence profiling and clustering based approaches are typically popular in this domain. Some of the popular approaches which have been applied to this domain are listed in Table 3.

3.2.2 *Mobile Phone Fraud Detection.* Mobile/cellular fraud detection is a typical activity monitoring problem. The task is to scan a large set of accounts,

examining the calling behavior of each, and to issue an alarm when an account appears to have been defrauded. Calling activity may be represented in various ways, but is usually described with call records. Each call record is a vector of features, both continuous (e.g., CALL-DURATION) and discrete (e.g., CALLING-CITY). However, there is no inherent primitive representation in this domain. Calls can be aggregated by time, for example into call-hours or call-days or user or area depending on the granularity desired. The outliers are typically `Type I` outliers and correspond to high volume of calls or calls made to unknown destinations etc. Some

Technique Used	Section	References
Statistical Profiling using Histograms	Section 8.2.1	Fawcett and Provost [1999],Cox et al. [1997]
Parametric Statistical Modeling	Section 8.1	Agarwal [2005],Scott [2001]
Markov Models	Section 8.1.4	Hollmen and Tresp [1999]
Neural Networks	Section 5.1	Barson et al. [1996],Taniguchi et al. [1998]
Rule-based Systems	Section 5.4	Phua et al. [2004],Taniguchi et al. [1998]
Visualization	Section 11	Cox et al. [1997]

Table 4. A categorization of different mobile phone fraud detection systems based on the outlier detection technique being used.

of the popular approaches applied to cell phone fraud detection are listed in Table 4.

3.2.3 *Insurance Claim Fraud Detection.* One of the most important problems in the property-casualty insurance industry is claims fraud. Individuals and conspiratorial rings of claimants and providers manipulate the claim processing system for unauthorized and illegal claims. One of the most prevalent forms of insurance frauds is found in the automobile insurance. Detection of such fraud has been very important for the associated companies to avoid financial losses.

The available data in this domain is obtained from the documents submitted by the claimants. Typically, claim adjusters and investigators assess these claims for frauds. Several fraud detection techniques treat these manually investigated cases as labeled instances and apply different supervised and semi-supervised techniques for fraud detection. The approaches extract different features (both categorical as well as continuous) from the claim related documents.

Insurance claim fraud detection is quite often handled as a generic activity monitoring problem [Fawcett and Provost 1999]. He et al. [2003] and Brockett et al. [1998] have applied neural network based techniques to identify outlying insurance claims.

3.2.4 *Insider Trading Detection.* Another recent application of outlier detection schemes has been in early detection of *Insider Trading*. Insider trading is a phenomenon found in stock markets, where people make illegal profits by acting on (or leaking) inside information before the information is made public. The inside information can be of different forms [Donoho 2004]. It could refer to the knowledge of a pending merger/acquisition, a terrorist attack affecting a particular industry, a pending legislation affecting a particular industry or any information which would

affect the stock prices in a particular industry. Early detection of insider trading done based on this information is required to prevent people/organizations from making illegal profits.

Insider trading results in outlying behavior in terms of trading activities in the market. The outlier detection schemes aim at capturing these outliers to detect insider trading. The available data is from several heterogenous sources such as option trading data, stock trading data, news. The data has temporal associations since the data is collected continuously. The temporal and streaming nature has also been exploited in certain schemes [Aggarwal 2005]. An outlier detection scheme identifies a set of features which would best discriminate the normal trading and insider trading behavior.

Insider trading detection is very recent application and thus has a limited outlier detection literature. Donoho [2004] and Aggarwal [2005] have applied statistical profiling techniques using histograms. Arning et al. [1996] have applied information theory based outlier detection technique to detect insider trading.

3.3 Medical and Public Health Data

Outlier detection in the medical and public health domains typically work with patient records. The data can have outliers due to several reasons such as abnormal patient condition or instrumentation errors or recording errors. Thus the outlier detection is a very critical problem in this domain and requires high degree of accuracy.

The data typically consists of records which may have several different types of features such as patient age, blood group, weight. The data might also have temporal as well as spatial aspect to it. Most of the current outlier detection schemes aim at detecting outlying records (`Type I` outliers). Typically the labeled data belongs to the healthy patients, hence most of the schemes adopt novelty detection approach in these cases. Some of the popular outlier detection approaches in this domain are listed in Table 5.

Technique Used	Section	References
Parametric Statistical Modeling	Section 8.1	Horn et al. [2001],Laurikkala et al. [2000],Solberg and Lahti [2005],Roberts [2002],Suzuki et al. [2003]
Neural Networks	Section 5.1	Campbell and Bennett [2001]
Rule-based Systems	Section 5.4	Aggarwal [2005]
Nearest Neighbor based Approaches	Section 7	Lin et al. [2005]

Table 5. A categorization of different outlier detection techniques being used in medical and public health domain.

3.4 Industrial Damage Detection

Industrial units suffer damage due to continuous usage and the normal wear and tear. Such damages need to be detected early to prevent further escalation and losses. The data in this domain is usually referred to as sensor data because it is recorded using different sensors and collected for analysis. Outlier detection techniques have been extensively applied in this domain to detect such damages.

Industrial damage detection can be further classified into two domains, one which deals with defects in mechanical components such as motors, engines etc and the other which deals with defects in physical structures.

3.4.1 *Fault Detection in Mechanical Units.* The outlier detection techniques in this domain monitor the performance of industrial components such as motors, turbines, oil flow in pipelines or other mechanical components and detect defects which might occur due to wear and tear or other unforseen circumstances. The data in this domain has typically a temporal aspect and time-series analysis is also used in some approaches [Keogh et al. 2002; Keogh et al. 2006; Basu and Meckesheimer 2007]. The outliers occur mostly because of an observation in a specific context (`Type II` outliers) or as an outlying sequence of observations (`Type III` outliers).

Technique Used	Section	References
Parametric Statistical Modeling	Section 8.1	Guttormsson et al. [1999], Keogh et al [1997; 2002; 2006]
Non-parametric Statistical Modeling	Section 8.2.3	Desforges et al. [1998]
Neural Networks	Section 5.1	Bishop [1994], Campbell and Bennett [2001], Diaz and Hollmen [2002], Harris [1993], Jakubek and Strasser [2002], King et al. [2002], Li et al. [2002], Petsche et al. [1996], Streifel et al. [1996], Whitehead and Hoyt [1993]
Spectral Decomposition	Section 10	Parra et al. [1996],Fujimaki et al. [2005]
Rule Based Systems	Section 5.4	Yairi et al. [2001]

Table 6. A categorization of different outlier detection techniques being used for fraud detection in mechanical units.

Typically, normal data (pertaining to components without defects) is readily available and hence novelty detection is the primary aim here. Neural networks have been used quite popularly in this domain. Some of the popular outlier detection techniques in this domain are listed in Table 6.

3.4.2 *Structural Defect Detection.* Structural defect and damage detection (such as cracks in beams, strains in airframes) is also similar to the industrial fault detection. The normal data and hence the models learnt are typically static over time. The data might have spatial correlations. The data typically has a few continuous features. The popular outlier detection techniques in this domain are listed in Table 7.

3.5 Image Processing

Outlier detection techniques dealing with images are either interested in any changes in an image over time (motion detection) or in regions which appear abnormal on the static image. This domain includes satellite imagery [Augusteijn and Folkert 2002; Byers and Raftery 1998; Moya et al. 1993; Torr and Murray 1993; Theiler and Cai 2003], digit recognition [Cun et al. 1990], spectroscopy [Chen et al. 2005; Davy and Godsill 2002; Hazel 2000; Scarth et al. 1995], mammographic image analysis [Spence et al. 2001; Tarassenko 1995], and video surveillance [Diehl and II 2002; Singh and Markou 2004; Pokrajac et al. 2007]. The data has spatial as well as

Technique Used	Section	References
Statistical Profiling using Histograms	Section 8.2.1	Manson [2002],Manson et al. [2001],Manson et al. [2000]
Parametric Statistical Modeling	Section 8.1	Ruotolo and Surace [1997]
Mixture of Models	Section 8.1.3	Hickinbotham et al [2000a; 2000b],Hollier and Austin [2002]
Neural Networks	Section 5.1	Brotherton et al [1998; 2001], Nairac et al [1999; 1997], Surace et al [1998; 1997], Sohn et al. [2001], Worden [1997]

Table 7. A categorization of different outlier detection techniques being used for structural damage detection.

Technique Used	Section	References
Mixture of Models	Section 8.1.3	Byers and Raftery [1998],Spence et al. [2001],Tarassenko [1995]
Regression	Section 8.1.2	Chen et al. [2005]
Bayesian Classification	Section 5.2	Diehl and II [2002]
Support Vector Machines	Section 5.3	Davy and Godsill [2002],Torr and Murray [1993],Song et al. [2002]
Neural Networks	Section 5.1	Augusteijn and Folkert [2002],Cun et al. [1990],Hazel [2000],Moya et al. [1993],Singh and Markou [2004]
Clustering	Section 6	Scarth et al. [1995]
Nearest Neighbor based Approaches	Section 7	Pokrajac et al. [2007]

Table 8. A categorization of different outlier detection techniques being used in image processing domain.

temporal characteristics. Each data point has a few continuous attributes such as color, lightness, texture etc. The interesting outliers are either anomalous points or regions in the images (`Type I` and `Type II` outliers). The outliers are caused by motion or insertion of foreign object or instrumentation errors. One of the key challenges in this domain is the large size of the input and hence computational efficiency of the outlier detection technique is an important issue.

The popular outlier detection techniques in this domain are listed in Table 8.

3.6 Novel Topic Detection in Text Data

Outlier detection techniques in this domain detect novel topics or events or news stories in a collection of documents or news articles. The data in this domain is typically high dimensional and very sparse. The data also has a temporal aspect since the documents are collected over time. The outliers are caused due to a new interesting event. The popular outlier detection techniques in this domain are listed in Table 9.

3.7 Other Domains

Outlier detection has also been applied to several other domains such as speech recognition [Albrecht et al. 2000; Emamian et al. 2000], novelty detection in robot behavior [Crook and Hayes 2001; Crook et al. 2002; Marsland et al. 1999; 2000b; 2000a], traffic monitoring [Shekhar et al. 2001], click through protection [Ihler et al.

Technique Used	Section	References
Mixture of Models	Section 8.1.3	Baker et al. [1999]
Statistical Profiling using Histograms	Section 8.2.1	Fawcett and Provost [1999]
Markov Models	Section 8.1.4	Yang et al. [2002]
Support Vector Machines	Section 5.3	Manevitz and Yousef [2002]
Neural Networks	Section 5.1	Manevitz and Yousef [2000]
Clustering	Section 6	Allan et al. [1998]

Table 9. A categorization of different outlier detection techniques being used for novelty topic detection in text data.

2006], detecting faults in web applications [Ide and Kashima 2004; Sun et al. 2005], detecting outliers in biological data [Kadota et al. 2003; Sun et al. 2006; Gwadera et al. 2005a; MacDonald and Ghosh 2007; Tomlins et al. 2005; Tibshirani and Hastie 2007], detecting outliers in census data [Lu et al. 2003], detecting associations among criminal activities [Lin and Brown 2003], detecting outliers in *Customer Relationship Management* (CRM) data [Zengyou He and Deng 2004b], detecting outliers in astronomical data [Dutta et al. 2007] and detecting ecosystem disturbances [Blender et al. 1997; Kou et al. 2006; Sun and Chawla 2004].

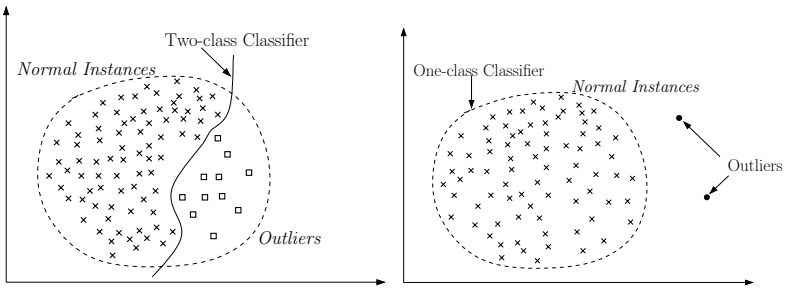
4. TECHNIQUES USED FOR OUTLIER DETECTION

Outlier detection has been an extensively explored problem. As described in Section 2, the different ways in which the problem can be formulated is huge, each with a unique set of inputs, requirements and constraints. Most of the existing techniques choose to address a specific problem by adopting concepts from a discipline of learning which appears to be best suited for the problem. In this section we classify outlier detection techniques based on the discipline from which they adopt their ideas. Some of the disciplines are very broad such as *Statistics* and *Information Theory*, while some of them are more of a sub-field such as *Classification* and *Clustering* but are so popularly used for outlier detection that we have dedicated a section to them.

Most of the techniques described in the next seven sections deal primarily with `Type I` outliers. Several of these techniques are easily extended to handle `Type II` and `Type III` outliers. There are techniques which have been proposed to exclusively handle these complex types of outliers. The discussion on how `Type II` and `Type III` outliers have been handled by different techniques is provided in Sections 12 and 13 respectively.

5. CLASSIFICATION BASED APPROACHES

Classification [Tan et al. 2005b; Duda et al. 2000] is an important machine learning and data mining concept. The primary aim of classification is to learn a set of labeled data instances (*training*) and then classify an unseen instance into one of the learnt class (*testing*). Outlier detection techniques based on classification also operate in the same two-phase fashion, using *normal* and *outlier* as the two



(a) Two-class Classification (b) One-class Classification

Fig. 7. Classification based approach for outlier detection.

classes[2]. The training phase builds a classification model using the available labeled training data. The testing phase classifies a test instance using the model learnt. The techniques following this approach fall under **supervised outlier detection techniques**. Certain techniques inject artificial outliers into normal training data and obtain fully labeled training data. Such approach has been discussed by Theiler and Cai [2003] and Abe et al. [2006], who use re-sampling techniques to inject artificial outliers in the training data.

In contrast with normal classification problems where one tries to distinguish between two (or more) classes of objects, *one-class classification* [Tax 2001] tries to describe one class of objects, and distinguish it from all other possible objects. A one-class classifier can then be trained to reject this object and to label it as an outlier. These techniques fall under the category of **semi-supervised outlier detection techniques**. The classification problem is modeled as a two-class problem where any new instance that does not belong to the learnt class is an outlier. In real scenarios, class labels for normal class are more readily available but there are also cases where only outlier class labels are available.

Figure 7 shows how classification techniques can be used for outlier detection. In the supervised mode, the two-class classifier learns a boundary between the two classes. Testing involves assigning a test instance to either side of this learnt class boundary. In the semi-supervised mode, the one-class classifier learns a boundary around the normal instances and declares any test instance outside this boundary as an outlier.

All techniques falling under this category follow the general scheme mentioned above but differ from each other along following dimensions

1. *What kind of classification model is used?.* We have further categorized classification based techniques into subcategories based on the type of classification model that they use. These include neural networks, Bayesian networks, *Support Vector Machines* (SVM), decision trees and regression models. We also include association analysis [Agrawal et al. 1993] based techniques in this category of outlier detection techniques, since they involve generating rules from the data which rep-

[2]In certain cases there might be more than one class representing the normal behavior making the problem a *multi-class* classification problem.

resents frequent behavior. These rules are then used to classify a new observation as normal or outliers.

2. *What labels are assumed to be available?.* Traditional classification algorithms assume availability of labeled instances for both classes for training. An outlier detection technique which assumes availability of both normal and outlier class labels for training can be addressed using *rare class classification algorithms* [Joshi et al. 2001; Phua et al. 2004; Weiss and Hirsh 1998; Vilalta and Ma 2002], since the outliers are far less than the normal class instances. These techniques fall under the category of **supervised outlier detection techniques**. This class of techniques will be ignored in this section since they are not very different from rare-class detection algorithms.

If only labels for only one class are available, one-class classification techniques are used. Typically the labels for normal instances are available, and hence outlier detection is analogous to novelty detection task [Markou and Singh 2003a]. Occasionally, the labels are available only for outliers. These techniques essentially learn signatures of known outlier patterns, and generalize them to detect outliers in test data. In system call intrusion detection domain, certain techniques learn *anomaly dictionaries* [Cabrera et al. 2001] from data containing labeled outlier instances.

Traditional classification as well as one-class classification assume availability of labeled training data for model building. But there are also techniques belonging to the class of **unsupervised outlier detection techniques** which can operate in an unsupervised mode without requiring any knowledge of class labels. Association analysis based rule generation algorithms assume that outlying instances do not belong to frequent patterns and hence are eliminated by simple thresholding during the rule generation step. Similarly, certain neural network techniques such as Self-organizing Maps (SOM) [Kohonen 1997] also perform unsupervised learning from the training data.

3. *When is an unseen data instance declared as an outlier?.* The testing phase of the classification based techniques assigns a class label of normal or outlier to an unseen data instance. For **supervised** techniques and **semi-supervised** techniques which learn outlier class only, an instance is predicted as an outlier. For several classifiers the prediction is a confidence of an instance [Zeevi et al. 1997] to belong to one of the classes. In such cases a threshold is required on the confidence to declare an instance as an actual outlier. Alternate methods have been applied to declare an unseen instance as an outlier. Brodley and Friedl [1996] propose to use an ensemble of classification algorithms and perform n-fold cross validation to detect instances which are misclassified by majority of the classifiers are removed as outliers from the training data. Tax and Duin [1998] propose an alternate method to determine outliers using semi-supervised classification approach by analyzing the instability in classification as a measure the likelihood of an instance to be an outlier.

The **semi-supervised** techniques which learn the normal class only, aim at learning the boundary around the normal instances and classify any instance falling outside this region as an outlier (*rejection*). But often times the unseen instance might belong to the normal class in which case the classifier has to update itself

to accommodate the new instance (*generalization*). The *generalization* vs *rejection* problem has been addressed by Chow [1970] and subsequently by Hansen et al. [1997]. This paper proposes an *optimum rejection rule* for any pattern recognition system using statistical approach. It derives a general relation between the error (misclassification) and reject (not classifying an instance to any of the class) probabilities for data which is uniform or normally distributed. The authors show the optimum rule is the one which for a given error rate gives the minimum rejection rate.

As mentioned earlier, the outlier detection techniques in this category adopt different classification based approaches to build classifiers – such as neural networks, decision trees, Bayes classifiers, regression models and support vector machines. We discuss outlier detection techniques using a specific classification algorithm in the following subsections.

5.1 Neural Networks Based

Neural Networks [Beale and Jackson 1990] are widely used for building classifiers by learning different weights associated with the network. A test instance when fed into this neural network may produce one of the output nodes as the output (determining the class of the test instance) or no output (determining that the test instance does not belong to any of the learnt classes). Neural networks have been extensively used for novelty detection [Markou and Singh 2003b; Odin and Addison 2000]. The basic idea is to train the neural network on the normal training data and then detect novelties by analyzing the response of the trained neural network to a test input. If the network accepts a test input, it is normal and if the network rejects a test input, it is an outlier [Stefano et al. 2000].

The above mentioned straight forward application of neural networks has been used for outlier detection in the *Neural Network Intrusion Detector* (NNID) [Ryan et al. 1998] system, for detecting outliers in medical diagnostics data [Roberts and Penny 1996], for detecting credit card fraud [Aleskerov et al. 1997; Dorronsoro et al. 1997] and for image sequence data [Singh and Markou 2004]. The NNID system trains a back-propagation network on the normal training data (a set of user commands) to identify the users who execute those commands. During testing a test instance is classified to belong to one of the learnt user profiles. If the output user does not match the actual user who generated that command, it signifies an intrusion. This is an example of a supervised approach, where each training instance has an associated class label (user name). Similarly, Manikopoulos et al [1991; 2002] propose a network intrusion detection system called *Hierarchical Intrusion Detection System* (HIDE). The authors compare the performances of five different types of neural networks: perceptron, back-propagation, *Perceptron Back-propagation Hybrid* (PBH), *Radial Based Function* (RBF), and fuzzy ARTMAP. Their experimental results showed that the BP and PBH networks had stronger classification.

The rich literature in the field of neural networks has resulted in an equally vast literature in the field of outlier detection. As was mentioned before, the numerous variations in an outlier detection problem means that a certain type of neural network is more suited for that problem than others. In the following subsections, we will see how different types of neural networks have been used for outlier detection.

Neural network based techniques for outlier detection can be classified into following categories based on the type of network used

5.1.1 *Multi Layered Perceptrons.* Multi Layered Perceptrons (MLP) are widely used neural networks for classification as well as outlier detection. MLP networks map the input data to a smaller number of output nodes. Thus they achieve dimensionality reduction as well as clustering of the data into fixed number of clusters. The most popular way to determine if a test instance is an outlier or not is by measuring the activation of the output nodes. If none of the output nodes are activated above a threshold, the input test instance is declared to be an outlier [Augusteijn and Folkert 2002]. Cun et al. [1990] enhance this criterion by using three conditions. The activity level of the winning node should be larger than a given threshold T_1, the activity level of the second winning node should be lower than a threshold T_2 and the absolute difference between T_1 and T_2 should be larger than a threshold T_d. All three thresholds are user-defined and in this study optimized using performance measures on the test set. The authors use this technique to detect novelties in handwriting recognition.

Sykacek [1997] introduces the concept of *equivalent error bars* which are used as a measure for outlier detection. Similar technique using MLPs for outlier detection has been applied for detecting intrusions in system call data [Ghosh et al. 1999a; Ghosh et al. 1998], fraud detection [Barson et al. 1996; He et al. 1997], detecting novelties in jet engine vibration data [Nairac et al. 1997] and for detecting strains in airframe structural data [Hickinbotham and Austin 2000b].

One disadvantage with MLP based outlier detection technique is that MLPs do not learn closed class boundaries [Moya et al. 1993]. Thus they tend to generalize and assign a novel instance to one of the learnt classes. This sometimes leads to missed outliers. This problem is addressed by Vasconcelos et al [1995; 1994]. A modification to the MLP structure is proposed to fix the problem of learning the boundaries well. It does that by constructing hyper-planes between different classes to better separate them from each other.

5.1.2 *Self Organizing Maps.* Self-organizing Maps (SOM) [Kohonen 1997] are unsupervised neural networks which cluster the input data into a fixed number of nodes. SOM is used for novelty detection in an engine health monitoring system [Harris 1993]. The SOM learns reference vectors from the normal data where each reference vector represents a cluster. For testing, the test input is compared to the reference vectors to assign it to one of the clusters (represented by a reference vector). If the input vector does not match any of the reference vectors it is declared to be an outlier. SOMs are used in a similar fashion to detect faults by a fault monitoring application [Ypma and Duin 1998; Emamian et al. 2000], for insurance fraud detection [Brockett et al. 1998] and to detect network intrusions [Labib and Vemuri 2002; Smith et al. 2002; Ramadas et al. 2003]. Ramadas et al. [2003] propose a system called *Anomalous Network-traffic Detection with Self Organizing Maps* (ANDSOM) which detects network anomalies by first training a SOM on the normal traffic (with 6 features). The testing involves feeding a test instance to the network which tries to assign it to one of the learnt clusters. If the distance to the best cluster is more than a pre-specified threshold, the instance is declared as an

outlier.

An interesting modification to SOMs for novelty detection is proposed by Theofilou et al. [2003]. This approach called *Long-term Depressant SOM* (LTD-SOM) works in opposite fashion to Kohonen SOM. Here the weight vectors (reference vectors) become dissimilar to the learnt vector. Thus a high match with the reference vector occurs only if the input vector is novel (or an outlier).

