A Survey of Outlier Detection Methods in Network Anomaly Identification

PRASANTA GOGOI[1], D K BHATTACHARYYA[1], B BORAH[1] AND JUGAL K KALITA[2]

[1]*Department of Computer Science and Engineering, Tezpur University, Napaam*
Tezpur, India 784028
[2]*Department of Computer Science, College of Engineering and Applied Science*
University of Colorado, Colorado Springs
Email: {prasant, dkb, bgb}@tezu.ernet.in kalita@eas.uccs.edu

The detection of outliers has gained considerable interest in data mining with the realization that outliers can be the key discovery to be made from very large databases. Outliers arise due to various reasons such as mechanical faults, changes in system behavior, fraudulent behavior, human error and instrument error. Indeed, for many applications the discovery of outliers leads to more interesting and useful results than the discovery of inliers. Detection of outliers can lead to identification of system faults so that administrators can take preventive measures before they escalate. It is possible that anomaly detection may enable detection of new attacks. Outlier detection is an important anomaly detection approach. In this paper, we present a comprehensive survey of well known distance-based, density-based and other techniques for outlier detection and compare them. We provide definitions of outliers and discuss their detection based on supervised and unsupervised learning in the context of network anomaly detection.

Keywords: Anomaly ; Outlier ; NIDS ; Density-based ; Distance-based ; Unsupervised

Received 27 September 2010; revised 9 February 2011

1. INTRODUCTION

Outlier detection refers to the problem of finding patterns in data that are very different from the rest of the data based on appropriate metrics. Such a pattern often contains useful information regarding abnormal behavior of the system described by the data. These anomalous patterns are usually called outliers, noise, anomalies, exceptions, faults, defects, errors, damage, surprise, novelty or peculiarities in different application domains. Outlier detection is a widely researched problem and finds immense use in application domains such as credit card fraud detection, fraudulent usage of mobile phones, unauthorized access in computer networks, abnormal running conditions in aircraft engine rotation, abnormal flow problems in pipelines, military surveillance for enemy activities and many other areas.

Outlier detection is important due to the fact that outliers can have significant information. Outliers can be candidates for aberrant data that may affect systems adversely such as by producing incorrect results, misspecification of models, and biased estimation of parameters. It is therefore important to identify them prior to modelling and analysis [1]. Outliers in data translate to significant (and often critical) information in a large 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 tasks such as credit card usage monitoring or mobile phone monitoring, a sudden change in usage pattern may indicate fraudulent usage such as stolen cards or stolen phone airtime. In public health data, outlier detection techniques are widely used to detect anomalous patterns in patient medical records, possibly indicating symptoms of a new disease. Outliers can also help discover critical entities such as in military surveillance where the presence of an unusual region in a satellite image in an enemy area could indicate enemy troop movement. In many safety critical environments, the presence of an outlier indicates abnormal running conditions from which significant performance degradation may result, such as an aircraft engine rotation defect or a flow problem in a pipeline.

An outlier detection algorithm may need access to certain information to work. A labelled training data set is one such piece of information that can be used with techniques from machine learning [2] and statistical learning theory [3]). A training data set is required by techniques which build an explicit predictive model. The labels associated with a data instance denote if that instance is *normal* or *outlier*. Based on the extent to which these labels are available or utilized, outlier detection techniques can be either *supervised* or *unsupervised*. Supervised outlier detection techniques assume the availability of a training data set which has labelled instances for the normal as well as the outlier class. In such techniques, predictive models are built for both normal and outlier classes. Any unseen data instance is compared against the two models to determine which class it belongs to. An unsupervised outlier detection technique makes no assumption about the availability of labelled training data. Thus, these techniques are more widely applicable. The techniques in this class make other assumptions about the data. For example, parametric statistical techniques assume a parametric distribution for one or both classes of instances. 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.