5.1.3 *Habituation Based.* Habituation based neural networks are quite similar to SOMs. The habituation based networks tend to ignore older patterns and give more weight to newly learnt instances by using memory leaking neurons. The application of habituation based SOMs (HSOM) for novelty detection in mobile robots has been proposed by Marsland et al [1999; 2000b; 2000a; 2002].

5.1.4 *Neural Trees.* A neural tree [Martinez 1998] is essentially a hierarchical quantization of the training data into equi-probable regions by fitting hyperplanes perpendicular to the coordinate axis. A neural tree works like an unsupervised hierarchical clustering algorithm. Martinez [1998] proposed building another neural tree for the test data and then comparing it with the learnt tree (from the training data) to detect novelties. The distance measure between two trees is obtained using either the *Kullback-Leibler Divergence* or *Log-likelihood Ratio*.

5.1.5 *Auto-associative Networks.* Auto-associative networks try to replicate the input exactly at the output layer. In other words, the internal layers in the network compress the input and then decompress it to recreate the input at the output layer. When an outlier is fed to such a system it fails to recreate it at the output and hence is detected. The difference between the input and the output is typically used to measure the likelihood of a test instance to be an outlier. These techniques essentially operate in a semi-supervised mode and assume availability of only normal instances for training.

The auto-associative network based technique proposed by Aeyels [1991], learns the principal components of the training data. After training, any input presented to the network produces one of the learned outputs, and the bitwise difference between input and output highlights novel components of the input. The paper also adds a "forgetting term" to discount old learnt patterns. Variants of this technique have been adopted in several other novelty detection techniques [Byungho and Sungzoon 1999; Japkowicz et al. 1995; Hawkins et al. 2002; Ko and Jacyna 2000; Manevitz and Yousef 2000; Petsche et al. 1996; Sohn et al. 2001; Song et al. 2001; Streifel et al. 1996; Thompson et al. 2002; Worden 1997]. Hawkins et al [2002; 2002] address the problem of outlier detection using *Replicator Neural Networks* which are based on auto-associative networks.

Diaz and Hollmen [2002] propose a different measure for the likelihood of a test instance to be an outlier, while using an auto-associative neural network. The authors use residual type statistics after fitting the neural network on the data to determine outlying instances. They apply this technique to detect outliers in vibration data for synchronous motors.

5.1.6 *Adaptive Resonance Theory Based.* Adaptive Resonance Theory has been shown to generate effective classifiers for novelty detection that have been shown

to outperform other classifiers such as SOM [Moya et al. 1993]. These are a type of self-organizing neural networks in which the different classes are also learnt online [Carpenter and Grossberg 1987]. ART based neural networks have also been used to learn sequential data and detect outlying test sequences [Dasgupta and Nino 2000; Caudell and Newman 1993].

5.1.7 *Radial Basis Function Based.* Generalized radial basis functions (RBF) neural networks have been proposed in several different applications for novelty detection [Albrecht et al. 2000; Bishop 1994; Brotherton et al. 1998; Brotherton and Johnson 2001; Li et al. 2002; Nairac et al. 1997; Ghosh and Reilly 1994]. Reverse connections from output layer to central layer are added, which makes it a self-organizing bayesian classifier, capable of novelty detection. Each neuron in the central layer has an associated normal distribution which is learnt from the training data. Any novel instance (outlier) does not fit any of these distributions and hence is not assigned to any output node. Nairac et al [1999; 1997] find the distance of a new instance from the nearest kernel center where the kernel represents the distribution at each neuron.

Jakubek and Strasser [2002] propose a technique which uses neural networks with ellipsoid basis functions for fault detection. The authors present their technique as a clustering algorithm. The ellipsoid basis function clusters similar instances together during training. A kernel function is estimated to represent the probability distribution for each cluster. Thus this technique resembles a mixture of models approach. The testing phase involves estimating the probability of occurrence of the test instance. A low value indicates that the new instance is an outlier.

5.1.8 *Hopfield Networks.* Hopfield networks have been applied to novelty detection in [Jagota 1991; Crook and Hayes 2001; Addison et al. 1999]. A Hopfield network with *Hebbian* learning rule is trained on the normal data. The energy of the Hopfield network because of the test stimuli is measured and used to determine if that test instance is novel or not. Crook et al [2001; 2002] apply this technique to detect novelties in a mobile robot's environment.

Boltzmann machines are typically viewed as the stochastic, generative counterparts of hopfield nets. Murray [2001] proposed a Boltzmann machine based novelty detection technique in embedded systems.

5.1.9 *Oscillatory Networks.* Oscillatory neural networks try to attain equilibrium between two types of neurons – excitory neurons and inhibitory neurons. When a new input is presented to an oscillatory neural network, the relaxation time of oscillatory behavior is used as a criterion for novelty detection [Ho and Rouat 1997; 1998; Kojima and Ito 1999; Borisyuk et al. 2000]. Ho et al [1997; 1998] use oscillatory neural networks using a *integrate and fire* neuron model. A test instance is fed to the trained model (using normal instances only). If the pattern has been learnt before the network reaches equilibrium quickly otherwise it takes a long time. The time taken by the network to reach equilibrium is a measure of novelty. The authors apply this technique for novelty detection in digit recognition task.

Martinelli and Perfetti [1994] described a *Cellular Neural Network* (CNN) for novelty. Each cell is connected to its neighboring inputs via an adaptive control op-

erator, and interacts with neighboring cells via nonlinear feedback. In the learning mode, the control operator is modified in correspondence to a given set of patterns applied at the input. In the application mode, the CNN behaves like a memoryless system, which detects novelty for those input patterns that cannot be explained as a linear combination of the learned patterns.

Weigend et al. [1995] propose a neural network based technique to detect outliers in turbine sensor data by using a mixture of experts. The authors show that using a mixture of these experts outperforms a single MLP based novelty detector. Similar technique is proposed by Zeevi et al. [1997] where the authors develop a mixture of experts (MEM) for time-series prediction. A new instance which cannot be predicted with a high confidence by the MEM is treated as a novel instance.

Zhang and Veenker [1991] propose an interesting technique where they differentiate between *passive learning* (learning using the examples provided by the environment or human) and *active learning* (generating training examples itself and learning from those). The authors propose the use of *genetic algorithms* to generate these training examples. This is done by combining two existing examples (parents) to create a novel training example (child). Thus the neural network based on this kind of learning becomes a self learning network.

5.2 Bayesian Network Based

A typical Bayesian network used for outlier detection aggregates information from different variables and provides an estimate on the expectancy of that event to belong to the normal class(es). Thus the training phase creates a tree type structure [Baker et al. 1999] where all child nodes are the variables (measuring the properties of the event) which feed the value to one root node for aggregation and classification of the event as normal or outlier. A very simple application of bayesian classification algorithm for novelty detection in video surveillance is proposed by Diehl and II [2002].

Naive Bayesian networks are also used to incorporate prior probabilities into a reasoning model which then classifies an event as normal or outlier based on the observed properties of the event and the prior probabilities [Sebyala et al. 2002]. Bayesian networks are sometimes also used to augment the capabilities of an existing statistical or any other outlier detection system.

Valdes and Skinner [2000] use Bayesian nets to create models of attack patterns along with the model of normal behavior. This approach prevents a system to "learn" an attack pattern as normal behavior. Similarly, Kruegel et al. [2003] use a bayesian network to models the causal dependencies between the different properties of the event and also any external knowledge about the measured properties to classify any event as normal or outlier.

Pseudo-Bayes estimators [Barbara et al. 2001b] are used to reduce the false alarm rate of an outlier detection system. The system learns the prior and posterior probabilities of unseen attacks from the training data. The instances classified as outliers by the system are then classified as "normal" or "new attacks" using the naive Bayes classifier. A *Denial of Service* (DOS) attack detection technique based on the Dempster's-Shafer's *Theory of Evidence* [Siaterlis and Maglaris 2004] involves data fusion from multiple sensor to obtain posterior probabilities and then using bayesian inference to estimate probability of an event in the monitored network.

5.3 Support Vector Machines

Support Vector Machines (SVMs) is a machine learning paradigm which is principally used as a binary classification tool. An SVM separates the data belonging to different classes by fitting a hyperplane between them which maximizes the separation. The performance of an SVM depends on how well-separable the data is. To overcome this, the data is mapped to a higher dimensional feature space where it can be easily separated by a hyperplane. But since finding the hyperplane requires the inner products between the vectors, explicit mapping is not essential. Instead a kernel function is used to approximate the dot product between the mapped vectors.

SVMs are applied for outlier detection in supervised mode [Steinwart et al. 2005]. In this paper, the authors discuss an unsupervised learning based technique which tries to learn the high density and the low density regions of data. Assuming that normal instances belong to the high density region while outliers belong to the low density region, the testing phase classies a test instance to one of these two classes and accordingly declares the instance as normal or outlying. The density level detection is done using a support vector machine.

SVMs have been applied to outlier detection by adapting them for single class classification (semi-supervised learning) [Ratsch et al. 2002]. Thus an intuitive technique would be to draw a smallest hyper-sphere[Tax and Duin 2001] which contains all points belonging to the normal class. Testing would simply involve determining which side of that hyper-sphere does a test point lie. Similar implementations are proposed for outlier detection in audio signal data [Davy and Godsill 2002], novelty detection in power generation plants [King et al. 2002] and system call intrusion detection [Eskin et al. 2002]. Ma and Perkins [2003a; 2003b] extend the one-class SVMs for outlier detection in temporal sequences . In this technique, the authors define a matching function to determine how well does a test sequence match the normal model. The matching function, $F(M_x(t_0-1), x(t_0))$, denoted as , is a function that can quantify how well the model $M_x(t_0)$ matches the temporal sequence.

Another variant of the above approach [Scholkopf et al. 1999] tries to separate the regions containing data from the regions containing no data. The primary assumption here is that the training data should be purely normal. The classifier thus obtained is a binary classifier which outputs +1 if the test instance falls inside one of the regions containing data and outputs -1 if the test instance falls inside one of the regions containing no data points. One drawback of this technique is pointed out [Campbell and Bennett 2001; Manevitz and Yousef 2002] that the performance hinges on the choice of the origin. More variants of this technique are discussed by Manevitz and Yousef [2002].

Robust Support Vector Machines (RSVM) [Song et al. 2002] aim at solving the over-fitting problem when outliers exist in the training data set. RSVM have been applied to system call intrusion detection by Hu et al. [2003].

5.4 Rule Based Models

Rule based techniques generate rules which either capture the normal behavior of a system [Skalak and Rissland 1990]. Any instance that is not covered by any such rule is considered as an outlier. Different techniques generate and use these rules

in different ways.

Classification techniques such as IREP and RIPPERk [Cohen 1995] can learn rules from noisy data by accommodating the outliers present in the training data. Fan et al [Fan et al. 2001] propose a supervised classification based outlier detection technique to detect network intrusions. Outlying instances are injected artificially so the classifier (RIPPER) can learn the boundaries between the two classes much better. Similarly, robust C4.5 algorithm [John 1995] is an adaptation of C4.5 algorithm to accommodate and detect outliers in the data. A similar approach is adopted by Abe et al. [2006] artificial instances of outlier class are injected in normal data to obtain a fully labeled training data set. The authors subsequently use an active learning technique (*Query by Bagging*) to classify test instances as normal or outlier.

Adaptation of multi-class classification algorithm to single-class classification problem can be used for semi-supervised learning of normal instances. The rules generated represent the normal behavior of the data and can be used to detect outlying instances during testing. Such techniques have been applied to detect intrusions in a sequence of system calls [Helmer et al. 1998; Lee et al. 1997; Salvador and Chan 2003; Teng et al. 1990].

Another possible way to generate rules is to use association rule mining which generates rules which have support above a specified threshold. An advantage of this technique is that it does not assume availability of class labels while training but utilizes the fact that outliers occur very rarely in the data set. Thus by judiciously choosing a support threshold, the rule generation algorithm can ensure that the outliers are not considered while generating rules.

An application of such technique for intrusion detection is proposed as the ADAM (Audit Data Analysis and Mining) system [Barbara et al. 2001a] and also by Otey et al. [2003] as an intrusion detection system embedded on the *Network Interface Card* (NIC). Similarly, LERAD [Mahoney and Chan 2002; 2003; Mahoney et al. 2003] finds association mining type rules from the training data of the form $P(\neg W|U)$ which denotes the conditional probability of one subset of attributes taking a particular set of values (denoted by W) given that a disjoint subset of attributes takes a particular set of values (denoted by U). In order to deal with the high number of rules that can be generated, it uses sampling and randomization techniques. Similar approach has been also adopted for credit card fraud detection [Brause et al. 1999] and for fraud detection in spacecraft house keeping data [Yairi et al. 2001].

Frequent itemsets are generated in the intermediate step of association rule mining algorithms. Zengyou He and Deng [2004a] proposes an outlier detection algorithm for categorical data sets by making the observation that an outlying transaction occurs in fewer frequent itemsets compared to a normal transaction. They propose a measure called *Frequent Pattern Outlier Factor* (FPOF) that ranks the transactions based on the number of frequent itemsets they occur in.

An extension of association mining to sequential data, known as *frequent episodes* [Mannila et al. 1997; Agrawal and Srikant 1995] has been proposed to generate rules from a sequence of events. These frequent episodes have been used to generate rules which represent the normal sequences in a sequence of system calls [Lee et al. 2000;

Lee and Stolfo 1998]. A test sequence which does not satisfy any of these rules is declared as an outlier. Qin and Hwang [2004] apply these frequent episodes to detect network intrusions.

Yamanishi et al. [2004] present a rule-based technique which can be used with the SmartSifter (discussed in Section 8) to detect outliers in an unsupervised mode. The outlier scores provided by the SmartSifter are used to label the data (as normal or outlier) and then use a supervised learning algorithm to learn a classifier from this data. This classifier is used to improve the accuracy of SmartSifter and to understand the nature of outliers.

6. CLUSTERING BASED APPROACHES

Cluster analysis [Jain and Dubes 1988] is a popular machine learning technique to group *similar*[3] data instances into *clusters*. It is either used as a stand-alone tool to get insight into the distribution of a data set, e.g. to focus further analysis and data processing, or as a preprocessing step for other algorithms operating on the detected clusters. Clustering is primarily an unsupervised technique though semi-supervised clustering [Basu et al. 2004] has also been explored lately. Clustering and outlier detection appear to be fundamentally very different from each other. While the former aims at detecting groups with similar behavior the latter aims at detecting instances which are not similar to any other instances in the data. But clustering based outlier detection techniques have been developed which make use of the fact that outliers do not belong to any cluster since they are very few and different from the normal instances.

The clustering based techniques involve a clustering step which partitions the data into groups which contain similar objects. The assumed behavior of outliers is that they either do not belong to any cluster, or belong to very small clusters, or are forced to belong to a cluster where they are very different from other members. Similarly, the normal instances belong to dense and large clusters. This distinction makes it possible to separate out the outliers from the rest of the data.

Clustering based outlier detection techniques can be broadly classified along following two dimensions

1. *What labels are assumed?.* Under this category, the techniques can be classifier as **semi-supervised** or **unsupervised**. The semi-supervised techniques typically use normal data to generate clusters which represent the normal modes of behavior of the data [Marchette 1999; Wu and Zhang 2003]. Any new test instance is assigned to one of the clusters. If the new instance is not close to any of the learnt clusters, it is termed outlier. This approach is applied for novelty detection task in different domains such as novel topic detection in news data [Allan et al. 1998]. He et al. [2002] incorporate the knowledge of labels to improve on their unsupervised clustering based outlier detection algorithm [He et al. 2003] by calculating a measure called *semantic outlier factor* which is high if the class label of an object in a cluster is different from the majority of the class labels in that cluster. A semi-supervised approach is proposed by Vinueza and Grudic [2004] where an instance can belong to one of the several class labels. The algorithm learns the parameters

[3]Based on a similarity measure

for the clusters representing each class. A distance measure is used to classify a test point to a cluster and declaring it as an outlier if it is far from all clusters or if within a cluster it is far from all other points. Thus this approach finds global as well as local outliers with respect to the cluster. Smith et al. [2002] studied *Self-Organizing Maps* (SOM), K-means Clustering, and *Expectation Maximization* (EM) to cluster training data and then use the clusters to classify test data.

Unsupervised techniques in this category use some known clustering algorithm and then analyze each instance in the data with respect to the clusters. A bootstrapping technique [Barbara et al. 2003] first separates normal data from outliers using frequent itemset mining. The data is divided into segments based on time. For each segment, frequent itemsets are generated. All itemsets which exist in more than one segment are considered normal. All data points corresponding to the frequent itemset are considered normal. Using the "clean" data, clusters are obtained using COOLCAT clustering technique [Barbara et al. 2002].

2. *How are the outliers detected?.* Most of the earlier clustering-based outlier detection techniques found outliers as the byproduct of a clustering algorithm [Tax and Duin 2001; Ester et al. 1996; Jain and Dubes 1988; Ng and Han 1994]. Thus any data point which does not fit in any cluster is called an outlier. Since the main aim is to find clusters, such approaches are not optimized to find outliers. The *FindOut* algorithm [Yu et al. 2002] is an extension of the *WaveCluster* algorithm [Sheikholeslami et al. 1998] in which the detected clusters are removed from the data and the residual points are declared as outliers. Other methods simply treat small clusters as outliers or look at the tightness of the clusters created [Pires and Santos-Pereira 2005; Otey et al. 2003].

Several clustering based techniques focus on detecting outliers. Thus the output of this techniques is actually the outliers and not the clusters. The CLAD algorithm [Mahoney et al. 2003] derives the width from the data by taking a random sample and calculating the average distance between the closest points. All those clusters whose density is lower than a threshold are declared as "local" outliers while all those clusters which are far away from other clusters are declared as "global" outliers. Similar approach is adopted in the *FindCBLOF* algorithm [He et al. 2003] which uses a clustering algorithm called *squeezer* [Zengyou et al. 2002] and determines the *Cluster-Based Local Outlier Factor* (CBLOF) for each point. A variant of the k-means clustering algorithm is used for outlier detection by Jiang et al. [2001] using a similar approach.

Clustering based approaches for anomaly detection have also been extended to time-series data [Blender et al. 1997] and other sequential data [Sequeira and Zaki 2002]. Blender et al. [1997] cluster cyclone paths and characterize different types of cyclones. These clusters are then manually analyzed to detect certain novel behavior in the cyclone behavior.

The advantage of the cluster based schemes is that they do not have to be supervised. Moreover, clustering based schemes are capable of being used in an incremental mode i.e after learning the clusters, new points can be fed in to the system and tested for outliers.

One disadvantage of clustering based approaches is that they are computationally expensive as they involve computation of pairwise distances. Fixed width clustering

is an approximation algorithm [Eskin et al. 2002; Portnoy et al. 2001; Mahoney et al. 2003; He et al. 2003]. A point is assigned to a cluster whose center is within a pre-specified distance to the point. If no such cluster exists then a new cluster with the point as the center is created. Then they determine which clusters are outliers based on their density and distance from other clusters. The width can either be a user-specified parameter [Eskin et al. 2002; Portnoy et al. 2001] or can be derived from the data [Mahoney et al. 2003]. Chaudhary et al. [2002] propose an outlier detection scheme using k-d trees which provide a partitioning of the data in linear time. They apply their approach to detect outliers in astronomical data sets where computational efficiency is an important requirement. Another technique which addresses this issue is proposed by Sun et al. [2004]. The authors propose an indexing technique called *CD-trees* to efficiently partition data into cells (or clusters). The data instances which belong to sparse cells are declared as outliers.

7. NEAREST NEIGHBOR BASED APPROACHES

Nearest neighbor analysis is a widely used concept in machine learning and data mining in which a data object is analyzed with respect to its nearest neighbors. This concept has been applied for different purposes such as classification, clustering and also outlier detection. The salient feature of nearest neighbor based outlier detection techniques is that they have an explicit notion of *proximity*, defined in the form of a distance or similarity measure for any two individual data instances, or a set of instances or a sequence of instances. These approaches typically map the data instances in a metric space defined over a finite number of features.

While clustering based schemes take a global view of the data, nearest neighbor based schemes analyze each object with respect to its local neighborhood. The basic idea behind such schemes is that an outlier will have a neighborhood where it will stand out, while a normal object will have a neighborhood where all its neighbors will be exactly like it. The obvious strength of these techniques is that they can work in an unsupervised mode, i.e. they do not assume availability of class labels. The various dimensions in which techniques under this category are different from each other are

1. *How is the distance/similarity computed?.* Different techniques compute distance (or similarity) between two data points in different ways. The metric chosen would depend on – type of features and number of features. For univariate and multivariate continuous attributes, Euclidean distance is a popular choice. More complex distance metrics are defined if the features are categorical. For sequences a distance metric between two sequences need to be defined. For spatial data, Kou et al. [2006] incorporate spatial correlation between data points while determining the distance.

Otey et al. [2006] propose a distance measure for data containing a mix of categorical and continuous attributes for outlier detection. The authors define links between two instances by adding distance for categorical and continuous attributes separately. For categorical attributes, the number of attributes for which the two instances have same values defines the distance between them. For continuous attributes, a covariance matrix is maintained to capture the dependencies between the continuous values.

2. *How is an instance declared as an outlier?.* Nearest neighbor based outlier detection techniques can be further categorized into two categories based on how the outliers are measured with respect to the nearest neighbors. The first category consists of techniques which measure the distance of an instance from its nearest neighbor set and apply different tests to detect outliers, such as – the instance is more than distance d from its closest point [Knorr and Ng 1998] or the instance is more than a distance d from its k-neighborhood [Eskin et al. 2002]. A popular definition of an outlier is [Knorr and Ng 1998; 1999; Knorr et al. 2000] – *A point p in a data set is an outlier with respect to the parameters k and λ, if no more than k points in the data set are at a distance λ or less from p.* The same authors [Knorr and Ng 1997] argue how this definition can be unified with any other definition of outliers defined in statistical community. They also propose a method to evaluate the obtained outliers and assign a strength measure to the outlier [Knorr and Ng 1999]. The drawback of this scheme is that it relies on the choice of the parameter λ. A simplified variant of this scheme, when k is 1, is applied to detecting shorted turns (outliers) in the DC field windings of large synchronous turbine-generators [Guttormsson et al. 1999].

A similar definition of outliers is [Ramaswamy et al. 2000] – *Given a k and n, a point p is an outlier if the distance to its k^{th} nearest neighbor is smaller than the corresponding value for no more than $n-1$ other points.* This scheme defines the outlier score of a point as the distance from its k^{th} nearest neighbor. This definition does not have an additional parameter as in the previous definition but relies on the value of the k (size of the nearest neighbor set). The authors use this definition to propose a *partition* based algorithm, which first clusters the points and computes lower and upper bounds on distance of a point from its k^{th} nearest neighbor for points in each partition. It then uses this information to identify the partitions that cannot possibly contain the top k outliers and prunes them. Outliers are then computed from the remaining points (belonging to unpruned partitions) in a final phase. This definition has been applied to credit card fraud detection [Bolton and Hand 1999], where a credit card user is analyzed with respect to its nearest neighbors (known as *peers*). A modification of this definition is applied by Eskin et al. [2002], Angiulli and Pizzuti [2002] and Zhang and Wang [2006], where the outlier score of a point is the sum of its distances from its k nearest neighbors. Zhang and Wang [2006] also propose a post processing of the outliers to identify the subspaces in which they exhibited outlying behavior. A similar approach to one proposed by Ramaswamy et al. [2000], called *Relative Density Factor* [Ren et al. 2004; Ren et al. 2004] uses a vertical data structure called P-tree to detect density based outliers in a more efficient manner.

Another technique is found in [Arning et al. 1996] where a minimum subset of points is obtained which causes the maximum deviation in the entire data. The points belonging to this subset are the outliers.

Saltenis [2004] presents an outlier detection method based on the distribution of pair-wise distances between data points. This method involves calculating all pair-wise distances between data points and a frequency function based on the distribution of distances. The basic idea is that, the distribution of distances between a data point and all other points will look similar to the cumulative distance dis-

tribution for all pair-wise distances if there are many other close-by points. This is an indirect way of detecting outliers. There might be points that are true outliers and for which the peaks in their distance distribution might match the peaks in the cumulative distribution. The techniques described above try to find global outliers

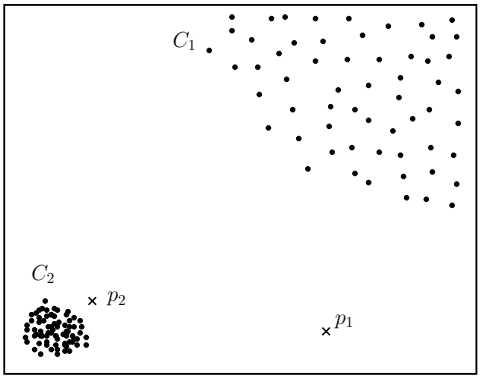


Fig. 8. A 2-D outlier example.

and perform poorly if the data has regions of different density. The second category of nearest neighbor based outlier detection techniques compute the density of regions in the data and declare the instances in low dense regions as outliers. Breunig et al. [2000] and Breunig et al. [1999] assign an outlier score to any given data point, known as *Local Outlier Factor* (LOF), depending on its distance from its local neighborhood. Thus this scheme finds the local outlier score of any data point and is not affected by the variations in the density of distribution of the data. The advantage of LOF over the simple nearest neighbor approach proposed by Ramaswamy et al. [2000] is illustrated in Figure 8. Due to the low density of the cluster C_1 it is apparent that for every example q inside the cluster C_1, the distance between the example q and its nearest neighbor is greater than the distance between the example p_2 and the nearest neighbor from the cluster C_2, and the example p_2 will not be considered as outlier. Therefore, the simple nearest neighbor approaches based on computing the distances fail in these scenarios. However, the example p_1 may be detected as outlier using only the distances to the nearest neighbor. On the other side, LOF is able to capture both outliers (p_1 and p_2) due to the fact that it considers the density around the points.

Jian Tang and W.Cheung [2002] discuss a variation of the LOF, which they call *Connectivity-based Outlier Factor* (COF). Instead of comparing the density of the point to its neighbors densities to calculate the outlier factor, the authors consider the *Set Based Nearest path* (SBN), which is basically a minimal spanning tree with k nodes, starting from the point in question. Outlier factor of a point is calculated using the SBN for that particular point and that of its neighbors. Both LOF and COF are very similar in nature. A variant of LOF is applied for detecting spatial outliers in climate data by Sun and Chawla [2004]. Yu et al. [2006] use similarity instead of distance to compute the nearest neighbors of any point in the data set. Similar technique has been proposed to detect sequential outliers in

protein sequences by Sun et al. [2006]. This approach uses *Probabilistic Suffix Trees* (PST) to find the nearest neighbors for a given sequence. Hautamaki et al. [2004] proposes a graph-based nearest neighbor approach, called *Outlier Detection using In-degree Number* (ODIN), which is essentially a distance based approach, where a neighborhood of a point is obtained using a graph. Pokrajac et al. [2007] extend LOF to work in an incremental fashion to detect outliers in video sensor data.

Papadimitriou et al. [2002] propose *Multi-granularity Deviation Factor* (MDEF) as a distance measure to detect outlier score of a point with respect to its neighborhood. This approach not only finds outlying points but also outlying micro-clusters. Instead of assigning an outlier score to a test point, the LOCI algorithm presented in paper a richer LOCI plot which contains information such as inter cluster distance, cluster diameter etc.

3. *Handling Computational Complexity.* A drawback of nearest neighbor based techniques is the $O(n^2)$ complexity required to compute the distances between every point. Most of the outlier detection techniques in this category involve computing nearest neighbors for each point. Angiulli and Pizzuti [2002] addresses this problem by linearizing the search space through the Hilbert space filling curve. The d-dimensional data set is fitted in a hypercube $D = [0, 1]^d$. This hypercube is then mapped to the interval $I = [0, 1]$ using the *Hilbert Space Filling Curve* and the k-nearest neighbors of a point are obtained by examining its successors and predecessors in I. Eskin et al. [2002] use canopy clusters [McCallum et al. 2000] as a tool to reduce the time to find the k-nearest neighbors. A similar clustering approach is proposed by Tao et al. [2006] to detect outliers in approximately linear time.

Bay and Schwabacher [2003] show that for a sufficiently randomized data, a simple pruning step could result in the average complexity of the nearest neighbor search to be nearly linear. After calculating the nearest neighbors for a point, the algorithm sets the outlier threshold for any point to the score of the weakest outlier found so far. Using this pruning the algorithm discards points which are close and hence not interesting. Similarly, there have been many other techniques which aim at improving the computational efficiency of the distance based outlier detection approach. Ghoting et al. [2006] propose *Recursive Binning and Re-Projection* (RBRP) algorithm, which aims at finding approximate nearest neighbors by first clustering (or binning) the data. The proposed algorithm reduces the search space for nearest neighbors by searching within a cluster for nearest neighbor of any data instance.

Jin et al. [2001] propose an interesting extension to the LOF approach, where the only the top n-outliers are found instead of finding LOF for every point. The approach includes finding micro-clusters in the data and then finding upper and lower bound on LOF for each of the micro-clusters. Chiu and chee Fu [2003] proposed three variants of LOF which enhance its performance by making certain assumptions about the problem. This gives a way to prune all those clusters which definitely do not contain points which will figure in the top n-outlier list. For the remaining clusters a detailed analysis is done to find the LOF score for each point in those clusters. Another approach to reduce the distance computation complexity is proposed in the LOADED algorithm [Ghoting et al. 2004], which uses frequent itemsets to estimate the distance between two instances. The technique is applicable

to categorical as well as continuous attributes.

The techniques discussed in previous two sections require a distance computation between two data points. Such techniques assume that the feature set that defines the data discriminates the outliers from normal points well enough. In situations where such assumptions about the feature set cannot be made, classification based or statistical techniques might be a better choice, since they are more robust to the choice of feature set.

8. STATISTICAL APPROACHES

The underlying principle of any statistical approach for outlier detection arises from the following definition - *An outlier is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed* [Anscombe and Guttman 1960]. Thus a statistical outlier detection technique can also be perceived as determining the generative probabilistic model (or estimating the probability distribution function for the data) and then testing if an instance is generated by that model or not.

Statistical outlier detection techniques are essentially model-based techniques; i.e. they assume or estimate a statistical model which captures the distribution of the data, and the data instances are evaluated with respect to how well they fit the model. If the probability of a data instance to be generated by this model is very low, the instance is deemed as an outlier.

The need for outlier detection was experienced by statisticians in as early as 19^{th} century [Edgeworth 1887]. The presence of outlying or discordant observations in the data biased the statistical analysis being performed on the data. This led to the notion of *accommodation or removal of outliers* in different statistical techniques (a detailed treatise on robust regression techniques which accommodate outliers in the data is given by Rousseeuw and Leroy [1987]). These techniques which were developed to accommodate or ignore outliers while performing some other form of statistical analysis gave rise to exact statistical methods which are meant to detect outliers in a given data set.

Like classification based approaches, these techniques typically operate in two phases - the *training phase* which comprises of estimating the statistical model (estimating the distribution parameters) and the *testing phase* where a test instance is compared to the model to determine if it is an outlier or not. The technique can estimate the probability density for either normal instances, or outliers (*semi-supervised techniques*), depending on the availability of labels. An *unsupervised technique* determines a statistical model which fits the majority of the observations and any observation which lies in a low probability region of the model is declared as an outlier.

The distinction in different techniques can arise due to the way each of the above two phases are handled:

1. *Building the probabilistic model (training phase).* Model fitting has been a very important statistical analysis tool. Two broad categories of model fitting techniques are:

- **Parametric Approaches** Several techniques assume that the data is generated from a known distribution. Thus the training phase involves estimating

the distribution parameters from the given sample. Several statistical tests, such as the frequently used *Grubb's* test [Grubbs 1969], assume a normal distribution of the data. Most of these techniques work with univariate as well as multivariate continuous data. Parametric regression modeling techniques have also been used to fit a regression model on the data [Rousseeuw and Leroy 1987]. For data with categorical attributes, a multinomial distribution might be assumed. Similarly, several techniques assume Markovian nature of the data when modeling sequential data [McCallum et al. 2000]. In real world scenarios, a single distribution does not effectively capture the actual data distribution. Thus several techniques assume that the data comes from a mixture of probability distributions [Eskin 2000] and thus the training phase involves estimating parameters for each of the probability distribution. Any parameter estimation technique can be used to estimate the parameters for each of the above cases [Duda et al. 2000; Baum et al. 1970; Jordan and Jacobs 1994].

- **Non-parametric Approaches** The other type of techniques do not assume the knowledge of the data distribution. One of the most widely used technique in this category is *histogram analysis* [Hofmeyr et al. 1998]. The model estimation here involves counting the frequency of occurrence of different data instances (thereby estimating the probability of occurrence of a data instance). Histogramming is very efficient for univariate data, but multivariate data induces additional complexities. Some techniques such as [Javitz and Valdes 1991] maintain histograms for each feature separately and thus detect outliers in each dimension independently. Another popular non-parametric technique to estimate the probability density is the *parzen windows* method [Desforges et al. 1998] which directly uses the samples drawn from an unknown distribution to model its density. These techniques can also be thought of *kernel based* approaches since they use a known kernel to model the samples and then extrapolate to the entire data.

2. *Determining outliers (testing phase).* Once the probabilistic model is known, the next step is to determine if a given data instance is an outlier with respect to the model or not. One of the approaches is to find the distance of the data instance from the estimated mean and declare any point above a threshold to be an outlier [Grubbs 1969]. This requires a threshold parameter to determine the length of the tail which has to be considered as an outlier.

A similar method is applied for multivariate data, by using a reduced sub-ordering [Barnett 1976] of each multivariate observation $\mathbf{x}_i$ to a scalar r_i as shown in Equation 1. These scalars can then be ordered as univariate observations.

$$r_i^2 = (\mathbf{x}_i - \mathbf{x}_0)'\Gamma^{-1}(\mathbf{x}_i - \mathbf{x}_0), \tag{1}$$

Here $\mathbf{x}_0$ is a reference point or an origin and Γ^{-1} weights variables inversely to their scatter. Different choices of these two parameters result in different distance metrics.

In certain cases, such as a mixture of models (see Figure 9), Markovian models or regression models, the distance from the mean might not be useful or might not make sense at all. In such cases, other statistical tests, such as the t-test [Ruotolo

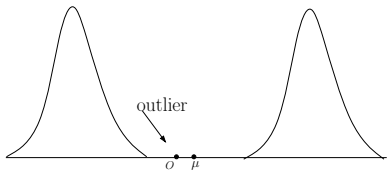


Fig. 9. A probability distribution of a data set containing two Gaussians. The outlier O is close to the mean μ of the distribution. Hence the distance from mean is not a good measure to detect outliers.

and Surace 1997], are used to determine if an observation comes from the same distribution as the training data.

Regression model based outlier detection techniques [Abraham and Chuang 1989] typically analyze the residuals obtained from the model fitting process to determine how outlying is an instance with respect to the fitted regression model.

Non-parametric techniques typically define a *distance* between a test observation and the statistical model and use some kind of threshold on this distance to determine if the observation is an outlier or not. Histogramming models compare the test instance to each of the categories in the histogram and test if it belongs to one of them or not [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991].

8.1 Parametric Approaches

As mentioned before, parametric techniques assume that the data is generated by a known distribution. Based on the type of distribution assumed, these techniques can be further categorized as follows

8.1.1 *Gaussian Models.* A substantial work has been done in detecting outliers in data which is assumed to be normally distributed. The training phase typically involves estimating mean and variance for the distribution using *Maximum Likelihood Estimates* MLE. For the testing phase, several statistical tests are discussed in [Barnett and Lewis 1994; Barnett 1976; Beckman and Cook 1983]. Four common outlier tests for normal distributions are the *Box-plot* rule [Laurikkala et al. 2000], the *Grubbs* test [Grubbs 1969; Stefansky 1972; Anscombe and Guttman 1960], *Rosner* test [Rosner 1983] and the *Dixon* test [Gibbons 1994]. The last two tests are variants of the *Grubbs* test.

A box plot is a way of summarizing data measured on an interval scale and is used often for exploratory data analysis. It is a type of graph (See Figure 10 for an example of a box plot) used to show the shape of distribution, its central value, its spread as well as the minimum and maximum values the data can take. Any data instance which lies beyond these extreme values are treated as outliers. The box plot rule has been applied to detect outliers for univariate and multivariate data in medical data [Laurikkala et al. 2000; Horn et al. 2001; Solberg and Lahti 2005], turbine rotors data [Guttormsson et al. 1999] and several other fields. For the multivariate case, the authors use Mahalanobis distance to reduce multivariate observations to univariate scalars. This is done by replacing $\mathbf{x}_0$ with the sample population mean and Γ with the sample covariance matrix as shown earlier in

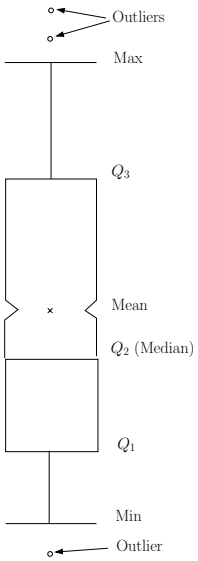


Fig. 10. A box plot for a univariate data set, showing the mean, median, min, max, 25% quartiles and outliers in the data

Equation 1. Sarawagi et al. [1998] apply similar concepts for detecting anomalies in an OLAP data cube.

The student's t-test has also been applied for outlier detection in [Surace and Worden 1998; Surace et al. 1997] to detect damages in structural beams. A normal sample, n_1 is compared with a test sample, n_2 using the t-test. If the test shows significant difference between them, it signifies an outlier in n_2. The multivariate version of student's t-test called the *Hotelling* t^2-test is also used as a outlier detection test statistic in [Liu and Weng 1991] to detect outliers in bioavailability/bioequivalence studies.

Grubb's test (also known as the *maximum normed residual test*) is used to detect outliers in a univariate data set [Grubbs 1969; Stefansky 1972; Anscombe and Guttman 1960]. It is based on the assumption of normality. This outlier is expunged from the data set and the test is iterated until no outliers are detected. However, multiple iterations change the probabilities of detection, and the test should not be used for sample sizes of six or less since it frequently tags most of the points as outliers. The two-way Grubb's test statistic is defined as

$$G = \frac{|Y_i - \bar{Y}|}{s} \tag{2}$$

with $\bar{Y}$ and s being the mean and standard deviation of the data respectively. *Grubb's* test has also been applied to detect spatial outliers in graph structured data [Shekhar et al. 2001]. The data in this case is spatial and multi variate and is assumed to be normally distributed in each feature. Aggarwal and Yu [2001] apply *Grubb's* test to multivariate data by counting the number of instances occurring in a d-dimensional cube (if the data has d attributes). The cubes are obtained by discretizing each attribute into k equi-depth bins.

In other similar approach [Abraham and Box 1979; Box and Tiao 1968; Agarwal 2005], the authors assume that the normal data is normally distributed with a distribution N(0,σ^2) and the outliers are also normally distributed around the same mean but with larger variance, N(0,$k^2\sigma^2$). The difference in these techniques from the Grubb's test lies in the testing phase. Using Bayesian inference, they try to estimate the probability of an observation to be generated by the outlier model. The authors assume that the priors of an observation being normal or outlier are known *apriori*.

Ye and Chen [2001] use a χ^2 statistic to determine outliers in operating system call data. The training phase assumes that the normal data has a multivariate normal distribution. The value of the χ^2 statistic is determined as:

$$X^2 = \sum_{i=1}^{n} \frac{(X_i - E_i)^2}{E_i} \tag{3}$$

where X_i is the observed value of the i_{th} variable, E_i is the expected value of the i_{th} variable (obtained from the training data) and n is the number of variables. A large value of X^2 denotes that the observed sample contains outliers.

Gwadera et al [2005b; 2004] adopt a Gaussian assumption to model normal "sequences" and estimate the parameters of the distribution. They then choose thresholds to declare a new subsequence to be normal or anomalous based on how well it fits the learnt normal model. This approach has been further discussed in the context of `Type III` outlier detection in Section 13.

Slippage Problem has been discussed in statistics to decide if one of the k populations has slipped to the right of the rest, under the *null hypothesis* that all of the populations are identical and continuous. Hawkins [1980] propose two test statistics - *Mosteller statistic* and *Doornbos statistic*, which are an extension of the slippage detection tests to outlier detection.

8.1.2 *Regression Models.* Outlier detection using regression has been extensively investigated for time-series data [Abraham and Chuang 1989; Abraham and Box 1979; Fox 1972]. Two types of outliers in time-series data have been identified

- *Observational outliers* - These occur when a single observation is extreme.
- *Innovational outliers* - These occur when an "innovation" at an instance is extreme which affects the observation at that instance as well as the subsequent observations.

The primary approach for detecting outliers in time-series data has been using regression analysis. The training phase involves fitting a regression model on the data. The testing phase is essentially a *model diagnostics* phase which involves evaluating each instance with respect to the model.

In several techniques [Abraham and Chuang 1989; Abraham and Box 1979; Fox 1972], the maximum likelihood estimates of the regression parameters are used as the criteria for outlier detection. The underlying approach in these techniques is to fit a regression model on the time-series and estimate certain statistics which are diagnosed to detect outliers in the time-series.

Some statisticians argue that using such significance tests are not always the best approach to detect outliers. Use of residuals obtained from regression model

fitting, to detect outliers, has been discussed in several approaches [Anscombe and Guttman 1960; Beckman and Cook 1983; Hawkins 1980]. *Studentized residuals* are also used to detect outliers in moving image data [Torr and Murray 1993]. Similarly, *AIC* (Akaike Information Content) has been used to detect outliers during model fitting [Kitagawa 1979]. Kadota et al. [2003] applied this approach to detecting outliers in gene micro-array data.

Another popular approach to handle outliers while fitting regression models is called *robust regression* [Rousseeuw and Leroy 1987] (estimation of regression parameters while accommodating outliers). The author argues that the robust regression techniques not only hide the outliers, but can also detect the outliers, because the outliers tend to have larger residuals from the robust fit. A similar robust outlier detection approach has been applied in ARIMA models [Bianco et al. 2001; Chen et al. 2005].

The above approaches are designed for univariate time-series, though some of them can be extended to handle multivariate time-series data as well. Detection of outliers in multivariate time-series data has been explicitly handled in several techniques. Tsay et al. [2000] discuss the additional complexity in multivariate time-series over the univariate time-series and come up with statistics (generalization of statistics proposed by Fox [1972]) which can be applied to detect outliers in multivariate ARIMA models. A similar approach is discussed by Galeano et al. [2004] with a difference that the authors consider a projection of the complete multivariate time-series while looking for outliers. The authors propose a technique called *projection pursuit* analyzes projections of the time series in a subset of features instead of looking at the entire data. The direction of projection is chosen by finding the projection which maximizes the *Kurtosis* coefficient (a measure for the degree of peakedness/flatness in the variable distribution) of the data. The actual outlier detection in that projection is done by using univariate test statistics [Fox 1972].

8.1.3 *Mixture of Parametric Models.* In several scenarios a single statistical model is not sufficient to represent the data. In such cases a mixture of parametric models is used. These techniques can work in two ways. First approach is *supervised* and involves modeling the normal instances and outliers as separate parametric distributions. The testing phase would involve determining which distribution the test instance belongs to. The second approach is *semi-supervised* and involves modeling the normal instances as a mixture of models. A test instance which does not belong to any of the learnt models is declared to be outlier.

Simplest application of the first approach is when both normal instances and outliers are modeled separately [Lauer 2001; Eskin 2000]. Eskin [2000] proposes to use *Expectation Maximization* (EM) algorithm to develop a mixture of models for the two classes, assuming that each data point is an outlier with a probability λ and normal with a probability $1 - \lambda$. Thus, if $\mathbf{D}$ represents the actual probability distribution of the entire data, and $\mathbf{M}$ and $\mathbf{A}$ represent the distributions of the normal and anomalous data respectively, then $\mathbf{D} = \lambda\mathbf{A} + (\mathbf{1} - \lambda)\mathbf{M}$. $\mathbf{M}$ is learnt using any machine learning technique while $\mathbf{A}$ is assumed to be uniform. Initially all points are considered to be in $\mathbf{M}$. The anomaly score is assigned to a point based on how much the distributions change if that point is removed from $\mathbf{M}$ and added to $\mathbf{A}$.

A similar approach is adopted by Baker et al. [1999] for novel event detection in text documents. Each of the known normal classes are assumed to be generated by some parametric model. These parameters are estimated using EM. For sparse classes, the authors adopt a technique called *hierarchical shrinkage* to estimate the parameters. The outlier detection is essentially a Bayesian classification task, where depending on the test document the authors predict if it belongs to a normal class or a novel class. Byers and Raftery [1998] apply the same concept to detecting mine-fields (outliers) in aircraft reconnaissance images. The authors assume that the normal and anomalous points are generated by two poisson distributions, whose parameters are estimated using EM. Agarwal [2006] proposed a similar approach for categorical data sets, in which the data is modeled as a mixture of two Gaussians, one for the normal instances and other for the outliers.

The second type of techniques in this category develop a mixture of models for only normal instances. A mixture of gaussian models to represent the normal instances has been used to detect strains in airframe data [Hickinbotham and Austin 2000a; Hollier and Austin 2002], to detect outliers in mammographic image analysis [Spence et al. 2001; Tarassenko 1995] and for network intrusion detection [Yamanishi and ichi Takeuchi 2001; Yamanishi et al. 2004]. Similar approach has been applied to detecting outliers in biomedical signal data [Roberts and Tarassenko 1994; Roberts 1999; 2002], where *extreme value statistics* are used to determine if a test point is an outlier with respect to the learnt mixture of models or not.

8.1.4 *Markov and Hidden Markov Models.* Markov [Duda et al. 2000; Ridgeway 1997] and Hidden Markov Models (HMMs) [Rabiner and Juang 1986; 1985] are the popular statistical techniques used to model sequential data. Variations of these models that follow the *Marokovian assumption* such as Maxent (maximum entropy models) [Pavlov and Pennock 2002; Pavlov 2003; McCallum et al. 2000], Conditional Random Fields [Lafferty et al. 2001], mixture of markov models [Zeevi et al. 1997; Jordan and Jacobs 1994; Smyth 1994] and mixture of HMMs [Smyth 1999; 1994] are also used to model sequential data. A substantial amount of work has been done in modeling sequential data using such models in biological sequences [Krogh et al. 1994; Gusfield 1997a; 1997b], speech recognition [Rabiner et al. 1989] and other domains. These techniques have been used to detect `Type II` and `Type II` outliers in sequential data and are further discussed in Sections 12 and 13 respectively.

8.2 Non-parametric Approaches

The outlier detection techniques in this category do not make any assumptions about the statistical distribution of the data. The most popular approaches for outlier detection have been *histograms* and *finite state automata (FSA)*.