Outlier detection is of interest in many practical applications. For example, an unusual flow of network packets, revealed by analysing system logs, may be classified as an outlier, because it may be a virus attack [4] or an attempt at an intrusion. Another example is automatic systems for preventing fraudulent use of credit cards. These systems detect unusual transactions and may block such transactions in early stages, preventing, large losses. The problem of outlier detection typically arises in the context of very high dimensional data sets. However, much of the recent work on finding outliers uses methods which make implicit assumptions regarding relatively low dimensionality of the data. A specific point to note in outlier detection is that the great majority of objects analysed are not outliers. Moreover, in many cases, it is not a priori known what objects are outliers.

1.1. Outlier Detection in Anomaly Detection

The anomaly detection problem is similar to the problem of finding outliers, specifically, in network intrusion detection. Intrusion detection is a part of a security management system for computers and networks. Intrusion [5] is a set of actions aimed to compromise computer security goals such as confidentiality, integrity and availability of resources. Traditional technologies such as firewalls are used to build a manual passive defence system against attacks. An Intrusion Detection System (IDS) is usually used to enhance the network security of enterprises by monitoring and analysing network data packets. Intrusion detection is a system's "second line of defence" [6]. IDSs play a vital role in network security. Network intrusion detection systems (NIDSs) can detect attacks by observing network activities. Intrusion detection techniques are used, primarily, for misuse detection and anomaly detection. Misuse based detection involves an attempt to define a set of rules (also called *signatures*) that can be used to decide that a given behavior is that of an intruder. For example, Snort [7] is a misuse based NIDS. The other approach, anomaly detection, involves the collection of data relating to the behavior of legitimate users over a period of time, and then applying tests to the gathered data to determine whether that behavior is legitimate user behavior or not. Anomaly detection has the advantage that it can detect new attacks that the system has never seen before as they deviate from normal behavior. ADAM [8] is a well known anomaly detection NIDS.

The key challenge for outlier detection in this domain is the huge volume of data. Outlier detection schemes need to be computationally efficient to handle these large sized inputs. An outlier can be an observation that is distinctly different or is at a position of abnormal distance from other values in the dataset. Detection of abnormal behavior can be based on features extracted from traces such as network trace or system call trace [9]. An intrusion can be detected by finding an outlier whose features are distinctly different from the rest of the data. Outliers can often be individuals or groups of clients exhibiting behavior outside the range of what is considered *normal*. In order to apply outlier detection to anomaly based network intrusion detection, it is assumed [10] that -

1. The majority of the network connections are normal traffic. Only a small amount of traffic is malicious.
2. Attack traffic is statistically different from normal traffic.

However, in a real-world network scenario, these assumptions may not be always true. For example, when dealing with DDoS (distributed denial of service) [11] or bursty attack [12] detection in computer networks, the anomalous traffic is actually more frequent than the normal traffic.

1.2. Contribution of The Paper

Outlier detection methods have been used for numerous applications in various domains. A lot of these techniques have been developed to solve focused problems in a particular application domain, while others have been developed in a more generic fashion. Outlier detection approaches found in literature [13, 14, 15, 16] have varying scopes and abilities.

The selection of an approach for detection of outlier(s) depends on the domain of application, type of data (e.g., numeric, categorical or mixed) and availability of labeled data. So, an adequate knowledge is highly essential regarding existing approaches to outlier detection while selecting an appropriate method for a specific domain. In this paper, we aim to provide a comprehensive up-to-date survey on outlier detection methods and approaches to network anomaly identification by using outlier detection methods. In particular, this paper contributes to the literature in outlier detection in the following ways.