8.2.1 *Histograms.* The most popular non-parametric statistical approach is to use histograms to maintain a profile of the normal data. Such approaches are also referred to as frequency based or counting based. The algorithms typically define a distance measure between a new test instance and the histogram based profile to determine if it is an outlier or not. The simplest form of histogram based outlier detection approach is when the data has a single feature. The training phase involves building histograms based on the different values taken by that feature in

the training data. The testing phase involves testing if the feature value in the test instance falls in any of the bins of the learnt histogram. The main challenges in these approaches are

- What features to maintain the histograms for? The features selected should be such that the outliers can be distinguished from normal instances based on those features.
- How to define a distance of a test instance from the profiles. In several cases, histograms are maintained for different features in the data. The distance of an instance from each of these histograms needs to be combined to get a single value.

The histogram based approaches are typically *semi-supervised*. They usually assume knowledge of normal labels [Anderson et al. 1994; Javitz and Valdes 1991; Helman and Bhangoo 1997] but there are also some which assume labels for outliers [Dasgupta and Nino 2000]. These approaches can be extended to operate in unsupervised mode under the assumption that the frequency of outliers is very low compared to the normal instances. In such cases, the histograms are dominated by the normal feature values.

These techniques are particularly popular in intrusion detection community [Eskin 2000; E. Eskin and Stolfo 2001] and fraud detection [Fawcett and Provost 1999], since the behavior of the data is governed by certain profiles (user or software or system) which can be efficiently captured using the histogram model. Denning [1987] describes an intrusion detection model which is based on maintaining histograms for different features and then using them to detect deviance from normal behavior.

Endler [1998] proposes to use the feature-wise histograms to estimate the likelihood for each feature. During testing, the likelihood values for the test data are estimated. A low likelihood value denotes higher outlier score of that test event. This paper applies this concept to detect intrusions in system call data. Similar approach is applied for fraud detection [Fawcett and Provost 1999], damage detection in structures [Manson 2002; Manson et al. 2001; Manson et al. 2000], network intrusion detection in IP networks [Ho et al. 1999; Yamanishi and ichi Takeuchi 2001; Yamanishi et al. 2004], detecting web-based attacks [Kruegel and Vigna 2003] and detecting novel topics in text data [Allan et al. 1998].

Kruegel and Vigna [2003; 2002] propose a histogram based outlier detection technique to detect web-based attacks. In this approach, several models are used to assign an outlier score to each test event. Each model involves comparing a set of attribute values (possibly a single attribute) with a normal historical profile for that set of attributes to find the likelihood probability for that model, p_m. The outlier score is assigned as,

$$Outlier\ Score = \sum_{m \epsilon Models} w_m * (1 - p_m)$$

where w_m specifies a confidence in a model, m. A variant of this approach is found in *Packet Header Anomaly Detection* (PHAD) and *Application Layer Anomaly Detection* (ALAD) [Mahoney and Chan 2002] which are network intrusion detection

systems. The authors argue that the network behavior is non-stationary with respect to time. So they propose a model in which an event is characterized by the time since it last occurs. So if a novel event occurs in a system, its outlier score also takes into account the time since last novel event occurred. The training phase involves estimating probabilities for different values of the attributes in a normal data set. During testing, outlier score is assigned to any event in which at least one attribute value did not occur in the training data, i.e the probability is 0. Thus, if $\{x_1, x_2 ... x_n\}$ are the attributes being monitored, then outlier score of the event is calculated as

$$Outlier\ Score = \sum_i \frac{t_i n_i}{r_i}$$

where attribute x_i takes a novel value. Here n_i refers to the number of events in training set, r_i is the number of different values taken by this attribute in the training set and t_i is the time since last occurrence of this novel value (If this is the first occurrence, then t_i is time since last occurrence of the last novel value detected for that attribute). Based on this approach the authors propose two different algorithms. In PHAD different attributes of a single packet are monitored. In ALAD, different features of an incoming TCP connection are treated as attributes.

The SRI International's real-time intrusion detection system (NIDES) [Anderson et al. 1994; Anderson et al. 1995; Porras and Neumann 1997] have a subsystem [Javitz and Valdes 1991] that maintains long-term statistical profiles that capture the normal behavior of a computer system. The authors propose *Q statistic* to compare a long-term profile with a short term profile (observation). This statistic is used to determine another measure called *S statistic* which reflects the extent to which the behavior in a particular feature is outlier with respect to the historical profile. The feature-wise *S statistic* are compared to get a single value called *IS statistic* which determines if a current observation is outlier or not. A variant has been proposed by Sargor [1998] for outlier detection in link-state routing protocols.

A substantial amount of research has been done in the field of outlier detection for sequential data (primarily to detect intrusions in computer system call data) using frequency based techniques. These algorithms are fundamentally similar to the instance based histogramming approaches as described above but are applied to sequential data to detect `Type III` outliers. These are discussed in Section 13.

8.2.2 *Finite State Machines.* A *Finite State Machine* (FSA) represents the model of behavior of data which has a temporal or sequential nature associated with it. Outlier detection techniques using FSA operate in a semi-supervised mode. The training phase involves identifying different states associated with the system and determining the state transition probabilites. Based on the labels available, these techniques develop finite state machine for normal or anomalous behavior. The drawback with these approaches is that they become too complicated if the number of possible states is large. The outlier detection techniques in this category primarily focus on detecting `Type II` or `Type III` outliers and are discussed in Sections 12 and 13 respectively.

8.2.3 *Kernel Functions.* A popular non-parametric approach to probability density estimation is *parzen windows estimation* [Parzen 1962]. This involves using

kernel functions to approximate the actual density distribution. Outlier detection techniques based on this method are similar to parametric methods described earlier. The only difference is the density estimation technique used. Desforges et al. [1998] proposed a semi-supervised probabilistic approach to detect novelties. Uses kernel functions to estimate the probability distribution function (*pdf*) for the normal instances. A new instance which lies in the low probability area of this *pdf* is declared to be novel. The approach discusses parametric as well as non-parametric estimation of pdf for univariate and multivariate data.

An instance of parzen window estimation for novelty detection is presented by Bishop [1994] for detecting novelties in oil flow data. A test instance is declared to be novel if it belongs to the low density area of the learnt density function. Similar application of parzen windows is proposed for network intrusion detection [Chow and Yeung 2002] and for mammographic image analysis [Tarassenko 1995].

9. INFORMATION THEORY BASED

Information Theory based techniques analyze the *information content* of a data set using different information theoretic measures such as *entropy, relative entropy etc.* The general idea behind these approaches is that outlying instances affect the information content of the data set because of their surprising nature. These approaches typically operate in an unsupervised mode. Lee and Xiang [2001] list different information theoretic measures which can used to detect outliers in a sequence of operating system call. These are *entropy, conditional entropy, relative conditional entropy, information gain*, and *information cost*. The general approach is to measure the regularity of a data set (using one of the aforementioned measures) with respect to each instance and classify the point as outlier if it induces irregularity in the data. Similar approach is adopted by Arning et al. [1996] who measure the overall *dissimilarity* (using the *Kolmogorov Complexity* [Li and Vitanyi 1993]) of a given data set. The algorithm tries to detect the smallest subset of the data, removing which results in the maximum reduction of the dissimilarity. Kolmogorov complexity is also used by Keogh et al. [2004] to detect outlying subsequences in a time-series.

He et al. [2005] find a k-sized subset from a given data set which when removed, makes the entropy of the remaining data set minimal. They use an approximate algorithm called *Local Search Algorithm* (LSA) He et al. [2006] to approximately determine this subset of outliers in a linear fashion.

A similar approach has been adopted to find surprising patterns in market basket data [Chakrabarti et al. 1998]. The authors encode a pattern using a set of bits and then observe the change in the information content of the pattern over time. A sudden change in the encoding of a pattern indicates a *surprising* change in the data. A variant of this technique is also applied to detect anomalous substructures in graph data by Noble and Cook [2003]. In this approach, the data is not temporal but spatial and for each substructure an encoding of its surroundings is determined. The substructures which require larger number of encoding bits are outliers, since they are different from their surroundings.

Another information theory based approach for spatial outlier detection has been proposed by Lin and Brown [2003]. The proposed approach declares a point outlier

if its set of nearest neighbors have a high relative uncertainty.

10. SPECTRAL DECOMPOSITION BASED

Spectral decomposition techniques in general deal with estimating the principle component vectors for a given data matrix. Thus in a way they try to detect the normal modes of behavior in the data (using the principle components) [Korn et al. 1997]. Several techniques use *Principal Component Analysis* (PCA) for dimensionality reduction before actual outlier detection to find a subset of features which capture the behavior of the data [Parra et al. 1996]. Spectral techniques can work in an unsupervised as well as semi-supervised setting.

The simplest approach detected in PCA literature for outlier detection is based on the fact that the top few principal components capture the bulk of variability in a given data set. Thus one would expect that the smallest principal components result in constant values. Thus any data point that violates this structure for the smallest components is an outlier. Dutta et al. [2007] adopt this approach to detect outliers in astronomy catalogs.

Shyu et al. [2003] proposed an outlier detection technique where the authors perform robust PCA to estimate the principal components from the correlation matrix of the normal training data. The testing phase involves comparing each point with the components and assigning an outlier score based on the point's distance from the principal components. Thus if the sample principal components of an observation x are y_1, y_2, ..., y_p and the corresponding eigen values are λ_1, λ_2, ..., λ_p, then

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} = \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + \ldots + \frac{y_q^2}{\lambda_q}, q \leq p \tag{4}$$

has a chi-square distribution. Using this result, the authors propose that, for a given significance level α, Observation x is an outlier if

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} > \chi_q^2(\alpha) \tag{5}$$

This approach has been applied to the network intrusion detection domain by several different groups [Shyu et al. 2003; Lakhina et al. 2005; Thottan and Ji 2003] and for detecting outliers in space craft components [Fujimaki et al. 2005].

Ide and Kashima [2004] apply this technique to detect anomalies in an activity graph modeling a set of computer nodes which interact with each other. Each graph is represented as an adjacency matrix for a given time. At every time instance the principle component of the matrix is chosen as the *activity vector* for the network. The time-series of the activity vectors is considered as a matrix and the principal left singular vector is obtained to capture the normal dependencies over time in the data. This vector is compared to a test activity vector (principle component vector for the test adjacency matrix) and the angle between the two vectors determines the anomaly score of the test graph.

Sun et al. [2007] propose an outlier detection technique using non-spectral matrix approximation. The data matrix is decomposed into a set of smaller matrices. An approximate of the original data matrix can be constructed from the decom-

positions. The authors monitor the residual error between original data and approximate data over time. A significant deviation from the normal would signal an anomaly in the data. The above technique is more suited when the data has massive anomalies (a lot of outliers) and aim is to detect the presence of outliers and not detect the actual outliers.

11. VISUALIZATION BASED APPROACHES

Visualization based approaches try to map the data in a coordinate space and detect the data instances which lie in sparse areas. A very simple visualization based approach to detect telecommunications fraud [Cox et al. 1997] which displays the call patterns of various users as a directed graph such that abnormal activity appears different on the display and can be visually identified by a user. Another approach is to find the depth contours of a given data set. The multivariate data is visualized as a cloud. The *depth* of a point with respect to this cloud represents how deep is that point located in the cloud. Ruts and Rousseeuw [1996] use the notion of *half space depth* of a point with respect to a data set. The points which have a smaller depth are declared to be outliers. The drawback of this approach is that calculation of *half space depth* is computationally very expensive even for bivariate data sets and hence infeasible to be extended to higher dimensions.

Haslett et al. [1991] propose several dynamic graphical techniques to explore spatial data and detect outliers. These include tools such as scatter plots, variogram clouds which allow spatial visualization of univariate as well as multivariate data and enable detection of outliers.

Bartkowiak and Szustalewicz [1997] use a visualization scheme called *grand tour* to detect the outliers. The authors impose a *concentration ellipse* on a given projection of the data in the grand tour, such that the points inside the ellipse belong to the majority pattern while points outside the ellipse are outlying in that projection. The number of times an instance is outlier in all projections of the data reflects the global outlier score of that instance.

12. HANDLING `TYPE II` OUTLIERS

The outlier detection techniques discussed in the previous sections primarily focus at detecting `Type I` outliers. In this section, we will discuss outlier detection techniques that handle `Type II` outliers.

As discussed in Section 2.3.2, `Type II` outliers require that the data has a set of *contextual attributes* (to define a context), and a set of *behavioral attributes* (to detect outliers within a context). Song et al. [2007] use the terms *environmental* and *indicator* attributes which are analogous to our terminology. `Type II` outlier detection techniques detect outliers within a given context. The contextual attributes provide a structure to the data, which can be of following types

1. *Spatial* — The data has spatial attributes, which define the location of a data instance and hence a spatial neighborhood. Several outlier detection techniques by [Lu et al. 2003; Shekhar et al. 2001; Kou et al. 2006; Sun and Chawla 2004] propose `Type II` outlier detection techniques for geographic information systems (GIS) data.

2. *Sequential* — The data is sequential, i.e. the contextual attributes of a data instance is its position in the sequence.
   Time-series data is a natural form of sequential data and several `Type II` outlier detection techniques [Abraham and Chuang 1989; Abraham and Box 1979; Rousseeuw and Leroy 1987; Bianco et al. 2001; Fox 1972; Salvador and Chan 2003; Tsay et al. 2000; Galeano et al. 2004; Zeevi et al. 1997] have been applied to time-series data.
   Another form of sequential data for which outlier detection techniques have been developed is event data, in which each event has a timestamp (such as operating system call data or web data [Ilgun et al. 1995; Vilalta and Ma 2002; Weiss and Hirsh 1998; Smyth 1994]). The difference between time-series data and event sequences is that for the latter, the inter-arrival time between consecutive events is uneven.
3. *Profile* — Often times the data might not have an explicit spatial or sequential structure, but can still be segmented or clustered into components using a set of contextual attributes. These attributes are typically used to profile and group users in *activity monitoring* systems, such as cell-phone fraud detection [Fawcett and Provost 1999; Teng et al. 1990], CRM databases [Zengyou He and Deng 2004b] and credit-card fraud detection [Bolton and Hand 1999]. The users are then analyzed within their group for outliers.

In comparison to the rich literature about `Type I` outlier detection techniques, the research on `Type II` outlier detection has been limited. Broadly, such techniques can be classified into following two categories.

12.1 Reduction to Type I Outlier Detection Problem

Since `Type II` outliers are individual data instances (like Type I outliers), but are outlying only with respect to a context, a popular approach is to apply a known `Type I` outlier detection technique within a context. Thus there are two steps involved — firstly, identifying a context and secondly detecting outliers within a context.

A straightforward example of this approach is applied to cell-phone fraud detection [Fawcett and Provost 1999]. The data in this case consists of cell-phone usage records. One of the attributes in the data is the cell-phone user which is used as the contextual attribute. The activity of each user is then monitored to detect outliers using other attributes. A similar approach is adopted for computer security [Teng et al. 1990], where the contextual attributes are – *user id, time of the day*. The remaining attributes are compared with existing rules representing normal behavior to detect outliers. *Peer group analysis* [Bolton and Hand 1999] is another similar approach where users are grouped together as *peers* and analyzed within a group for fraud. Zengyou He and Deng [2004b] propose the concept of *class outlier detection*, which is essentially segmenting the data using the class labels, and then applying a known clustering based outlier detection technique [He et al. 2002] to detect outliers within this subset.

Song et al. [2007] proposes a conditional anomaly detection technique for cases where identifying the context is not straightforward. The authors assume that the attributes are already partitioned into *contextual* and *behavioral* attributes. Thus

each data instance d can be represented as $[x, y]$. They partition the contextual data using a mixture of gaussian model, say U. The behavioral data is also partitioned using another mixture of gaussian model, say V. A mapping function, $p(V_j|U_i)$ is also learnt. This mapping indicates the probability of the indicator part of a data point y to be generated from a mixture component V_j, when the environmental part x is generated by U_i. Thus for a a given point $d = [x, y]$, the outlier score is given by

$$Outlier\ Score = \sum_{i=1}^{n_U} p(x \in U_i) \sum_{j=1}^{n_V} p(y \in V_j) p(V_j|U_i)$$

where n_U is the number of mixture components in U and N_V is the number of mixture components in V. $p(x \in U_i)$ indicates the probability of a sample point x to be generated from the mixture component U_i. $p(y \in U_j)$ indicates the probability of a sample point y to be generated from the mixture component V_j.

For spatial data, neighborhoods are intuitive and straightforward to detect [Ng and Han 1994] by using the location coordinates. Graph-based outlier detection [Shekhar et al. 2001; Lu et al. 2003; Kou et al. 2006] use Grubb's score [Grubbs 1969] or similar statistical `Type I` outlier detection techniques to detect outliers within a spatial neighborhood. Sun and Chawla [2004] use a distance based measure called *SLOM* (Spatial Local Outlier Measure [Sun and Chawla 2006]) to detect spatial outliers within a neighborhood.

A simple approach for outlier detection in time-series data is proposed by Basu and Meckesheimer [2007]. For a given instance in a time-series the authors compare the observed value to the median of the neighborhood values. A transformation technique for time-series data has been proposed by using phase spaces [Ma and Perkins 2003b]. This technique converts a time-series into a set of vectors by unfolding the time-series into a phase space using a time-delay embedding process. The temporal relations at any time instance are embedded in the phase vector for that instance. The authors use this technique to transform a time-series into feature space and then use one-class SVMs to detect outliers. Each outlier can be translated to a value at certain time instance in the original time-series.

12.2 Utilizing the Structure in Data

In several scenarios, breaking up data into contexts is not straightforward. This is typically true for time-series data and event sequence data. In such cases, time-series modeling and sequence modeling schemes are extended to detect `Type II` outliers in the data. Prominent among these are regression based approaches for time-series modeling such as robust regression [Rousseeuw and Leroy 1987], auto-regressive models [Fox 1972], ARMA models [Abraham and Chuang 1989; Abraham and Box 1979; Galeano et al. 2004; Zeevi et al. 1997] and ARIMA models [Bianco et al. 2001; Tsay et al. 2000]. Yi et al. [2000] extend this approach to detect `Type II` outliers in a set of co-evolving sequences by modeling the regression as well as correlation between the sequences.

One of the earliest works in time-series anomaly detection was proposed by Fox [1972], where a time-series was modeled as a stationary auto-regressive process. Any observation is tested to be outlier by comparing it with the covariance matrix of the auto-regressive process. If the observation falls outside the modeled error for

the process, it is declared to be an outlier. An extension to this approach is made by using *Support Vector Regression* to estimate the regression parameters and then using the learnt model to detect novelties in the data [Ma and Perkins 2003a].

An interesting approach to detect a single outlier (discord) in a sequence of alphabets was proposed by Keogh et al. [2004]. The technique adopts a divide and conquer approach. The sequence is divided into two parts and the *Kolmogorov Complexity* is calculated for each. The one with higher complexity contains the outlier. The sequence is recursively divided until they are left with a single event which is declared to be the outlier in the sequence.

Weiss and Hirsh [1998] propose a scheme to detect rare events in sequential data, where they use events occurring before a particular time to predict the event occurring at that time instance. If the prediction does not match the actual event, it is declared to be rare. This idea is extended in other areas, where the authors have used Frequent Itemset Mining [Vilalta and Ma 2002], *Finite State Automaton* (FSA) [Ilgun et al. 1995; Salvador and Chan 2003] and Markov Models [Smyth 1994] to determine conditional probabilities for events based on the history of events. Marceau [2000] use FSA to predict the next event of a sequence based on the previous n events. They apply this technique to the domain of system call intrusion detection. Hollmen and Tresp [1999] employ HMM for cell phone fraud detection. The authors use a *hierarchical regime switching call model* to model the cell phone activity of a user. The model predicts the probability of a fraud taking place for a call using the learnt model. The parameter estimation is done using the EM algorithm.

A more powerful model to detect intrusions in telephone networks was proposed by Scott [2001] and for modeling web click data by Ihler et al. [2006]. Both papers follow a technique in which they assume that the normal behavior in a time-series is generated by a non-stationary Poisson process while the outliers are generated by a homogenous Poisson process. The transition between normal and outlying behavior is modeled using a Markov process. The proposed technique in each of these papers use *Markov Chain Monte Carlo* (MCMC) estimation technique to estimate the parameters for these processes. For testing, a time series is modeled using this process and the time instances for which the outlying behavior was active are considered as outliers.

Sun et al. [2005] utilize the bipartite graph structure in P2P networks to first identify a neighborhood for any node in the graph, and then detecting the relevance of that node within the neighborhood. A node with a low relevance score is treated as an outlier. The authors also propose an approximate technique where the graph is first partitioned into non-overlapping subgraphs using graph partitioning algorithm such as METIS Karypis and Kumar [1998]. The neighborhood of a node is then computed within its partition.

Reducing a `Type II` outlier detection problem to a `Type I` outlier detection problem allows the use of the vast existing research on `Type I` outlier detection. In certain cases, though, this reduction is not straightforward. Often times the reduction results in loss of certain structural aspect in the data. In such cases, modeling the structure in the data and using the model to detect outliers is often used.

13. HANDLING TYPE III OUTLIERS

This section discusses the outlier detection techniques which focus on detecting Type III outliers. As mentioned earlier, Type III outliers are a subset of instances that occur together as a substructure and whose occurrence is not normal with respect to a normal behavior. The individual instances belonging to this substructure are not necessarily outliers by themselves, but it is their co-occurrence in a particular form that makes them outlier. Type III outlier detection problem is more challenging than Type I and Type II because it involves exploring the structure in the data for outlying regions.

A primary data requirement for Type III outlier detection, is the presence of structure in the data. Based on this requirement, there are two categories of Type III outlier detection techniques

1. *Sequential* Type III *Outlier Detection Techniques.* These techniques work with sequential data and find subsequences as outliers (also referred to as *sequential outliers*). Typical data sets include event sequence data, such as system call data [Forrest et al. 1999] or numerical time-series data [Chan and Mahoney 2005].

2. *Spatial* Type III *Outlier Detection Techniques.* These techniques work with spatial data and find connected subgraphs or subregions within the data as outliers (also referred to as *spatial outliers*). Outlier detection techniques have been applied to multi-spectral imagery data [Hazel 2000] and graph data [Noble and Cook 2003].

Substantial research has been done in the field of sequential outlier detection; this can be attributed to the existence of sequential data in several important application domains. Spatial outlier detection has been explored mostly in the domain of image analysis. The following subsections discuss each of these categories in detail.

13.1 Handling Sequential Outliers

As mentioned earlier, Type III outlier detection in sequence data involves detecting sequences that are outliers with respect to a definition of normal behavior. Sequence data is very common in a wide range of domains where a natural ordering is imposed on data instances by either time or position. In outlier detection literature, two types of sequences are dealt with

1. Each element of the sequence is either a symbol belonging to a finite alphabet or an event belonging to a finite set of events. Biological sequences such as protein sequences [Sun et al. 2006], operating system call sequences [Hofmeyr et al. 1998], text document data [Gwadera et al. 2005b] are the examples of first category. Web behavior data [Cadez et al. 2000], market basket transaction data [Chakrabarti et al. 1998] etc are the examples of second category. Techniques dealing with this type of data assume that the sequences do not have any outlying symbols or events (or remove them using a Type I outlier technique as a pre-processing step). Chakrabarti et al. [1998] deal with sequences of market basket itemsets, where each itemset has multiple items.
2. Each event is a continuous observation (univariate or multivariate). Time-series data is a typical example of such data sets and have been explored for outlier detection in domains such as traffic data monitoring, click through protection Ihler et al. [2006], climate data [Blender et al. 1997] etc. Time-series data has

been extensively explored in the `Type II` outlier detection category. In the `Type III` problem domain, the outliers are observations that occur together and hence are considered anomalous. Ihler et al. [2006] monitor freeway traffic as a time series of counts of vehicles passing under a sensor. The technique proposed in this paper focusses in detecting sustained outlying events (very high or very low traffic volumes for a substantial period of time) and ignore occasional spikes. Keogh et al. [2002] convert a continuous time-series into a string of alphabets and then employ techniques to detect `Type III` outliers in the transformed data.

In several of the techniques that will be discussed below, a short pattern or sequence is tested for occurrence within a long sequence. Most of the techniques assume that a sequence occurs within another sequence only if it occurs as a substring within the longer sequence. But the occurrence of a sequence s in a sequence T ($|s| < |T|$), can be defined in multiple ways, as defined below

- *s is a substring of T.* This is true when there exists an integer i, such that
  $1 \leq i \leq |T| - |s|$ and
  $s_1 = T_i, s_2 = T_{i+1}, \ldots, s_{|s|} = T_{i+|s|}$
  For system call intrusion detection as well as in biological domains, this definition makes most sense and presence of another symbol within s alters the nature of s significantly.
- *s is a subsequence of T.* This is true when there exists a set of integers i_1, i_1, $\ldots$ $i_{|s|}$ such that the following conditions hold
  $1 \leq i_1 < i_2 < \ldots < i_{|s|}$ and
  $s_1 = T_{i_1}, s_2 = T_{i_2}, \ldots, s_{|s|} = T_{i_{|s|}}$
  In domains like market transaction data and network intrusion detection, this definition is more appropriate. For example, consider the connections made to a web-server inside a network. If the sequence of connections is —
  {`buffer overflow`, `remote login`, `ftp`}
  This would indicate that an external attacker hacked into the web-server and is stealing unauthorized data. Now consider the same sequence interleaved with normal requests to the web-server —
  {`buffer overflow`, *http request*, *http request*, `remote login`, *http request*, `ftp`}
  Even this sequence should be considered as an outlier because it still signifies an intrusion. This formulation poses an additional challenge that now the window in which s can occur is larger than the length of s itself.
- *Any permutation of s is a subsequence of T.* This is true when there exists a set of integers i_1, i_1, $\ldots$ $i_{|s|}$ such that
  $1 \leq i_1 < |T|, 1 \leq i_2 < |T|, \ldots 1 \leq i_{|s|} < |T|$ and
  $s_1 = T_{i_1}, s_2 = T_{i_2}, \ldots, s_{|s|} = T_{i_{|s|}}$
  This formulation is the most inclusive definition of a subregion and is appropriate in domains where the relative ordering of events in a subregion is not important. For example, in market transaction data, suppose a customer purchases a gun and then purchases bullets. This could be equivalent to the case where the customer first purchases bullets and then purchases a gun.

The outlier detection problem for sequences can be defined in different ways and are discussed below.

13.1.1 *Detecting outlier sequence in a set of sequences*

DEFINITION 1. *Given a set of n sequences,* $\Im = \{T_1, T_2, \ldots, T_n\}$*, and a query sequence* S*, find if* S *is an outlier with respect to* $\Im$*.*

If each sequence in $\Im$ is labeled as either normal or outlier, the problem formulation is similar to a `Type I` **supervised outlier detection** problem. Similarly, if only one type of labels are available for sequences in $\Im$, the problem formulation is similar to `Type I` **semi-supervised outlier detection**. In the absence of labels and if $S \in \Im$, this formulation is similar to a `Type I` **unsupervised outlier detection** problem.

Key challenges faced by techniques which follow Definition 1 are

- The sequences in $\Im$ are not of equal length.
- The sequences are not aligned. This is a fundamental problem with biological sequences [Gusfield 1997a] where different sequence alignment and sequence matching techniques are explored.

Techniques addressing this problem follow one of the following two approaches

1. *Reduction to Type I outlier detection problem.* A general approach to solve the above problem would be to transform the sequences in $\Im$ as well as S to a feature space and then use a `Type I` outlier detection technique to detect outliers.

Certain techniques assume that each sequence in $\Im$ is of equal length. Thus they treat each sequence as a vector of attributes and employ a `Type I` outlier detection technique to detect outliers. For example, if a data set contains length 10 sequences, they can be treated as data records with 10 features. A similarity or distance measure can be defined between a pair of sequences and any `Type I` outlier detection technique can be applied to such data sets. This approach has been adopted for time-series data sets [Caudell and Newman 1993; Blender et al. 1997]. In the former paper, the authors apply ART (Adaptive Resonance Theory) neural networks based outlier detection technique to detect outliers in a time-series data set, while the latter paper uses a clustering based outlier detection technique to identify cyclone regimes (outliers) in weather data.

As mentioned earlier, the sequences in $\Im$ as well as the sequence S may not be of equal length. Certain techniques address this issue by transforming each sequence into a record of equal number of attributes. A transformation technique has been proposed for multiple time-series data [Chan and Mahoney 2005], known as *Box Modeling*. In a box model, for each time-series, each instance of this time-series is assigned to a box depending on its value. These boxes are then treated as features (the number of boxes is the number of features in the transformed feature space). The authors then apply `Type I` outlier detection techniques — a Euclidean distance based technique and a classification based technique using RIPPER to detect outlying time series in the data.

2. *Modeling Sequences.* The transformations discussed in the previous section are appropriate when all the sequences are properly aligned. Often times the align-

ment assumption becomes too prohibitive. Research dealing with system call data, biological data explore other alternatives to detect `Type III` outliers. Two approaches have been followed in this category.

The first approach builds a model for normal behavior (and/or outliers depending on availability of labels) from the given set of sequences $\mathcal{S}$ and then compares the query sequence S to this model. If S does not match the model it is declared as an outlier.

Sequential association modeling has been used to generate sequential rules from sequences [Teng et al. 1990]. The authors use an approach called *time-based inductive learning* to generate rules from the set of normal sequences $\mathcal{S}$. The test sequence is compared to these rules and is declared an outlier if it contains patterns for which no rules have been generated.

Markovian modeling of sequences has been the most popular approach in this category. The modeling techniques used in this category range from *Finite State Automatons* (FSA) to markov models. Sekar et al [2002; 1999] propose a technique using FSA to model network protocol behavior. Outliers are detected when a given sequence of events, S does not result in reaching one of the final states. The authors also apply their technique to operating system call intrusion detection [Sekar et al. 2001].

Ye [2004] proposes a simple 1-order markov chain modeling approach to detect if a given sequence S is an outlier w.r.t $\mathcal{S}$. The author determines the likelihood of S, $P(S)$ using the following equation

$$P(S) = q_{S_1} \prod_{t=2}^{|S|} p_{S_{t-1}S_t}$$

where q_{S_1} is the probability of observing the symbol S_1 in $\mathcal{S}$ and $p_{S_{t-1}S_t}$ is the probability of observing the symbol S_t after symbol S_{t-1} in $\mathcal{S}$. The inverse of $P(S)$ is the outlier score for S. The drawback of this technique is that single order markov chain cannot model higher order dependencies in the sequences.

Forrest et al. [1999] propose a *Hidden Markov Model* (HMM) based technique to detect outlying program traces in operating system call data. The authors train an HMM using the sequences in $\mathcal{S}$. The authors propose two testing techniques. In the first technique they compute the likelihood of a sequence S to be generated by the learnt HMM using the *Viterbi* algorithm. The second technique is to use the underlying *Finite State Automaton* (FSA) of the HMM. The state transitions and the outputs made by the HMM to produce the test sequence are recorded. The authors count the number of times the HMM had to make an unlikely state transition or output an unlikely symbol (using a user-defined threshold) as mismatches. The total number of mismatches denote the outlier score for that sequence.

A *Probabilistic Suffix Trees* (PST) is another modeling tool which has been applied to detect `Type III` outliers in sequential databases. A PST is a compact representation of a variable order markov chain. Yang and Wang [2003] use PST to cluster sequences and detect outlier sequences as a by-product. Similarly, Smyth [1997] and Cadez et al. [2000] use HMMs to cluster the set of sequences $\mathcal{S}$ and detect any sequences which do not belong to any cluster as outliers.

Another modeling tool used for sequential outlier detection is *Sparse Markov*

Trees (SMT), which is similar to a PST with the difference that it allows wild card symbols within a path. This technique has been used by E. Eskin and Stolfo [2001], who train a mixture of SMT using the training set $\Im$. Each SMT has a different location of wildcards. Testing phase involves predicting the probability $P(S_n|S_{n-1}\dots S_1)$ using the best SMT from the mixture. If this probability is below a certain threshold, S is declared as an outlier.

Often times a sequence S is outlier in a small portion of the entire sequence. Eskin et al. [2000] discuss the phenomenon of *sparseness* in real sequences, where only a fraction of a sequence carries significant statistical information about the outlier characteristics of the entire sequence. Comparing the entire sequence to a model might average out the effect of the outlying subregion. This problem has been discussed in the context of operating system call data by Snyder [2001]. This has motivated the need for a second approach towards handling the sequential outlier detection problem defined in Definition 1. The authors argue that the outlying portion of S usually occurs together in a short consecutive region within S. This observation has resulted in research which essentially comprises of three steps

1. In the first step short sequences are extracted from the query sequence S
2. The second step is to determine if any of these short sequences are outlier with respect to $\Im$ by solving the outlier detection problem formulated as following

   DEFINITION 2. *Given a short query pattern s and a set of $n \geq 1$ sequences $\Im$, determine if s is an outlier with respect to $\Im$.*

   This definition can also be treated as a pattern matching problem, where a given query pattern is matched with a reference data set. For time-series data it is also referred to as *query by content* [Keogh and Smyth 1997; Faloutsos et al. 1994].
3. The third step involves combining the results of the second step and determining an outlier score for the entire test sequence S.

The above mentioned three step approach has been adopted by several techniques, specifically in the domain of operating system call intrusion detection. Forrest et al. [2004] describe a formal framework for outlier detection using this general approach. The basic template followed to solve this problem is to build a model from the training data $\Im$ and then compare each short subsequence in S to this model to determine its outlier score (or if it is an outlier or not). The results for all subsequences are then combined to provide a single score or decision for the entire sequence S. We will discuss each of the three steps in following three sections

1. *Extracting Short Sequences from A Long Sequence.* A straightforward technique to extract short sequences from a long sequence is to break the long sequence into smaller non-overlapping segments of a fixed length. This technique is applied by Debar et al. [1998] for operating system call data. The drawback of this technique is that a lot of information lies across two segments which is lost when they are treated separately.

One of the most widely used technique to extract patterns from a long sequence is a *sliding window*. The sliding window is a simple technique, in which a window of a fixed size, w, is moved over the sequence, k event at a time and the subsequence

under the window is recorded. This is illustrated in Figure 11, where a sliding window with $w = 3$ and $k = 1$, is used to extract 7 subsequences from a sequence of length 10. Typically k is set to 1. Note that if $k = w$, then the extracted subsequences are non-overlapping segments of the original sequence.The width of the window is a key challenge with the sliding window approach. If the window size is chosen to be very small, the number of subsequences generated will be huge and might not be representative enough for normal or outlying behavior. A large window size might put both normal and outlying behaviors in the same subsequence, thereby making it hard for the outlier detection algorithm to detect them. Hofmeyr et al. [1998] have empirically shown that a window of width $w = 6$ ($k = 1$) is optimal for operating system call intrusion detection.

E. Eskin and Stolfo [2001] propose an entropy modeling approach to determine the optimal window size for a given sequence data set. The authors use the prediction model in whereby the probability of w^{th} system call is estimated from the previous w - 1 system calls. To determine the optimal value of w, they measure the regularity of the data using the following equation

$$H(\Sigma|S^{w-1}) = -\sum_{x\in\Sigma, s\in S^{w-1}} P(x,s)\log_2 P(x|s)$$

where Σ denotes the alphabet and S^{w-1} is the set of all sequences of length w-1 in the normal data. Thus the quantity in the above equation measures the *conditional entropy* of data for a given value of w. The authors compute this quantity for different values of w and choose the one which has lowest entropy.

Qiao et al. [2002] and Zhang et al. [2003] propose using HMM to transform the original set of sequences before applying the sliding window technique to extract short sequences. Qiao et al. [2002] use the normal sequences to train an HMM (using the *Baum-Welch* algorithm). Each training sequence is then fed into this HMM and the most probable sequence of state transitions is obtained. Any repeating state in the sequence is ignored. Thus for each training sequence a possibly shorted state transition sequence is generated. Same is done for the test sequence. Zhang et al. [2003] use two level of HMMs in a similar fashion to transform the input sequences into state transition sequences.

2. *Addressing Definition 2.* The second step is to build a model from the training data and then comparing a test subsequence to the training data. The model could be built for normal behavior or outlying behavior or for both depending on the availability of appropriately labeled data. Typically, model of normalcy is built using the training data $\Im$. Certain techniques which build classifiers, model both behaviors [Lee et al. 1997; Lee and Stolfo 1998]. Certain earlier techniques modeled only the outlying behavior [Dasgupta and Nino 2000; Forrest et al. 1994].

Models can be built from the training data in two ways. One is to directly build a model from the sequences in $\Im$. Sun et al. [2006] generate a PST from $\Im$ with maximum depth constrained at L (where $L <$ length of sequences in $\Im$). The authors also prune the PST by applying a threshold on number of times a suffix is observed in the training data. The testing is done by computing

$$P^T(s_1)P^T(s_2|s_1)\dots P^T(s_L|s_1s_2\dots s_{L-1})$$

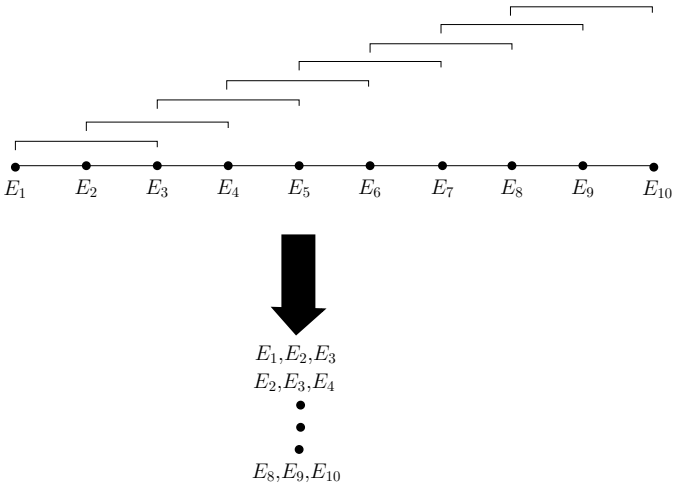


Fig. 11. A sliding window of width 3 used to extract subsequences from a long event sequence.

where $s_1 s_2 \dots s_L$ is the subsequence of length L and $P^T(s_i|s_1 s_2 \dots s_{i-1})$ is the probability of observing the symbol s_i after a suffix $s_1 s_2 \dots s_{i-1}$ in the PST denoted by T. Marceau [2000] also use suffix trees to build a *Deterministic Finite Automaton* (DFA) from $\Im$ and use the probability $P^T(s_L|s_1 s_2 \dots s_{L-1})$ as outlier score for the subsequence of length L.

The second method to build models is by first extracting short subsequences from the training data $\Im$, and then building a model using this set of short sequences.

Building dictionaries (or collections) of normal short subsequences is the most straightforward model adopted in operating system call intrusion detection domain [Hofmeyr et al. 1998; Forrest et al. 1994; Endler 1998; Debar et al. 1998; Ghosh et al. 1999b; 1999a; Lane and Brodley 1999; 1997b; Cabrera et al. 2001; Dasgupta and Nino 2000]. The only parameter to these techniques is the length of the short sequences to be extracted. Storing these dictionaries can be done efficiently by using suffix trees as shown by Debar et al. [1998]. The testing of a short subsequence with a given set of short subsequences can be done in one of the following three ways —

1. *Exact Matches* If the test subsequence is not found in the dictionary it is an outlier. This approach is used by Endler [1998],Dasgupta and Nino [2000] and Ghosh et al [1999a; 1999b].
2. *Number of Mismatches* The test subsequence is compared to each subsequence in the dictionary. For each dictionary subsequence, the number of mismatches are counted (*Hamming Distance*). The distance of the test subsequence to its closest dictionary subsequence is its outlier score. This approach is used by Hofmeyr et al. [1998], Debar et al. [1998], Cabrera et al. [2001] and Forrest et al. [1994]. This method is appropriate when then all subsequences are of equal length. To avoid this assumption, Lane et al [1999; 1997b] propose a suite of similarity measures where they are interested in the number of matches. The inverse of similarity is treated as the outlier score of the test subsequence.

Forrest et al. [1996] proposed use of *lookahead pairs* to model the short subsequences. They first extract the subsequences of a certain length w from the training data. For each symbol which occurs in the first position of a subsequence, the authors count the occurrence of other symbols at each of the next w - 1 positions. Thus for each symbol ($\in \Sigma$), the authors calculate $(w-1)|\Sigma|$ lookup pairs. For testing, w - 1 pairs of symbols (using the first symbol and every other symbol) is checked if it exists in the lookahead pair profile constructed from the training data. The number of mismatching pairs is the outlier score for the test subsequence.

Lee et al [1997; 1998] treat each short subsequence as a data instance with as many features as the length of the subsequence. They use RIPPER [Cohen 1995] to build a classifier from this training data set. Li et al. [2007] use a linear time hierarchical classifier called CHIP. Ghosh et al. [1999a] use two different neural networks (*Feedforward networks* and *Elman networks*) as classifiers. Testing comprises of classifying the test subsequence as normal or outlier. Gao et al. [2002] propose constructing HMMs for normal and outlying subsequences in the training data set. These HMMs can then be used to classify an subsequence as normal or outlier.

Finite state machines are a popular tool to model discrete sequences. Michael and Ghosh [2000] propose a technique to construct a DFA from the extracted short sequences. The authors first extract segments from the training data of length $n+l$. Each state of the DFA corresponds to one or more subsequences of length n. The outgoing transitions from a state are labeled with subsequences of length l that follow one of the n length subsequences in the state. A similar DFA construction is also discussed by Kosoresow and Hofmeyr [1997]. In Michael and Ghosh [2000], the authors also provide a slightly different DFA in which they force one state to represent only a unique n length subsequence. The testing is done by running the entire test sequence S through this DFA. At each step a segment of length $n+l$ is considered. A normal sequence reaches the final state of the DFA without making any improbable transitions. An outlier sequence would either not find a state to transition to, or would make a transition which occurs rarely in the DFA.

3. *Combining outlier scores of individual subsequences.* The techniques that assign a label (outlier or normal) to the subsequences typically look for any one outlier subsequence in the entire test sequence [Ghosh et al. 1999a; Michael and Ghosh 2000; Ghosh et al. 1999b; Kosoresow and Hofmeyr 1997] or count the number of outlier subsequences in the test sequence [Dasgupta and Nino 2000; Forrest et al. 1994; Marceau 2000].

Certain techniques that count the number of mismatches between a test subsequence and its closest match in a dictionary either use the test subsequence with maximum mismatches as the outlier score for the entire sequence [Hofmeyr et al. 1998] or use an aggregate (sum or average) of individual mismatch counts [Marceau 2000; Debar et al. 1998; Forrest et al. 1996; Lane and Brodley 1999].

Lee et al. [1997] adopt an interesting way to combine the labels that are assigned to each subsequence. The authors obtain a sequence of n - w + 1 predicted labels (where n is the length of test sequence S and w is the length of the sliding window). They apply a sliding window of length $2w$ with a sliding step of w on this sequence. For each of the window, if the number of outlier labels are more than w, the entire window is labeled as outlier otherwise it is labeled as normal. The total number

of outlier windows is the outlier score for the test sequence. Similar approach has been adopted by Gao et al. [2002].

Another interesting method is used by Ghosh et al. [1999a] called the *leaky bucket*. They use the sequence of n - w + 1 predicted labels as before. For each outlier label 1 is added to the global outlier score and for each normal label 1 is subtracted (the score is never allowed to fall below 0). Thus consecutive outliers result in the global score to increase. If at any time the global score goes above a threshold, the entire test sequence is declared to be outlier. This approach is useful in cases where a sustained outlier behavior constitutes an anomaly while an isolated outlier is not considered interesting.

13.1.2 *Detecting outlier subsequences in a long sequence*

DEFINITION 3. *Detect short subsequences in a long sequence T, that are outlier with respect to rest of T.*

Typically in this category of techniques, a subsequence is a region within T consisting of consecutive observations or events. This problem formulation occurs in event and time-series data sets where the data is in the form of a long sequence and contains regions that are outliers. The techniques that address this problem, typically work in an unsupervised mode, due to the lack of any training data. The underlying assumption is that the normal behavior of the time-series follows a defined pattern. A subsequence within T which does not conform to this pattern is an outlier.

Key challenges faced by techniques which follow Definition 2 are

- The length of the outlying subsequence to be detected is not generally defined. A long sequence could contain outlying regions of variable lengths. Thus fixed length segmenting of the sequence is often not useful.
- Since the input sequence T contains outlying regions, it becomes challenging to create a robust model of normalcy for T.

Chakrabarti et al. [1998] propose a surprise detection technique in market basket transactions. The data is a sequence of itemsets, T (can be treated as a binary vector) ordered by time. The authors propose to segment T such that the sum of number of bits require to encode each segment (using Shannon's classical Information Theorem) is minimized. The authors show that a optimal $O(|T|^2)$ solution exists to find such segmentation. The segments which require highest number of bits for encoding are treated as outliers.

Keogh et al. [2004] propose an algorithm called *Window Comparison Anomaly Detection* (WCAD), where the authors extract subsequences out of a given sequence of continuous observations using a sliding window. The authors compare each subsequence with the entire sequence using a compression based dissimilarity measure. The outlier score of each subsequence is its dissimilarity with the entire sequence.

Keogh et al [2005; 2006] propose a related technique (HOT SAX) to solve the above problem for continuous time-series. The basic approach followed by the authors is to extract subsequences out of the given sequence using sliding window, and then computing the distance of each subsequence to its closest non-overlapping subsequence within the original sequence. The outlier score of a subsequence is

proportional to its distance from its nearest neighbors. Distance between two sequences is measured using *Euclidean* measure. Similar approach is also applied to the domain of medical data by Lin et al. [2005]. The same authors propose the use of *Haar Wavelet* based transformation to make the previous technique more efficient [Fu et al. 2006; Bu et al. 2007].

Maximum Entropy Markov Models (Maxent) [McCallum et al. 2000; Pavlov and Pennock 2002; Pavlov 2003] as well as *Conditional Random Fields* (CRF) [Lafferty et al. 2001], have been used for segmenting text data. The problem formulation there is to predict the most likely state sequence for a given observation sequence. Any outlying segment within the observation sequence will have a low conditional probability for any state sequence.

13.1.3 *Determining if the frequency of a query pattern in a given sequence is outlier w.r.t its expected frequency.* Such formulation of the outlier detection problem is motivated from the *case vs control* type of data [Helman and Bhangoo 1997]. The idea is to detect patterns whose occurrence in a given test data set (case) is different from its occurrence in a normal data set (control). For sequences this can be exactly defined as

DEFINITION 4. *Given a short query pattern s, a long test sequence S and a set of long sequences $\Im$ ($|\Im| \geq 1$), determine if the frequency of occurrence of s in S is outlier with respect to frequency of occurrence of s in $\Im$.*

The general solution that the authors provide for this problem is to model the expected frequency of s using $\Im$. The observed frequency of occurrence of s in S is compared to the expected frequency to determine if s is an outlier or not. The occurrence of s can be defined as a substring or episode or a permutation, as discussed in the previous section.