- We have found general surveys on outlier detection such as [13, 14, 15, 16, 17, 18] and surveys on network anomaly detection such as [19, 20, 21]. But survey papers on the specific topic of anomaly identification using outlier detection method are not available. This survey emphasizes anomaly identification by using outlier detection approach.
- In network traffic, most traffic is normal. Traffic related to attacks is naturally rare and therfore outlier. Thus, it is befitting that the problem of network anomaly identification be studied as an outlier detection problem. So, it will be beneficial for researchers as well as practitioners to have a resource where papers that use outlier detection for network anomaly identification are surveyed.
- We believe that our classification of outliers into six cases provides a unique and novel insight into understanding the concept of outlier. This insight is likely to have implications on the design and development of algorithms for outlier detection whether for network anomaly detection or other context.
- Although other surveys classify outlier detection techniques into the categories of supervised and unsupervised, our survey is most up-to-date.
- Our classification of anomaly scores into three categories is also novel. An appropriate selection of anomaly score is crucial when applying outlier detection methods in specific domains. This survey will help readers select an appropriate anomaly score for their purpose.
- We identify various key research issues and challenges of outlier detection methods in network anomaly identification.

1.3. Organization of The Paper

The remainder of this paper is organized as follows. In the next section, we present preliminaries necessary to understand outlier detection methodologies. In Section 3, we explain issues in anomaly detection of network intrusion detection. Existing outlier detection approaches and a classification of these approaches are presented in Section 4. In Section 5, we outline various research challenges and possibilities of future work. Finally, Section 6 concludes the paper.

2. PRELIMINARIES

Outlier detection searches for objects that do not obey rules and expectations valid for the major part of the data. The detection of an outlier object may be an evidence that there are new tendencies in data. Although, outliers are considered noise or errors, they may have important information. What is an outlier often depends on the applied detection methods and hidden assumptions regarding data structures used. Depending on the approaches used in outlier detection, the methodologies can be broadly classified as:

1. Distance-based,
2. Density-based, and
3. Machine learning or soft-computing based.

These are discussed below.

2.1. Distance-based Outlier Detection

Distance-based methods for outlier detection are based on the calculation of distances among objects in the data with clear geometric interpretation. We can calculate a so-called outlier factor as a function $F : x \rightarrow R$ to quantitatively characterize an outlier [14]. The function F depends on the distance between the given object x and other objects R in the dataset being analysed. We introduce some commonly available definitions of *distance-based* outlier detection from [22, 23, 24] below.

Definition 1: Hawkins Outlier - Outliers are observations which deviate significantly from other observations as to arouse suspicion that these are generated by a different mechanism [22].

This notion is formalized by Knorr and Ng [23] as follows: Let o, p, q denote objects in a dataset and let $d(p, q)$ denote the distance between objects p and q. C is a set of objects and $d(p, C)$ denotes the minimum distance between p and object q in C:

$$d(p, C) = min \{d(p, q) | q \in C\} . \tag{1}$$

Definition 2: DB(pct, dmin) Outlier - An object p in a dataset D is a $DB(pct, dmin)$ outlier if at least pct percentage of the objects in D lies at distance greater than $dmin$ from p, i.e., the cardinality of the set $\{q \in D | d(p, q) \leq dmin\}$ is less than or equal to $(100 - pct)\%$ of the size of D [23].

To illustrate, consider a 2-D data set depicted in *Fig.* 1. This is a simple 2-dimensional dataset containing 602 objects. There are 500 objects in the first cluster C_1, 100 objects in the cluster C_2, and two additional objects O_1 and O_2. In this example, C_2 forms a denser cluster than C_1. According to Hawkins' definition, both O_1 and O_2 are outliers, whereas objects in C_1 and C_2 are not. In contrast, within the framework of distance-based outliers, only O_1 is a reasonable $DB(pct, dmin)$-outlier in the following sense.