Gwadera et al. [2005b] discuss the case where s can occur as a subsequence within S (and $T \in \Im$). They also propose a similar technique [Gwadera et al. 2004] where they consider the occurrence of any permutation of s within S (and $T \in \Im$).

A key observation to make here is that s can be outlier if its observed frequency in S is very low with respect to the expected frequency (under-represented pattern) or if its observed frequency in S is very high with respect to the expected frequency (over-represented pattern).

The first paper assumes that the symbols in s are generated by a memoryless Bernoulli source. Thus the expected frequency of s in $T \in \Im$ is computed using the expected frequency of the individual symbols only. The same authors extend this technique to sequences assumed to be generated from a Markov Model [Gwadera et al. 2005a]. They use an *Interpolated Markov Model* (IMM) to estimate the expected frequency of s.

The application of the techniques solving the problem stated as Definition 4 makes sense when s and S exist. Thus these techniques are more suited to prioritize the patterns (s) existing in a known outlying sequence (S) based on their frequency in S. Keogh et al. [2002] extract substrings from a given string of alphabets, S using a sliding window. For each of these substrings they employ an approach similar to the Markov Model approach by Gwadera et al. [2005a] to determine if this substring is anomalous with respect to a normal database of strings, $\Im$. Instead of IMM, the

authors use *suffix trees* to estimate the expected frequency of a substring in the normal database of strings.

13.2 Handling Spatial Outliers

`Type III` outlier detection in spatial data involves finding subgraphs or subcomponents in the data that are anomalous. A limited amount of research has been done in this category so we will discuss them individually.

Hazel [2000] propose a technique to detect regions in an image that are outlying with respect to rest of the image. The proposed technique makes use of *Multivariate Gaussian Random Markov Fields* (MGMRF) to segment a given image. The authors make an assumption that each pixel belonging to an outlying region of the image is also a `Type II` outlier within its segment. These pixels are detected as `Type II` outliers with respect to the segments (by estimating the conditional probability of each pixel), and then connected using a spatial structure available, to find the `Type III` outliers.

Outlier detection for graphs has been explored in application domains where the data can be modeled as graphs. Noble and Cook [2003] address two distinct `Type III` outlier detection problems for graph data. The first problem involves detecting outlying subgraphs in a given large graph. The authors use a bottom-up subgraph enumeration technique and analyze the frequency of a subgraph in the given graph to determine if it is an outlier or not. The size of the sub-graph is also taken into account, since a large sub-graph (such as the graph itself) is bound to occur very rarely in the graph while a small sub-graph (such as an individual node) will be more frequent. The second problem involves detecting if a given sub-graph is an outlier with respect to a large graph. The authors measure the regularity or entropy of the sub-graph in the context of the entire graph to determine its outlier score.

14. RELATIONSHIP BETWEEN `TYPE I`, `TYPE II` AND `TYPE III` OUTLIER DETECTION TECHNIQUES

It should be noted that `Type II` outlier detection is orthogonal to `Type I` and `Type III` outlier detection problem. `Type II` outlier detection means that the behavior of any pattern is analyzed within a context (determined by a set of *contextual* attributes). In principle, a `Type II` outlier detection approach can be applied in a `Type III` setting also. If the contextual attributes are ignored, a `Type II` outlier detection technique reduces to a `Type I` or a `Type III` outlier detection technique.

Choice between `Type I` and `Type III` outlier detection technique is determined by the nature of outliers and the type of data that is being dealt with. But the choice of applying a `Type II` outlier detection technique is determined by the meaningfulness of the `Type II` outliers in the target application domain. Another key factor is the availability of *contextual* attributes. In several cases defining a context is straightforward, and hence applying a `Type II` outlier detection technique makes sense. Several times, defining a context is not easy and hence such techniques cannot be applied.

15. EVALUATION OF OUTLIER DETECTION TECHNIQUES

Evaluation of an outlier detection technique is very important to establish its usefulness in detecting outliers in a given data set. In previous sections we saw that several techniques require a number of parameters that need to be determined empirically. An evaluation metric is required in such cases to determine the best values for the involved parameters. As there is not unique formulation of the outlier detection problem, there has not been a unique evaluation strategy adopted by the different techniques discussed in the survey. In this section we discuss the different evaluation strategies adopted to evaluate an outlier detection technique. We have divided the discussion in two parts – the first part deals with detecting outliers in a given data set, and the second part deals with detecting outliers in an application domain.

15.1 Detecting outliers in a given data set

The objective of any outlier detection technique is to detect outliers in a data set. Evaluating a technique for this objective involves running the technique on a labeled validation set and measuring how well the outliers are detected. Different measures have been used by different techniques. Here we discuss the different evaluation methodologies adopted to measure the capability of a technique to detect outliers in a given data set.

A labeling type of outlier detection technique is typically evaluated using any of the evaluation techniques from 2-class classification literature [Duda et al. 2000]. First of all a benchmark data set is chosen. The primary requirement is that the outliers should be labeled in the data set. For techniques that involve training, the evaluation data is split into training and testing data sets (*using techniques such as hold-out, cross-validation, jack-knife estimation*). The outlier detection technique is then applied to the *test* part of the validation data and the instances are labeled as outliers or normal. The predicted labels are compared with the actual labels to construct a confusion matrix as shown in Table 10(this table coincide exactly with confusion matrix constructed from classification output). An evaluation metric

		Actual	
		Outliers	Normal Instances
Predicted	Outliers	O_t	O_f
	Normal Instances	N_f	N_t

Table 10. A confusion matrix generated after running an outlier detection technique on validation data

is constructed using the above quantities, represented by $f(O_t, O_f, N_f, N_t, \Theta, C)$. Here Θ represents the parameters associated with the outlier detection technique and C is a cost matrix that assigns weights of each of the four quantities.

Various evaluation metrics [Mitchell 1997] have been applied in outlier detection literature, such as precision, recall, accuracy, false positive rate, false negative rate, detection rates, ROC-curve etc.

For scoring type of outlier detection techniques, there is typically a threshold parameter to determine the cut-off above which the instances are treated as outliers.

The threshold parameter is either determined using the outlier scores or left for the users to choose. After applying this cut-off, the above confusion matrix is constructed and the evaluation metric is computed for the given value of threshold (incorporated in Θ).

The choice of an evaluation data set depends on what type of data is the outlier detection technique targeted for. Parametric statistical techniques are typically evaluated on artificially generated data sets from known distributions [Abraham and Chuang 1989; Abraham and Box 1979]. Outliers are artificially injected in the data ensuring that they do not belong to the distribution(s) from which rest of the samples are generated.

In the data mining community, the UCI KDD archive [Bay et al. 2000] is a valuable repository of benchmark data sets for evaluating different data mining algorithms. Validation data sets for outlier detection are not included in the repository, but several authors have adapted some of the available data sets to be used for outlier detection. The data sets containing rare class are chosen for this purpose and the instances belonging to the rare class are treated as outliers in the data. Such analysis is done by Aggarwal and Yu [2001]. Hawkins et al. [1984] provide a data set known as *HBK* data set that has been used by statistical techniques for evaluation. Another technique is to take a labeled data set and remove instances of any one class. This reduced set forms the normal data. All or few instances from the removed class are injected in the normal data as outliers. This evaluation methodology is used by Lazarevic et al. [2003] to evaluate different outlier techniques.

Most of the available benchmark data sets are adapted to be used for evaluating `Type I` outlier detection techniques. No data sets are available to evaluate `Type II` outlier detection techniques, though Song et al. [2007] have adapted the UCI KDD data sets for this purpose. For `Type III` outlier detection techniques there are no publicly available benchmark data sets, that can be used to evaluate any such technique. Data sets focused on application domains have been used to evaluate such techniques and are discussed in the next subsection.

Several outlier detection techniques are evaluated for other objective functions such as scalability, ability to work with higher dimensional data sets, ability to handle noise. Same metrics as discussed above are used for such evaluation. The only difference is the choice of data sets that can capture the complexity being evaluated.

15.2 Evaluation in application domain

Outlier detection is a highly application domain oriented concept. As mentioned earlier, there is no universal technique that can be applied in every possible setting. We discussed in Section 3 how different outlier detection techniques have been developed for various application domains. Such techniques need to be evaluated not only for how well they detect outliers in a data set (as discussed in previous subsection), but also for how well they perform in the target application domain. The objective here is to detect outliers from the application domain perspective.

Such evaluation is done by choosing a validation data set that represents a sample belonging to the target application domain. The validation data should have labeled entities that are considered to be outliers in the domain. The evaluation strategies

mentioned in the previous subsection can be applied here too.

A key observation here is that such evaluation measures the performance of the entire outlier detection setting (including the technique, the features chosen and other related parameters/assumptions). It could be possible that an outlier detection technique that performs well based on the evaluation as discussed in previous subsection, might not perform as well when evaluated in the application domain. But such evaluation is necessary to establish the usefulness of the technique in the target application domain.

Some of the popular application domains have benchmark data sets that are listed in Table 11.

Application Domain	**Benchmark Data Set**
Network Intrusion Detection	MIT Lincoln Lab 1998 Data set [Lippmann et al. 2000] and 1999 KDDCup Data (*Available in the UCI KDD archive [Bay et al. 2000]*)
System Call Intrusion Detection	Forrest et al [Forrest et al. 1996]
Mobile Phone Fraud Detection	VoiceTone Data [Douglas et al. 2004]
Credit Card Fraud Detection	Transaction Generator [Aleskerov et al. 1997]
Novel Topic Detection	TDT corpus (`http://www.ldc.upenn.edu`)

Table 11. A list of benchmark data sets used for evaluating outlier detection techniques in application domains

Such benchmark data sets allow a standardized comparative evaluation of outlier detection techniques and hence are very useful. But often times the lack of such benchmark data sets have forced researchers to evaluate their techniques on proprietary or confidential data sets. Such data sets are not available publicly. Another challenge with evaluation from application domain perspective is that labeled validation data is often not available at all. For example, techniques dealing with aircraft fault detection do not have readily available labeled validation data. In such cases a qualitative analysis is performed, which typically involves a domain expert.

16. DISCUSSIONS AND CONCLUSIONS

Outlier detection is an extremely important problem with direct application in a wide variety of domains. A key observation with outlier detection is that it is not a well-formulated problem. We have discussed the different ways in which the problem has been formulated in literature. Every unique problem formulation entails a different approach, resulting in a huge literature on outlier detection techniques. Several techniques have been proposed to target a particular application domain. The survey can hopefully allow mapping such existing techniques to other application domains.

The concept of using a context to detect `Type II` outliers has not been completely understood. Several techniques unknowingly have adopted a `Type II` outlier detection approach. Song et al. [2007] have shown that using a context improves the outlier detection capability of a technique.

REFERENCES

Abe, N., Zadrozny, B., and Langford, J. 2006. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, New York, NY, USA, 504–509.

Abraham, B. and Box, G. E. P. 1979. Bayesian analysis of some outlier problems in time series. *Biometrika 66,* 2, 229–236.

Abraham, B. and Chuang, A. 1989. Outlier detection and time series modeling. *Technometrics 31,* 2, 241–248.

Addison, J., Wermter, S., and MacIntyre, J. 1999. Effectiveness of feature extraction in neural network architectures for novelty detection. In *Proceedings of the 9th International Conference on Artificial Neural Networks*. Vol. 2. 976–981.

Aeyels, D. 1991. On the dynamic behaviour of the novelty detector and the novelty filter. In *Analysis of Controlled Dynamical Systems- Progress in Systems and Control Theory*, B. Bonnard, B. Bride, J. Gauthier, and I. Kupka, Eds. Vol. 8. Springer, Berlin, 1–10.

Agarwal, D. 2005. An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In *Proceedings of the 5th IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 26–33.

Agarwal, D. 2006. Detecting anomalies in cross-classified streams: a bayesian approach. *Knowledge and Information Systems 11,* 1, 29–44.

Aggarwal, C. 2005. On abnormality detection in spuriously populated data streams. In *Proceedings of 5th SIAM Data Mining*. 80–91.

Aggarwal, C. and Yu, P. 2001. Outlier detection for high dimensional data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM Press, 37–46.

Agrawal, R., Imieliski, T., and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM Press, 207–216.

Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering*. IEEE Computer Society, Washington, DC, USA, 3–14.

Albrecht, S., Busch, J., Kloppenburg, M., Metze, F., and Tavan, P. 2000. Generalized radial basis function networks for classification and novelty detection: self-organization of optional bayesian decision. *Neural Networks 13,* 10, 1075–1093.

Aleskerov, E., Freisleben, B., and Rao, B. 1997. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of IEEE Computational Intelligence for Financial Engineering*. 220–226.

Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. 1998. Topic detection and tracking pilot study. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*. 194–218.

Anderson, Lunt, Javitz, Tamaru, A., and Valdes, A. 1995. Detecting unusual program behavior using the statistical components of NIDES. Tech. Rep. SRI–CSL–95–06, Computer Science Laboratory, SRI International. may.

Anderson, D., Frivold, T., Tamaru, A., and Valdes, A. 1994. Next-generation intrusion detection expert system (nides), software users manual, beta-update release. Tech. Rep. SRI–CSL–95–07, Computer Science Laboratory, SRI International. May.

Angiulli, F. and Pizzuti, C. 2002. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*. Springer-Verlag, 15–26.

Anscombe, F. J. and Guttman, I. 1960. Rejection of outliers. *Technometrics 2,* 2, 123–147.

Arning, A., Agrawal, R., and Raghavan, P. 1996. A linear method for deviation detection in large databases. In *Proceedings of 2nd International Conference of Knowledge Discovery and Data Mining*. 164–169.

Augusteijn, M. and Folkert, B. 2002. Neural network classification and novelty detection. *International Journal on Remote Sensing 23,* 14, 2891–2902.

Baker, D., Hofmann, T., McCallum, A., and Yang, Y. 1999. A hierarchical probabilistic model for novelty detection in text. In *Proceedings of International Conference on Machine Learning*.

Barbara, D., Couto, J., Jajodia, S., and Wu, N. 2001a. Adam: a testbed for exploring the use of data mining in intrusion detection. *SIGMOD Rec. 30,* 4, 15–24.

Barbara, D., Couto, J., Jajodia, S., and Wu, N. 2001b. Detecting novel network intrusions using bayes estimators. In *Proceedings of the First SIAM International Conference on Data Mining*.

Barbara, D., Li, Y., and Couto, J. 2002. Coolcat: an entropy-based algorithm for categorical clustering. In *Proceedings of the eleventh international conference on Information and knowledge management*. ACM Press, 582–589.

Barbara, D., Li, Y., Couto, J., Lin, J.-L., and Jajodia, S. 2003. Bootstrapping a data mining intrusion detection system. In *Proceedings of the 2003 ACM symposium on Applied computing*. ACM Press, 421–425.

Barnett, V. 1976. The ordering of multivariate data (with discussion). *Journal of the Royal Statistical Society. Series A 139*, 318–354.

Barnett, V. and Lewis, T. 1994. *Outliers in statistical data*. John Wiley and sons.

Barson, P., Davey, N., Field, S. D. H., Frank, R. J., and McAskie, G. 1996. The detection of fraud in mobile phone networks. *Neural Network World 6,* 4.

Bartkowiak, A. and Szustalewicz, A. 1997. The grand tour as a method for detecting multivariate outliers. *Machine Graphics and Vision 6*, 487–505.

Basu, S., Bilenko, M., and Mooney, R. J. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 59–68.

Basu, S. and Meckesheimer, M. 2007. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems 11,* 2 (February), 137–154.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. 1970. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. In *Annals of Mathematical Statistics*. Vol. 41(1). 164–171.

Bay, S. D., Kibler, D. F., Pazzani, M. J., and Smyth, P. 2000. The UCI KDD archive of large data sets for data mining research and experimentation. *SIGKDD Explorations 2,* 2, 81–85.

Bay, S. D. and Schwabacher, M. 2003. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 29–38.

Beale, R. and Jackson, T. 1990. *Neural computing: an introduction*. IOP Publishing Ltd., Bristol, UK.

Beckman, R. J. and Cook, R. D. 1983. Outlier...s. *Technometrics 25,* 2, 119–149.

Bianco, A. M., Ben, M. G., Martinez, E. J., and Yohai, V. J. 2001. Outlier detection in regression models with arima errors using robust estimates. *Journal of Forecasting 20,* 8, 565–579.

Bishop, C. 1994. Novelty detection and neural network validation. In *Proceedings of IEEE Vision, Image and Signal Processing*. Vol. 141. 217–222.

Blender, R., Fraedrich, K., and Lunkeit, F. 1997. Identification of cyclone-track regimes in the north atlantic. *Quarterly Journal of the Royal Meteorological Society 123,* 539, 727–741.

Bolton, R. and Hand, D. 1999. Unsupervised profiling methods for fraud detection. In *Credit Scoring and Credit Control VII*.

Borisyuk, R., Denham, M., Hoppensteadt, F., Kazanovich, Y., and Vinogradova, O. 2000. An oscillatory neural network model of sparse distributed memory and novelty detection. *Biosystems 58*, 265–272.

Box, G. E. P. and Tiao, G. C. 1968. Bayesian analysis of some outlier problems. *Biometrika 55,* 1, 119–129.

Brause, R., Langsdorf, T., and Hepp, M. 1999. Neural data mining for credit card fraud detection. In *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*. 103–106.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. 1999. Optics-of: Identifying local outliers. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*. Springer-Verlag, 262–270.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. 2000. Lof: identifying density-based local outliers. In *Proceedings of 2000 ACM SIGMOD International Conference on Management of Data*. ACM Press, 93–104.

Brockett, P. L., Xia, X., and Derrig, R. A. 1998. Using kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *Journal of Risk and Insurance 65,* 2 (June), 245–274.

Brodley, C. E. and Friedl, M. A. 1996. Identifying and eliminating mislabeled training instances. In *Proceedings of the 30th National Conference on Artificial Intelligence*. AAI Press, Portland OR, 799–805.

Brotherton, T. and Johnson, T. 2001. Anomaly detection for advance military aircraft using neural networks. In *Proceedings of 2001 IEEE Aerospace Conference*.

Brotherton, T., Johnson, T., and Chadderdon, G. 1998. Classification and novelty detection using linear models and a class dependent– elliptical basis function neural network. In *Proceedings of the IJCNN Conference*. Anchorage AL.

Bu, Y., Leung, T.-W., Fu, A., Keogh, E., Pei, J., and Meshkin, S. 2007. Wat: Finding top-k discords in time series database. In *Proceedings of 7th SIAM International Conference on Data Mining*.

Byers, S. D. and Raftery, A. E. 1998. Nearest neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association 93*, 577–584.

Byungho, H. and Sungzoon, C. 1999. Characteristics of autoassociative mlp as a novelty detector. In *Proceedings of IEEE International Joint Conference on Neural Networks*. Vol. 5. 3086–3091.

Cabrera, J. B. D., Lewis, L., and Mehra, R. K. 2001. Detection and classification of intrusions and faults using sequences of system calls. *SIGMOD Records 30,* 4, 25–34.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S. 2000. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 280–284.

Campbell, C. and Bennett, K. 2001. A linear programming approach to novelty detection. In *Proceedings of Advances in Neural Information Processing*. Vol. 14. Cambridge Press.

Carpenter, G. A. and Grossberg, S. 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision Graphical Image Processing 37,* 1, 54–115.

Caudell, T. and Newman, D. 1993. An adaptive resonance architecture to define normality and detect novelties in time series and databases. In *IEEE World Congress on Neural Networks*. IEEE, Portland, OR, 166–176.

Chakrabarti, S., Sarawagi, S., and Dom, B. 1998. Mining surprising patterns using temporal description length. In *Proceedings of the 24rd International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 606–617.

Chan, P. K. and Mahoney, M. V. 2005. Modeling multiple time series for anomaly detection. In *Proceedings of the Fifth IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 90–97.

Chaudhary, A., Szalay, A. S., and Moore, A. W. 2002. Very fast outlier detection in large multidimensional data sets. In *Proceedings of ACM SIGMOD Workshop in Research Issues in Data Mining and Knowledge Discovery (DMKD)*. ACM Press.

Chen, D., Shao, X., Hu, B., and Su, Q. 2005. Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra. *Analytical Sciences 21,* 2, 161–167.

Chiu, A. and chee Fu, A. W. 2003. Enhancements on local outlier detection. In *Proceedings of 7th International Database Engineering and Applications Symposium*. 298–307.

Chow, C. 1970. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory 16,* 1, 41–46.

Chow, C. and Yeung, D.-Y. 2002. Parzen-window network intrusion detectors. In *Proceedings of the 16th International Conference on Pattern Recognition*. Vol. 4. IEEE Computer Society, Washington, DC, USA, 40385.

Cohen, W. W. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, A. Prieditis and S. Russell, Eds. Morgan Kaufmann, Tahoe City, CA, 115–123.

Cox, K. C., Eick, S. G., Wills, G. J., and Brachman, R. J. 1997. Visual data mining: Recognizing telephone calling fraud. *Journal of Data Mining and Knowledge Discovery 1,* 2, 225–231.

Crook, P. and Hayes, G. 2001. A robot implementation of a biologically inspired method for novelty detection. In *Proceedings of Towards Intelligent Mobile Robots Conference*. Manchester, UK.

Crook, P. A., Marsland, S., Hayes, G., and Nehmzow, U. 2002. A tale of two filters - on-line novelty detection. In *Proceedings of International Conference on Robotics and Automation*. 3894–3899.

Cun, Y. L., Boser, B., Denker, J. S., Howard, R. E., Habbard, W., Jackel, L. D., and Henderson, D. 1990. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 396–404.

Dasgupta, D. and Majumdar, N. 2002. Anomaly detection in multidimensional data using negative selection algorithm. In *Proceedings of the IEEE Conference on Evolutionary Computation*. Hawaii, 1039–1044.

Dasgupta, D. and Nino, F. 2000. A comparison of negative and positive selection algorithms in novel pattern detection. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 1. Nashville, TN, 125–130.

Davy, M. and Godsill, S. 2002. Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. Orlando, USA.

Debar, H., Dacier, M., Nassehi, M., and Wespi, A. 1998. Fixed vs. variable-length patterns for detecting suspicious process behavior. In *Proceedings of the 5th European Symposium on Research in Computer Security*. Springer-Verlag, London, UK, 1–15.

Denning, D. E. 1987. An intrusion detection model. *IEEE Transactions of Software Engineering 13,* 2, 222–232.

Desforges, M., Jacob, P., and Cooper, J. 1998. Applications of probability density estimation to the detection of abnormal conditions in engineering. In *Proceedings of Institute of Mechanical Engineers*. Vol. 212. 687–703.

Diaz, I. and Hollmen, J. 2002. Residual generation and visualization for understanding novel process conditions. In *Proceedings of IEEE International Joint Conference on Neural Networks*. IEEE, Honolulu, HI, 2070–2075.

Diehl, C. and II, J. H. 2002. Real-time object classification and novelty detection for collaborative video surveillance. In *Proceedings of IEEE International Joint Conference on Neural Networks*. IEEE, Honolulu, HI.

Donoho, S. 2004. Early detection of insider trading in option markets. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 420–429.

Dorronsoro, J. R., Ginel, F., Sanchez, C., and Cruz, C. S. 1997. Neural fraud detection in credit card operations. *IEEE Transactions On Neural Networks 8,* 4 (July), 827–834.

Douglas, S., Agarwal, D., Alonso, T., Bell, R., Rahim, M., Swayne, D. F., and Volinsky, C. 2004. Mining customer care dialogs for "daily news". In *Proceedings of 8th International Conference on Spoken Language Processing*.

Duda, R. O., Hart, P. E., and Stork, D. G. 2000. *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Dutta, H., Giannella, C., Borne, K., and Kargupta, H. 2007. Distributed top-k outlier detection in astronomy catalogs using the demac system. In *Proceedings of 7th SIAM International Conference on Data Mining*.

E. Eskin, W. L. and Stolfo, S. 2001. Modeling system call for intrusion detection using dynamic window sizes. In *Proceedings of DARPA Information Survivability Conference and Exposition*.

Edgeworth, F. Y. 1887. On discordant observations. *Philosophical Magazine 23,* 5, 364–375.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings National Academy of Science USA 95,* 25, 14863–14868.

Emamian, V., Kaveh, M., and Tewfik, A. 2000. Robust clustering of acoustic emission signals using the kohonen network. In *Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing*. IEEE Computer Society.

Endler, D. 1998. Intrusion detection: Applying machine learning to solaris audit data. In *Proceedings of the 14th Annual Computer Security Applications Conference*. IEEE Computer Society, 268.

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P.-N., Kumar, V., Srivastava, J., and Dokas, P. 2004. MINDS - Minnesota Intrusion Detection System. In *Data Mining - Next Generation Challenges and Future Directions*. MIT Press.

Eskin, E. 2000. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 255–262.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. 2002. A geometric framework for unsupervised anomaly detection. In *Proceedings of Applications of Data Mining in Computer Security*. Kluwer Academics, 78–100.

Eskin, E., Grundy, W. N., and Singer, Y. 2000. Protein family classification using sparse markov transducers. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 134–145.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, E. Simoudis, J. Han, and U. Fayyad, Eds. AAAI Press, Portland, Oregon, 226–231.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. 1994. Fast subsequence matching in time-series databases. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*. ACM Press, New York, NY, USA, 419–429.

Fan, W., Miller, M., Stolfo, S. J., Lee, W., and Chan, P. K. 2001. Using artificial anomalies to detect unknown and known network intrusions. In *Proceedings of the 2001 IEEE International Conference on Data Mining*. IEEE Computer Society, 123–130.

Fawcett, T. and Provost, F. 1999. Activity monitoring: noticing interesting changes in behavior. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 53–62.

Forrest, S., D'haeseleer, P., and Helman, P. 1996. An immunological approach to change detection: Algorithms, analysis and implications. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 110.

Forrest, S., Esponda, F., and Helman, P. 2004. A formal framework for positive and negative detection schemes. In *IEEE Transactions on Systems, Man and Cybernetics, Part B*. IEEE, 357–373.

Forrest, S., Hofmeyr, S. A., Somayaji, A., and Longstaff, T. A. 1996. A sense of self for unix processes. In *Proceedinges of the 1996 IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press, 120–128.

Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. 1994. Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*. IEEE Computer Society, Washington, DC, USA, 202.

Forrest, S., Warrender, C., and Pearlmutter, B. 1999. Detecting intrusions using system calls: Alternate data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*. IEEE Computer Society, Washington, DC, USA, 133–145.

Fox, A. J. 1972. Outliers in time series. *Journal of the Royal Statistical Society. Series B(Methodological) 34,* 3, 350–363.

Fu, A. W.-C., Leung, O. T.-W., Keogh, E. J., and Lin, J. 2006. Finding time series discords based on haar transform. In *Proceeding of the 2nd International Conference on Advanced Data Mining and Applications*. Springer Verlag, 31–41.

Fujimaki, R., Yairi, T., and Machida, K. 2005. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM Press, New York, NY, USA, 401–410.

Galeano, P., Pea, D., and Tsay, R. S. 2004. Outlier detection in multivariate time series via projection pursuit. Statistics and Econometrics Working Papers ws044211, Universidad Carlos III, Departamento de Estadstica y Econometra. Sep.

Gao, B., Ma, H.-Y., and Yang, Y.-H. 2002. Hmms (hidden markov models) based on anomaly intrusion detection method. In *Proceedings of International Conference on Machine Learning and Cybernetics*. IEEE Computer Society, 381–385.

Ghosh, A. K., Schwartzbard, A., and Schatz, M. 1999a. Learning program behavior profiles for intrusion detection. In *Proceedings of 1st USENIX Workshop on Intrusion Detection and Network Monitoring*. 51–62.

Ghosh, A. K., Schwartzbard, A., and Schatz, M. 1999b. Using program behavior profiles for intrusion detection. In *Proceedings of SANS Third Conference and Workshop on Intrusion Detection and Response*.

Ghosh, A. K., Wanken, J., and Charron, F. 1998. Detecting anomalous and unknown intrusions against programs. In *Proceedings of the 14th Annual Computer Security Applications Conference*. IEEE Computer Society, 259.

Ghosh, S. and Reilly, D. L. 1994. Credit card fraud detection with a neural-network. In *Proceedings of the 27th Annual Hawaii International Conference on System Science*. Vol. 3. Los Alamitos, CA.

Ghoting, A., Otey, M. E., and Parthasarathy, S. 2004. Loaded: Link-based outlier and anomaly detection in evolving data sets. In *Proceedings of the 4th International Conference on Data Mining*. 387–390.

Ghoting, A., Parthasarathy, S., and Otey, M. 2006. Fast mining of distance-based outliers in high dimensional datasets. In *Proceedings of the SIAM International Conference on Data Mining*.

Gibbons, R. D. 1994. *Statistical Methods for Groundwater Monitoring*. John Wiley & Sons, Inc.

Gonzalez, F. A. and Dasgupta, D. 2003. Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines 4,* 4, 383–403.

Grubbs, F. 1969. Procedures for detecting outlying observations in samples. *Technometrics 11,* 1, 1–21.

Gusfield, D. 1997a. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, New York, NY, USA.

Gusfield, D. 1997b. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, New York, NY, USA, Chapter 5, 89–93.

Guttormsson, S., II, R. M., and El-Sharkawi, M. 1999. Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion 14,* 1 (March).

Gwadera, R., Atallah, M. J., and Szpankowski, W. 2004. Detection of significant sets of episodes in event sequences. In *Proceedings of the Fourth IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 3–10.

Gwadera, R., Atallah, M. J., and Szpankowski, W. 2005a. Markov models for identification of significant episodes. In *Proceedings of 5th SIAM International Conference on Data Mining*.

Gwadera, R., Atallah, M. J., and Szpankowski, W. 2005b. Reliable detection of episodes in event sequences. *Knowledge and Information Systems 7,* 4, 415–437.

Hansen, L. K., Liisberg, C., and Salamon, P. 1997. The error-reject tradeoff. *Open Systems and Information Dynamics 4,* 2, 159–184.

Harris, T. 1993. Neural network in machine health monitoring. *Professional Engineering*.

Haslett, J., Brandley, R., Craig, P., Unwin, A., and Wills, G. 1991. Dynamic graphics for exploring spatial data with application to locating global and local anomalies. *The American Statistician 45,* 3, 234–242.

Hautamaki, V., Karkkainen, I., and Franti, P. 2004. Outlier detection using k-nearest neighbour graph. In *Proceedings of 17th International Conference on Pattern Recognition*. Vol. 3. IEEE Computer Society, Washington, DC, USA, 430–433.

Hawkins, D. 1980. Identification of outliers. *Monographs on Applied Probability and Statistics*.

Hawkins, D. M., Bradu, D., and Kass, G. V. 1984. Location of several outliers in multiple-regression data using elemental sets. *Technometrics 26,* 3 (August), 197–208.

Hawkins, S., He, H., Williams, G. J., and Baxter, R. A. 2002. Outlier detection using replicator neural networks. In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*. Springer-Verlag, 170–180.

Hazel, G. G. 2000. Multivariate gaussian mrf for multispectral scene segmentation and anomaly detection. *GeoRS 38,* 3 (May), 1199–1211.

He, H., Wang, J., Graco, W., and Hawkins, S. 1997. Application of neural networks to detection of medical fraud. *Expert Systems with Applications 13,* 4, 329–336.

He, Z., Deng, S., and Xu, X. 2002. Outlier detection integrating semantic knowledge. In *Proceedings of the Third International Conference on Advances in Web-Age Information Management*. Springer-Verlag, London, UK, 126–131.

He, Z., Deng, S., Xu, X., and Huang, J. Z. 2006. A fast greedy algorithm for outlier mining. In *Proceedings of 10th Pacific-Asia Conference on Knowledge and Data Discovery*. 567–576.

He, Z., Xu, X., and Deng, S. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters 24,* 9-10, 1641–1650.

He, Z., Xu, X., and Deng, S. 2005. An optimization model for outlier detection in categorical data. In *Proceedings of International Conference on Intelligent Computing*. Vol. 3644. Springer.

Helman, P. and Bhangoo, J. 1997. A statistically based system for prioritizing information exploration under uncertainty. In *IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 27. IEEE, 449–466.

Helmer, G., Wong, J., Honavar, V., and Miller, L. 1998. Intelligent agents for intrusion detection. In *Proceedings of IEEE Information Technology Conference*. 121–124.

Hickinbotham, S. J. and Austin, J. 2000a. Novelty detection in airframe strain data. In *Proceedings of 15th International Conference on Pattern Recognition*. Vol. 2. 536–539.

Hickinbotham, S. J. and Austin, J. 2000b. Novelty detection in airframe strain data. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Vol. 6. 24–27.

Ho, L. L., Macey, C. J., and Hiller, R. 1999. A distributed and reliable platform for adaptive anomaly detection in ip networks. In *Proceedings of the 10th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*. Springer-Verlag, London, UK, 33–46.

Ho, T. V. and Rouat, J. 1997. A novelty detector using a network of integrate and fire neurons. *Lecture Notes in Computer Science 1327*, 103–108.

Ho, T. V. and Rouat, J. 1998. Novelty detection based on relaxation time of a network of integrate-and-fire neurons. In *Proceedings of Second IEEE World Congress on Computational Intelligence*. Anchorage, AK, 1524–1529.

Hodge, V. and Austin, J. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review 22,* 2, 85–126.

Hofmeyr, S. A., Forrest, S., and Somayaji, A. 1998. Intrusion detection using sequences of system calls. *Journal of Computer Security 6,* 3, 151–180.

Hollier, G. and Austin, J. 2002. Novelty detection for strain-gauge degradation using maximally correlated components. In *Proceedings of the European Symposium on Artificial Neural Networks*. 257–262–539.

Hollmen, J. and Tresp, V. 1999. Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*. MIT Press, Cambridge, MA, USA, 889–895.

Horn, P. S., Feng, L., Li, Y., and Pesce, A. J. 2001. Effect of outliers and nonhealthy individuals on reference interval estimation. *Clinical Chemistry 47,* 12, 2137–2145.

Hu, W., Liao, Y., and Vemuri, V. R. 2003. Robust anomaly detection using support vector machines. In *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282–289.

Ide, T. and Kashima, H. 2004. Eigenspace-based anomaly detection in computer systems. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 440–449.

Ihler, A., Hutchins, J., and Smyth, P. 2006. Adaptive event detection with time-varying poisson processes. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 207–216.

Ilgun, K., Kemmerer, R. A., and Porras, P. A. 1995. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering 21,* 3, 181–199.

Jagadish, H. V., Koudas, N., and Muthukrishnan, S. 1999. Mining deviants in a time series database. In *Proceedings of the 25th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., 102–113.

Jagota, A. 1991. Novelty detection on a very large number of memories stored in a hopfield-style network. In *Proceedings of the International Joint Conference on Neural Networks*. Vol. 2. Seattle, WA, 905.

Jain, A. K. and Dubes, R. C. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Inc.

Jakubek, S. and Strasser, T. 2002. Fault-diagnosis using neural networks with ellipsoidal basis functions. In *Proceedings of the American Control Conference*. Vol. 5. 3846–3851.

Japkowicz, N., Myers, C., and Gluck, M. A. 1995. A novelty detection approach to classification. In *Proceedings of International Joint Conference on Artificial Intelligence*. 518–523.

Javitz, H. S. and Valdes, A. 1991. The sri ides statistical anomaly detector. In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society.

Jian Tang, Zhixiang Chen, A. W.-c. F. and W.Cheung, D. 2002. Enhancing effectiveness of outlier detections for low density patterns. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 535–548.

Jiang, M. F., Tseng, S. S., and Su, C. M. 2001. Two-phase clustering process for outliers detection. *Pattern Recognition Letters 22,* 6-7, 691–700.

Jin, W., Tung, A. K. H., and Han, J. 2001. Mining top-n local outliers in large databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 293–298.

John, G. H. 1995. Robust decision trees: Removing outliers from databases. In *Proceeding of Knowledge Discovery and Data Mining*. 174–179.

Jordan, M. I. and Jacobs, R. A. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural Computing 6,* 2, 181–214.

Joshi, M. V., Agarwal, R. C., and Kumar, V. 2001. Mining needle in a haystack: classifying rare classes via two-phase rule induction. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. ACM Press, New York, NY, USA, 91–102.

Kadota, K., Tominaga, D., Akiyama, Y., and Takahashi, K. 2003. Detecting outlying samples in microarray data: A critical assessment of the effect of outliers on sample classification. *Chem-Bio Informatics 3,* 1, 30–45.

Karypis, G. and Kumar, V. 1998. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing 48,* 1, 96–129.

Keogh, E., Lin, J., and Fu, A. 2005. Hot sax: Efficiently finding the most unusual time series subsequence. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 226–233.

Keogh, E., Lin, J., Lee, S.-H., and Herle, H. V. 2006. Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems 11,* 1, 1–27.

Keogh, E., Lonardi, S., and chi' Chiu, B. Y. 2002. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 550–556.

Keogh, E., Lonardi, S., and Ratanamahatana, C. A. 2004. Towards parameter-free data mining. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 206–215.

Keogh, E. and Smyth, P. 1997. A probabilistic approach to fast pattern matching in time series databases. In *Proceedings of Third International Conference on Knowledge Discovery and Data Mining*, D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, Eds. AAAI Press, Menlo Park, California., Newport Beach, CA, USA, 24–30.

King, S., King, D., P. Anuzis, K. A., Tarassenko, L., Hayton, P., and Utete, S. 2002. The use of novelty detection techniques for monitoring high-integrity plant. In *Proceedings of the 2002 International Conference on Control Applications*. Vol. 1. Cancun, Mexico, 221–226.

Kitagawa, G. 1979. On the use of aic for the detection of outliers. *Technometrics 21,* 2 (may), 193–199.

Knorr, E. M. and Ng, R. T. 1997. A unified approach for mining outliers. In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 11.

Knorr, E. M. and Ng, R. T. 1998. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., 392–403.

Knorr, E. M. and Ng, R. T. 1999. Finding intensional knowledge of distance-based outliers. In *The VLDB Journal*. 211–222.

Knorr, E. M., Ng, R. T., and Tucakov, V. 2000. Distance-based outliers: algorithms and applications. *The VLDB Journal 8,* 3-4, 237–253.

Ko, H. and Jacyna, G. 2000. Dynamical behavior of autoassociative memory performing novelty filtering. In *IEEE Transactions on Neural Networks*. Vol. 11. 1152–1161.

Kohonen, T., Ed. 1997. *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Kojima, K. and Ito, K. 1999. Autonomous learning of novel patterns by utilizing chaotic dynamics. In *IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 1. IEEE, Tokyo, Japan, 284–289.

Korn, F., Labrinidis, A., Kotidis, Y., Faloutsos, C., Kaplunovich, A., and Perkovic, D. 1997. Quantifiable data mining using principal component analysis. Tech. Rep. CS-TR-3754, University of Maryland, College Park. February.

Kosoresow, A. P. and Hofmeyr, S. A. 1997. Intrusion detection via system call traces. *IEEE Software 14,* 5, 35–42.

Kou, Y., Lu, C.-T., and Chen, D. 2006. Spatial weighted outlier detection. In *Proceedings of SIAM Conference on Data Mining*.

Krogh, A., Brown, M., Mian, I., Sjolander, K., and Haussler, D. 1994. Hidden markov models in computational biology: application to protein modeling. In *Journal of Molecular Biology*. Vol. 235. 1501–1531.

Kruegel, C., Mutz, D., Robertson, W., and Valeur, F. 2003. Bayesian event classification for intrusion detection. In *Proceedings of the 19th Annual Computer Security Applications Conference*. IEEE Computer Society, 14.

Kruegel, C., Toth, T., and Kirda, E. 2002. Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM symposium on Applied computing*. ACM Press, 201–208.

Kruegel, C. and Vigna, G. 2003. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*. ACM Press, 251–261.

Labib, K. and Vemuri, R. 2002. Nsom: A real-time network-based intrusion detection using self-organizing maps. *Networks and Security*.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282–289.

Lakhina, A., Crovella, M., and Diot, C. 2005. Mining anomalies using traffic feature distributions. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, New York, NY, USA, 217–228.

Lane, T. and Brodley, C. E. 1997a. An application of machine learning to anomaly detection. In *Proceedings of 20th NIST-NCSC National Information Systems Security Conference*. 366–380.

Lane, T. and Brodley, C. E. 1997b. Sequence matching and learning in anomaly detection for computer security. In *Proceedings of AI Approaches to Fraud Detection and Risk Management*, Fawcett, Haimowitz, Provost, and Stolfo, Eds. AAAI Press, 43–49.

Lane, T. and Brodley, C. E. 1999. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information Systems and Security 2,* 3, 295–331.

Lauer, M. 2001. A mixture approach to novelty detection using training data with outliers. In *Proceedings of the 12th European Conference on Machine Learning*. Springer-Verlag, London, UK, 300–311.

Laurikkala, J., Juhola1, M., and Kentala., E. 2000. Informal identification of outliers in medical data. In *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*. 20–24.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., and Srivastava, J. 2003. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the Third SIAM International Conference on Data Mining*. SIAM.

Lee, W. and Stolfo, S. 1998. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*. San Antonio, TX.

Lee, W., Stolfo, S., and Chan, P. 1997. Learning patterns from unix process execution traces for intrusion detection. In *Proceedings of the AAAI 97 workshop on AI methods in Fraud and risk management*.

Lee, W., Stolfo, S. J., and Mok, K. W. 2000. Adaptive intrusion detection: A data mining approach. *Artificial Intelligence Review 14,* 6, 533–567.

Lee, W. and Xiang, D. 2001. Information-theoretic measures for anomaly detection. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 130.

Li, M. and Vitanyi, P. M. B. 1993. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin.

Li, X., Han, J., Kim, S., and Gonzalez, H. 2007. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *Proceedings of 7th SIAM International Conference on Data Mining*.

Li, Y., Pont, M. J., and Jones, N. B. 2002. Improving the performance of radial basis function classifiers in condition monitoring and fault diagnosis applications where unknown faults may occur. *Pattern Recognition Letters 23,* 5, 569–577.

Lin, J., Keogh, E., Fu, A., and Herle, H. V. 2005. Approximations to magic: Finding unusual medical time series. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*. IEEE Computer Society, Washington, DC, USA, 329–334.

Lin, S. and Brown, D. E. 2003. An outlier-based data association method for linking criminal incidents. In *Proceedings of 3rd SIAM Data Mining Conference*.

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. P., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K., and Zissman, M. A. 2000. Evaluating intrusion detection systems - the 1998 darpa off-line intrusion detection evaluation. In *DARPA Information Survivability Conference and Exposition (DISCEX) 2000*. Vol. 2. IEEE Computer Society Press, Los Alamitos, CA, 12–26.

Liu, J. P. and Weng, C. S. 1991. Detection of outlying data in bioavailability/bioequivalence studies. *Statistics Medicine 10,* 9, 1375–89.

Lu, C.-T., Chen, D., and Kou, Y. 2003. Algorithms for spatial outlier detection. In *Proceedings of 3rd International Conference on Data Mining*. 597–600.

Ma, J. and Perkins, S. 2003a. Online novelty detection on temporal sequences. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 613–618.

Ma, J. and Perkins, S. 2003b. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks*. Vol. 3. 1741– 1745.

MacDonald, J. W. and Ghosh, D. 2007. Copa–cancer outlier profile analysis. *Bioinformatics 22,* 23, 2950–2951.

Mahoney, M. V. and Chan, P. K. 2002. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 376–385.

Mahoney, M. V. and Chan, P. K. 2003. Learning rules for anomaly detection of hostile network traffic. In *Proceedings of the 3rd IEEE International Conference on Data Mining*. IEEE Computer Society, 601.

Mahoney, M. V., Chan, P. K., and Arshad, M. H. 2003. A machine learning approach to anomaly detection. Tech. Rep. CS–2003–06, Department of Computer Science, Florida Institute of Technology Melbourne FL 32901. march.

Manevitz, L. M. and Yousef, M. 2000. Learning from positive data for document classification using neural networks. In *Proceedings of Second Bar-Ilan Workshop on Knowledge Discovery and Learning*. Jerusalem.

Manevitz, L. M. and Yousef, M. 2002. One-class svms for document classification. *Journal of Machine Learning Research 2*, 139–154.

Manikopoulos, C. and Papavassiliou, S. 2002. Network intrusion and fault detection: a statistical anomaly approach. *IEEE Communication Magazine 40*.

Mannila, H., Toivonen, H., and Verkamo, A. I. 1997. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery 1,* 3, 259–289.

Manson, G. 2002. Identifying damage sensitive, environment insensitive features for damage detection. In *Proceedings of the IES Conference*. Swansea, UK.

Manson, G., Pierce, G., and Worden, K. 2001. On the long-term stability of normal condition for damage detection in a composite panel. In *Proceedings of the 4th International Conference on Damage Assessment of Structures*. Cardiff, UK.

Manson, G., Pierce, S. G., Worden, K., Monnier, T., Guy, P., and Atherton, K. 2000. Long-term stability of normal condition data for novelty detection. In *Proceedings of Smart Structures and Integrated Systems*. 323–334.

Marceau, C. 2000. Characterizing the behavior of a program using multiple-length n-grams. In *Proceedings of the 2000 workshop on New Security Paradigms*. ACM Press, New York, NY, USA, 101–110.

Marchette, D. 1999. A statistical method for profiling network traffic. In *Proceedings of 1st USENIX Workshop on Intrusion Detection and Network Monitoring*. Santa Clara, CA, 119–128.

Markou, M. and Singh, S. 2003a. Novelty detection: a review-part 1: statistical approaches. *Signal Processing 83,* 12, 2481–2497.

Markou, M. and Singh, S. 2003b. Novelty detection: a review-part 2: neural network based approaches. *Signal Processing 83,* 12, 2499–2521.

Marsland, S., Nehmzow, U., and Shapiro, J. 1999. A model of habituation applied to mobile robots. In *Proceedings of Towards Intelligent Mobile Robots*. Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-99-3-1.

Marsland, S., Nehmzow, U., and Shapiro, J. 2000a. Novelty detection for robot neotaxis. In *Proceedings of the 2nd International Symposium on Neural Compuatation*. 554 – 559.

Marsland, S., Nehmzow, U., and Shapiro, J. 2000b. A real-time novelty detector for a mobile robot. In *Proceedings of the EUREL Conference on Advanced Robotics Systems*.

Marsland, S., Nehmzow, U., and Shapiro, J. 2002. Environment-specific novelty detection. In *Proceedings of the 7th international conference on simulation of adaptive behavior on From animals to animats*. MIT Press, Cambridge, MA, USA, 36–45.

Martinelli, G. and Perfetti, R. 1994. Generalized cellular neural network for novelty detection. *IEEE Transactions on Circuits Systems I: Fundamental Theory Application 41,* 2, 187–190.