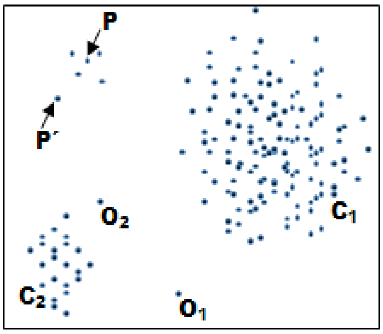

FIGURE 1. A 2-D data set

If for every object q^{O_i} in C_1, the distance between q^{O_i} and its nearest neighbour is greater than the distance between O_2 and C_2 (i.e., $d(O_2, C_2)$), we can show that there is no appropriate value of *pct* and *dmin* such that O_2 is a $DB(pct, dmin)$-outlier but the objects in C_1 are not. The reason is as follows. If the *dmin* value is less than the distance $d(O_2, C_2)$, all 601 objects ($pct = 100 * 601/602$) are further away from O_2 than *dmin*. But the same condition holds also for every object q in C_1. Thus, in this case, O_2 and all objects in C_1 are $DB(pct, dmin)$ outliers. Otherwise, if the *dmin* value is greater than the distance $d(O_2, C_2)$, it is easy to see that O_2 is a $DB(pct, dmin)$ outlier implying that there are many objects q in C_1 such that q is also a $DB(pct, dmin)$ outlier. This is because the cardinality of the set $p \in D | d(p, O_2) \leq dmin$ is always bigger than the cardinality of the set $p \in D | d(p, q) \leq dmin$.

Definition 3: D_n^k Outlier - Given an input dataset with N points, parameters n and k can be used to denote a D_n^k outlier for a point p if there are no more than n-1 other points p' such that $D^k(p') > D^k(p)$ [24].

$D^k(p)$ denotes distance of point p from its k^{th} nearest neighbour. The points can be ranked according to their $D^k(p)$ distances. For a given value of k and n, a point p is an outlier if no more than $n-1$ other points in the data set have a higher value of distance than $D^k(p)$. As can be seen in $Fig.$ 1 for $n = 6$, D_6^k is outlier for a point p, since there is no more than $(6-1) = 5$ other points p', such that $D^k(p') > D^k(p)$.

Definition 3 has intuitive appeal to rank each point based on its distance from its k^{th} nearest neighbour. With this definition of outliers, it is possible to rank outliers based on $D^k(p)$ distances. Outliers with larger $D^k(p)$ distances have fewer points close to them and are thus intuitively stronger outliers. Various proximity measures can be used to measure the distance between a pair of points with numeric as well as categorical data.

Based on these definitions, we observe that distance-based outliers are data points that are situated away from the majority of points using some geometric distance measure following a fixed or changeable threshold related to the domain of interest. The advantage of distance-based methods is the high degree of accuracy of distance measures. High dimensional data is always sparse related to some dimension or attribute. Because of the sparsity of data, distance-based approaches usually do not perform well in situations where the actual values of the distances are similar for many pairs of points. So, researchers [25] working with *distance-based* outlier detection methods take a non-parametric approach. Although distance-based approaches for outlier detection are non-parametric, the drawback is the amount of computation time required. In *distance-based* outlier detection methods, effective use of the adaptive or conditional threshold value can result in better performance.

2.2. Density-based Outlier Detection

The *density-based* approach was originally proposed in [26]. *Density-based* methods estimate the density distribution of the input space and then identify outliers as those lying in regions of low density [27]. *Density-based* outlier detection techniques estimate the density of the neighbourhood of each data instance. An instance that lies in a neighbourhood with low density is declared to be an outlier while an instance that lies in a dense neighbourhood is declared to be normal. A generalized definition of density-based outlier based on [28] is given next. This approach is very sensitive to parameters defining the neighbourhood. The definition is complex and therefore, is introduced in several steps.

Definition 4: LOF based Outlier - A local outlier factor (LOF) [28] is computed for each object in the dataset, indicating its degree of outlierness. This quantifies how outlying an object is. The outlier factor is local in the sense that only a restricted neighbourhood of each object is taken into account. The LOF of an object is based on the single parameter called $MinPts$, which is the number of nearest neighbours used in defining the local neighbourhood of the object. The LOF of an object p can be defined as

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}. \qquad (2)$$

The outlier factor of object p captures the degree to which we can call p an outlier. It is the average of the ratio of the local reachability density of p and those of p's $MinPts$-nearest neighbours. The lower p's local reachability density (lrd) is, and the higher lrd of p's $MinPts$-nearest neighbours are, the higher is the LOF value of p.