Martinez, D. 1998. Neural tree density estimation for novelty detection. *IEEE Transactions on Neural Networks 9,* 2, 330–338.

McCallum, A., Freitag, D., and Pereira, F. C. N. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 591–598.

McCallum, A., Nigam, K., and Ungar, L. H. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 169–178.

Michael, C. C. and Ghosh, A. 2000. Two state-based approaches to program-based anomaly detection. In *Proceedings of the 16th Annual Computer Security Applications Conference*. IEEE Computer Society, 21.

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill Higher Education.

Moya, M., Koch, M., and Hostetler, L. 1993. One-class classifier networks for target recognition applications. In *Proceedings on World Congress on Neural Networks, International Neural Network Society*. Portland, OR, 797–801.

Murray, A. F. 2001. Novelty detection using products of simple experts - a potential architecture for embedded systems. *Neural Networks 14,* 9, 1257–1264.

Nairac, A., Corbett-Clark, T., Ripley, R., Townsend, N., and Tarassenko, L. 1997. Choosing an appropriate model for novelty detection. In *Proceedings of the 5th IEEE International Conference on Artificial Neural Networks*. 227–232.

Nairac, A., Townsend, N., Carr, R., King, S., Cowley, P., and Tarassenko, L. 1999. A system for the analysis of jet engine vibration data. *Integrated Computer-Aided Engineering 6,* 1, 53–56.

Ng, R. T. and Han, J. 1994. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 144–155.

Noble, C. C. and Cook, D. J. 2003. Graph-based anomaly detection. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 631–636.

Odin, T. and Addison, D. 2000. Novelty detection using neural network technology. In *Proceedings of the COMADEN Conference*. Houston, TX.

Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S., and Panda, D. 2003. Towards nic-based intrusion detection. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 723–728.

Otey, M. E., Ghoting, A., and Parthasarathy, S. 2006. Fast distributed outlier detection in mixed-attribute data sets. *Data Min. Knowl. Discov. 12,* 2-3, 203–228.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B., and Faloutsos, C. 2002. Loci: Fast outlier detection using the local correlation integral. Tech. Rep. IRP-TR-02-09, Intel Research Laboratory, Pittsburgh, PA. July.

Parra, L., Deco, G., and Miesbach, S. 1996. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computing 8,* 2, 260–269.

Parzen, E. 1962. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics 33*, 1065–1076.

Pavlov, D. 2003. Sequence modeling with mixtures of conditional maximum entropy distributions. In *Proceedings of the Third IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 251.

Pavlov, D. and Pennock, D. 2002. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Proceedings of Advances in Neural Information Processing*. MIT Press.

Petrovskiy, M. I. 2003. Outlier detection algorithms in data mining systems. *Programming and Computer Software 29,* 4, 228–237.

Petsche, T., Marcantonio, A., Darken, C., Hanson, S., Kuhn, G., and Santoso, I. 1996. A neural network autoassociator for induction motor failure prediction. In *Proceedings of Advances in Neural Information Processing*. Vol. 8. 924–930.

Phoha, V. V. 2002. *The Springer Internet Security Dictionary*. Springer-Verlag.

Phua, C., Alahakoon, D., and Lee, V. 2004. Minority report in fraud detection: classification of skewed data. *SIGKDD Explorer Newsletter 6,* 1, 50–59.

Pires, A. and Santos-Pereira, C. 2005. Using clustering and robust estimators to detect outliers in multivariate data. In *Proceedings of International Conference on Robust Statistics*. Finland.

Pokrajac, D., Lazarevic, A., and Latecki, L. J. 2007. Incremental local outlier detection for data streams. In *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*.

Porras, P. A. and Neumann, P. G. 1997. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *Proceedings of 20th NIST-NCSC National Information Systems Security Conference*. 353–365.

Portnoy, L., Eskin, E., and Stolfo, S. 2001. Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM Workshop on Data Mining Applied to Security*.

Qiao, Y., Xin, X. W., Bin, Y., and Ge, S. 2002. Anomaly intrusion detection method based on hmm. *Electronics Letters 38,* 13, 663–664.

Qin, M. and Hwang, K. 2004. Frequent episode rules for internet anomaly detection. In *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications*. IEEE Computer Society.

Rabiner, L. R. and Juang, B. H. 1985. A probabilistic distance measure for hidden markov models. *ATT Technical Journal 64,* 2, 391–408.

Rabiner, L. R. and Juang, B. H. 1986. An introduction to hidden markov models. *IEEE ASSP Magazine 3,* 1, 4–16.

Rabiner, L. R., Lee, C. H., Juang, B. H., and Wilpon, J. G. 1989. Hmm clustering for connected word recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE Press, 405–408.

Ramadas, M., Ostermann, S., and Tjaden, B. C. 2003. Detecting anomalous network traffic with self-organizing maps. In *Proceedings of Recent Advances in Intrusion Detection*. 36–54.

Ramaswamy, S., Rastogi, R., and Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. ACM Press, 427–438.

Ratsch, G., Mika, S., Scholkopf, B., and Muller, K.-R. 2002. Constructing boosting algorithms from svms: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24,* 9, 1184–1199.

Ray, A. 2004. Symbolic dynamic analysis of complex systems for anomaly detection. *Signal Processing 84,* 7, 1115–1130.

Ren, D., Rahal, I., Perrizo, W., and Scott, K. 2004. A vertical distance-based outlier detection method with local pruning. In *Proceedings of the 13th ACM international conference on Information and knowledge management*. ACM Press, New York, NY, USA, 279–284.

Ren, D., Wang, B., and Perrizo, W. 2004. Rdf: A density-based outlier detection method using vertical data representation. In *Proceedings of 4th IEEE International Conference on Data Mining*. 503–506.

Ridgeway, G. 1997. Finite discrete markov process clustering. Tech. Rep. TR-97-24, Microsoft Research, Redmond, WA.

Roberts, S. 1999. Novelty detection using extreme value statistics. In *Proceedings of IEEE - Vision, Image and Signal processing*. Vol. 146. 124–129.

Roberts, S. 2002. Extreme value statistics for novelty detection in biomedical signal processing. In *Proceedings of the 1st International Conference on Advances in Medical Signal and Information Processing*. 166–172.

Roberts, S. and Penny, W. 1996. Novelty, confidence and errors in connectionist systems. In *Proceedings of IEEE Colloquium on Intelligent Sensors and Fault Detection*. Savoy Place, London, 261.

Roberts, S. and Tarassenko, L. 1994. A probabilistic resource allocating network for novelty detection. *Neural Computing 6,* 2, 270–284.

Rosner, B. 1983. Percentage points for a generalized esd many-outlier procedure. *Technometrics 25,* 2 (may), 165–172.

Rousseeuw, P. J. and Leroy, A. M. 1987. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA.

Ruotolo, R. and Surace, C. 1997. A statistical approach to damage detection through vibration monitoring. In *Proceedings of the 5th Pan American Congress of Applied Mechanics*. Puerto Rico.

Ruts, I. and Rousseeuw, P. J. 1996. Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis 23,* 1, 153–168.

Ryan, J., Lin, M.-J., and Miikkulainen, R. 1998. Intrusion detection with neural networks. In *Proceedings of Advances in Neural Information Processing Systems*. MIT Press, 10.

Saltenis, V. 2004. Outlier detection based on the distribution of distances between data points. *Informatica 15,* 3, 399–410.

Salvador, S. and Chan, P. 2003. Learning states and rules for time-series anomaly detection. Tech. Rep. CS–2003–05, Department of Computer Science, Florida Institute of Technology Melbourne FL 32901. march.

Sarawagi, S., Agrawal, R., and Megiddo, N. 1998. Discovery-driven exploration of olap data cubes. In *Proceedings of the 6th International Conference on Extending Database Technology*. Springer-Verlag, London, UK, 168–182.

Sargor, C. 1998. Statistical anomaly detection for link-state routing protocols. In *Proceedings of the Sixth International Conference on Network Protocols*. IEEE Computer Society, Washington, DC, USA, 62.

Saunders, R. and Gero, J. 2000. The importance of being emergent. In *Proceedings of Artificial Intelligence in Design*.

Scarth, G., McIntyre, M., Wowk, B., and Somorjai, R. 1995. Detection of novelty in functional images using fuzzy clustering. In *Proceedings of the 3rd Meeting of International Society for Magnetic Resonance in Medicine*. Nice, France, 238.

Scholkopf, B., Platt, O., Shawe-Taylor, J., Smola, A., and Williamson, R. 1999. Estimating the support of a high-dimensional distribution. Tech. Rep. 99-87, Microsoft Research.

Scott, S. L. 2001. Detecting network intrusion using a markov modulated nonhomogeneous poisson process. Submitted to the Journal of the American Statistical Association.

Sebyala, A. A., Olukemi, T., and Sacks, L. 2002. Active platform security through intrusion detection using naive bayesian network for anomaly detection. In *Proceedings of the 2002 London Communications Symposium*.

Sekar, R., Bendre, M., Dhurjati, D., and Bollineni, P. 2001. A fast automaton-based method for detecting anomalous program behaviors. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 144.

Sekar, R., Guang, Y., Verma, S., and Shanbhag, T. 1999. A high-performance network intrusion detection system. In *Proceedings of the 6th ACM conference on Computer and communications security*. ACM Press, 8–17.

Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H., and Zhou, S. 2002. Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM conference on Computer and communications security*. ACM Press, 265–274.

Sequeira, K. and Zaki, M. 2002. Admit: anomaly-based data mining for intrusions. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 386–395.

Sheikholeslami, G., Chatterjee, S., and Zhang, A. 1998. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24rd International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 428–439.

Shekhar, S., Lu, C.-T., and Zhang, P. 2001. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 371–376.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. 2003. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of 3rd IEEE International Conference on Data Mining*. 353–365.

Siaterlis, C. and Maglaris, B. 2004. Towards multisensor data fusion for dos detection. In *Proceedings of the 2004 ACM symposium on Applied computing*. ACM Press, 439–446.

Singh, S. and Markou, M. 2004. An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering 16,* 4, 396–407.

Skalak, D. B. and Rissland, E. L. 1990. Inductive learning in a mixed paradigm setting. In *Proceedings of National Conference of American Association for Artificial Intelligence*. 840–847.

Smith, R., Bivens, A., Embrechts, M., Palagiri, C., and Szymanski, B. 2002. Clustering approaches for anomaly based intrusion detection. In *Proceedings of Intelligent Engineering Systems through Artificial Neural Networks*. ASME Press, 579–584.

Smyth, P. 1994. Markov monitoring with unknown states. *IEEE Journal on Selected Areas in Communications, Special Issue on Intelligent Signal Processing for Communications 12,* 9 (december), 1600–1612.

Smyth, P. 1997. Clustering sequences with hidden markov models. In *Advances in Neural Information Processing*. Vol. 9. MIT Press.

Smyth, P. 1999. Probabilistic model-based clustering of multivariate and sequential data. In *Artificial Intelligence and Statistics*. Morgan Kaufman, San Mateo, CA, 299–304.

Snyder, D. 2001. Online intrusion detection using sequences of system calls. M.S. thesis, Department of Computer Science, Florida State University.

Sohn, H., Worden, K., and Farrar, C. 2001. Novelty detection under changing environmental conditions. In *Proceedings of Eighth Annual SPIE International Symposium on Smart Structures and Materials*. Newport Beach, CA.

Solberg, H. E. and Lahti, A. 2005. Detection of outliers in reference distributions: Performance of horn's algorithm. *Clinical Chemistry 51,* 12, 2326–2332.

Song, Q., Hu, W., and Xie, W. 2002. Robust support vector machine with bullet hole image classification. *IEEE Transactions on Systems, Man, and Cybernetics – Part C:Applications and Reviews 32,* 4.

Song, S., Shin, D., and Yoon, E. 2001. Analysis of novelty detection properties of autoassociators. In *Proceedings of Condition Monitoring and Diagnostic Engineering Management*. 577–584.

Song, X., Wu, M., Jermaine, C., and Ranka, S. 2007. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering 19,* 5, 631–645.

Spence, C., Parra, L., and Sajda, P. 2001. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*. IEEE Computer Society, Washington, DC, USA, 3.

Stefano, C., Sansone, C., and Vento, M. 2000. To reject or not to reject: that is the question–an answer in case of neural classifiers. *IEEE Transactions on Systems, Management and Cybernetics 30,* 1, 84–94.

Stefansky, W. 1972. Rejecting outliers in factorial designs. *Technometrics 14,* 2, 469–479.

Steinwart, I., Hush, D., and Scovel, C. 2005. A classification framework for anomaly detection. *Journal of Machine Learning Research 6*, 211–232.

Streifel, R., Maks, R., and El-Sharkawi, M. 1996. Detection of shorted-turns in the field of turbine-generator rotors using novelty detectors–development and field tests. *IEEE Transactions on Energy Conversations 11,* 2, 312–317.

Sun, H., Bao, Y., Zhao, F., Yu, G., and Wang, D. 2004. Cd-trees: An efficient index structure for outlier detection. 600–609.

Sun, J., Qu, H., Chakrabarti, D., and Faloutsos, C. 2005. Neighborhood formation and anomaly detection in bipartite graphs. In *Proceedings of the 5th IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 418–425.

Sun, J., Xie, Y., Zhang, H., and Faloutsos, C. 2007. Less is more: Compact matrix representation of large sparse graphs. In *Proceedings of 7th SIAM International Conference on Data Mining*.

Sun, P. and Chawla, S. 2004. On local spatial outliers. In *Proceedings of 4th IEEE International Conference on Data Mining*. 209–216.

Sun, P. and Chawla, S. 2006. Slom: a new measure for local spatial outliers. *Knowledge and Information Systems 9,* 4, 412–429.

Sun, P., Chawla, S., and Arunasalam, B. 2006. Mining for outliers in sequential databases. In *Proceedings of SIAM Conference on Data Mining*.

Surace, C. and Worden, K. 1998. A novelty detection method to diagnose damage in structures: an application to an offshore platform. In *Proceedings of Eighth International Conference of Off-shore and Polar Engineering*. Vol. 4. Colorado, USA, 64–70.

Surace, C., Worden, K., and Tomlinson, G. 1997. A novelty detection approach to diagnose damage in a cracked beam. In *Proceedings of SPIE*. Vol. 3089. 947–953.

Suzuki, E., Watanabe, T., Yokoi, H., and Takabayashi, K. 2003. Detecting interesting exceptions from medical test data with visual summarization. In *Proceedings of the 3rd IEEE International Conference on Data Mining*. 315–322.

Sykacek, P. 1997. Equivalent error bars for neural network classifiers trained by bayesian inference. In *Proceedings of the European Symposium on Artificial Neural Networks*. 121–126.

Tan, P.-N., Steinbach, M., and Kumar, V. 2005a. *Introduction to Data Mining*. Addison-Wesley, Chapter 2, 19–96.

Tan, P.-N., Steinbach, M., and Kumar, V. 2005b. *Introduction to Data Mining*. Addison-Wesley.

Tang, J., Chen, Z., Fu, A. W., and Cheung, D. W. 2006. Capabilities of outlier detection schemes in large datasets, framework and methodologies. *Knowledge and Information Systems 11,* 1, 45–84.

Taniguchi, M., Haft, M., Hollmn, J., and Tresp, V. 1998. Fraud detection in communications networks using neural and probabilistic methods. In *Proceedings of IEEE International Conference in Acoustics, Speech and Signal Processing*. Vol. 2. IEEE Computer Society, 1241–1244.

Tao, Y., Xiao, X., and Zhou, S. 2006. Mining distance-based outliers from large databases in any metric space. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 394–403.

Tarassenko, L. 1995. Novelty detection for the identification of masses in mammograms. In *Proceedings of the 4th IEEE International Conference on Artificial Neural Networks*. Vol. 4. Cambridge, UK, 442–447.

Tax, D. M. J. 2001. One-class classification; concept-learning in the absence of counter-examples. Ph.D. thesis, Delft University of Technology.

Tax, D. M. J. and Duin, R. P. W. 1998. Outlier detection using classifier instability. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*. Springer-Verlag, London, UK, 593–601.

Tax, D. M. J. and Duin, R. P. W. 2001. Combining one-class classifiers. *Lecture Notes in Computer Science 2096*, 299–317.

Teng, C. 2003. Applying noise handling techniques to genomic data: A case study. In *Proceedings of the 3rd IEEE International Conference on Data Mining*. 743–746.

Teng, H., Chen, K., and Lu, S. 1990. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*. IEEE Computer Society Press, 278–284.

Theiler, J. and Cai, D. M. 2003. Resampling approach for anomaly detection in multispectral images. In *Proceedings of SPIE 5093*, 230-240, Ed.

Theofilou, D., Steuber, V., and Schutter, E. 2003. Novelty detection in kohonen-like network with a long-term depression learning rule. *Neurocomputing 52*, 411–417.

Thompson, B., II, R. M., Choi, J., El-Sharkawi, M., Huang, M., and Bunje, C. 2002. Implicit learning in auto-encoder novelty assessment. In *Proceedings of International Joint Conference on Neural Networks*. Honolulu, 2878–2883.

Thottan, M. and Ji, C. 2003. Anomaly detection in ip networks. *IEEE Transactions on Signal Processing 51,* 8, 2191–2204.

Tibshirani, R. and Hastie, T. 2007. Outlier sums for differential gene expression analysis. *Biostatistics 8,* 1, 2–8.

Tomlins, S. A., Rhodes, D. R., Perner, S., Dhanasekaran, S. M., Mehra, R., Sun, X. W., Varambally, S., Cao, X., Tchinda, J., Kuefer, R., Lee, C., Montie, J. E., Shah, R., Pienta, K. J., Rubin, M., and Chinnaiyan, A. M. 2005. Recurrent fusion of tmprss2 and ets transcription factor genes in prostate cancer. *Science 310,* 5748, 603–611.

Torr, P. and Murray, D. 1993. Outlier detection and motion segmentation. In *Proceedings of SPIE, Sensor Fusion VI, Paul S. Schenker; Ed.* Vol. 2059. 432–443.

Tsay, R. S., Pea, D., and Pankratz, A. E. 2000. Outliers in multivariate time series. *Biometrika 87,* 4, 789–804.

Vaidya, J. and Clifton, C. 2004. Privacy-preserving outlier detection. In *Proceedings of the 4th IEEE International Conference on Data Mining*. 233–240.

Valdes, A. and Skinner, K. 2000. Adaptive, model-based monitoring for cyber attack detection. In *Proceedings of the 3rd International Workshop on Recent Advances in Intrusion Detection*. Springer-Verlag, 80–92.

Vapnik, V. N. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Vasconcelos, G., Fairhurst, M., and Bisset, D. 1994. Recognizing novelty in classification tasks. In *Proceedings of Neural Information Processing Systems Workshop on Novelty Detection and Adaptive Systems monitoring*. Denver, CO.

Vasconcelos, G. C., Fairhurst, M. C., and Bisset, D. L. 1995. Investigating feedforward neural networks with respect to the rejection of spurious patterns. *Pattern Recognition Letters 16,* 2, 207–212.

Vigna, G. and Kemmerer, R. A. 1999. Netstat: A network-based intrusion detection system. *Journal of Computer Security 7,* 1.

Vilalta, R. and Ma, S. 2002. Predicting rare events in temporal domains. In *Proceedings of the 2002 IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 474.

Vinueza, A. and Grudic, G. 2004. Unsupervised outlier detection and semi-supervised learning. Tech. Rep. CU-CS-976-04, Univ. of Colorado at Boulder. May.

Weigend, A. S., Mangeas, M., and Srivastava, A. N. 1995. Nonlinear gated experts for time-series - discovering regimes and avoiding overfitting. *International Journal of Neural Systems 6,* 4, 373–399.

Weiss, G. M. and Hirsh, H. 1998. Learning to predict rare events in event sequences. In *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining*, R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, Eds. AAAI Press, Menlo Park, CA, New York, NY, 359–363.

Whitehead, B. and Hoyt, W. 1993. A function approximation approach to anomaly detection in propulsion system test data. In *In Proceedings of 29th AIAA/SAE/ASME/ASEE Joint Propulsion Conference*. IEEE Computer Society, Monterey, CA, USA.

Williams, G., Baxter, R., He, H., Hawkins, S., and Gu, L. 2002. A comparative study of rnn for outlier detection in data mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 709.

Worden, K. 1997. Structural fault detection using a novelty measure. *Journal of Sound Vibration 201,* 1, 85–101.

Wu, N. and Zhang, J. 2003. Factor analysis based anomaly detection. In *Proceedings of IEEE Workshop on Information Assurance*. United States Military Academy, West Point, NY, USA.

Yairi, T., Kato, Y., and Hori, K. 2001. Fault detection by mining association rules from housekeeping data. In *In Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

Yamanishi, K. and ichi Takeuchi, J. 2001. Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 389–394.

Yamanishi, K., Takeuchi, J.-I., Williams, G., and Milne, P. 2004. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery 8*, 275–300.

Yang, J. and Wang, W. 2003. Cluseq: Efficient and effective sequence clustering. In *Proceddings of International Conference on Data Engineering*. 101–112.

Yang, Y., Zhang, J., Carbonell, J., and Jin, C. 2002. Topic-conditioned novelty detection. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 688–693.

Ye, N. 2004. A markov chain model of temporal behavior for anomaly detection. In *Proceedings of the 5th Annual IEEE Information Assurance Workshop*. IEEE.

Ye, N. and Chen, Q. 2001. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International 17*, 105–112.

Yi, B.-K., Sidiropoulos, N., Johnson, T., Jagadish, H. V., Faloutsos, C., and Biliris, A. 2000. Online data mining for co-evolving time sequences. In *Proceedings of the 16th International Conference on Data Engineering*. IEEE Computer Society, Washington, DC, USA, 13.

Ypma, A. and Duin, R. 1998. Novelty detection using self-organizing maps. In *Progress in Connectionist Based Information Systems*. Vol. 2. Springer, 1322–1325.

Yu, D., Sheikholeslami, G., and Zhang, A. 2002. Findout: finding outliers in very large datasets. *Knowledge And Information Systems 4,* 4, 387–412.

Yu, J. X., Qian, W., Lu, H., and Zhou, A. 2006. Finding centric local outliers in categorical/numerical spaces. *Knowledge and Information Systems 9,* 3, 309–338.

Zeevi, A. J., Meir, R., and Adler, R. 1997. Time series prediction using mixtures of experts. In *Advances in Neural Information Processing*. Vol. 9. MIT Press.

Zengyou, H., Xiaofei, X., and Shengchun, D. 2002. Squeezer: an efficient algorithm for clustering categorical data. *Journal of Computer Science and Technology 17,* 5, 611–624.

Zengyou He, Xiaofei Xu, J. Z. H. and Deng, S. 2004a. A frequent pattern discovery method for outlier detection. 726–732.

Zengyou He, Xiaofei Xu, J. Z. H. and Deng, S. 2004b. Mining class outliers: Concepts, algorithms and applications. 588–589.

Zhang, B. and Veenker, G. 1991. Neural networks that teach themselves through genetic discovery of novel examples. In *Proceedings of IEEE International Joint Conference on Neural Networks*. Vol. 1. Singapore, 690–695.

Zhang, J. and Wang, H. 2006. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and Information Systems 10,* 3, 333–355.

Zhang, X., Fan, P., and Zhu, Z. 2003. A new anomaly detection method based on hierarchical hmm. In *Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies*. 249–252.

Zhang, Z., Li, J., Manikopoulos, C., Jorgenson, J., and Ucles, J. 2001. Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In *Proceedings of IEEE Workshop on Information Assurance and Security*. West Point, 85–90.