The local reachability density (lrd) of an object p is the inverse of the average reachability distance (*reach-dist*) based on the $MinPts$ nearest neighbours of p. Note that the local density can be ∞ if all the reachability distances in the summation are 0. This may occur for an object p if there are at least $MinPts$ objects, different from p, but sharing the same spatial coordinates, i.e., if there are at least $MinPts$ duplicates of p in the dataset. lrd is defined as:

$$lrd_{MinPts}(p) = \left(\frac{\sum\limits_{o \in N_{MinPts}(p)} reach\text{-}dist_{MinPts}(p,o)}{|N_{MinPts}(p)|} \right)^{-1} . \tag{3}$$

The reachability distance of an object p with respect to object o is $reach\text{-}dist_{MinPts}(p,o)$:
$reach\text{-}dist_{MinPts}(p,o)$

$$= max \left\{ MinPts\text{-}dist(o), dist(p,o) \right\} . \tag{4}$$

For any positive integer k, the k-distance of object p, denoted as k-distance(p), is defined as the distance $d(p,o)$ between p and an object $o \in D$ where D is a dataset such that:

1. for at least k objects $o' \in D \mid \{p\}$ it holds that $d(p,o') \leq d(p,o)$, and
2. for at most k-1 objects $o' \in D \mid \{p\}$ it holds that $d(p,o') < d(p,o)$.

The *k-distance neighborhood* of p contains every object whose distance from p is not greater than the k-distance, i.e., $N_{k\text{-}distance(p)}(p) = q \in D| \{p\} |d(p,q) \leq k\text{-}distance(p)$. These objects q are called the k-nearest neighbours of p. The notation $N_k(p)$ is used as a shorthand for $N_{k\text{-}distance(p)}(p)$. k-distance(p) is well defined for any positive integer k, although the object o may not be unique. In such a case, the cardinality of $N_k(p)$ is greater than k. For example, suppose that there are: (i) 1 object with distance 1 unit from p; (ii) 2 objects with distance 2 units from p; and (iii) 3 objects with distance 3 units from p. Then 2-*distance*(p) is identical to 3-*distance*(p). Assume now that there are 3 objects of 4-*distance*(p) from p. Thus, the cardinality of $N_4(p)$ can be greater than 4; in this case it is 6.

Fig. 2 illustrates the idea of reachability distance with $k = 4$. Intuitively, if object p is far away from o (e.g., p_2 in the figure), the reachability distance between the two is simply their actual distance. However, if they are sufficiently close (e.g., p_1 in the figure), the actual distance is replaced by the k-distance of o. The reason is that in so doing, the statistical fluctuations of $d(p,o)$ for all the p's close to o can be significantly reduced. The strength of this smoothing effect can be controlled by the parameter k. The higher the value of k, the more similar the reachability distances for objects within the same neighbourhood.

Density-based outlier detection is a parameter based approach. The performance of a *density-based* method is largely dependent on optimized parameter selection. With reference to intrusion detection, we can consider density-based outliers as data points lying in low density regions with respect to specific attributes or parameters.

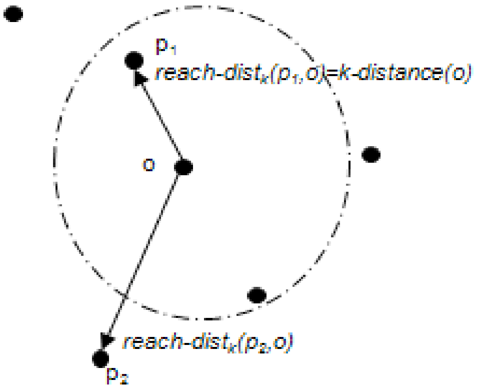

FIGURE 2. Reachability distance

2.3. Outlier Detection based on Soft computing Approaches

In this section we present some definitions of outliers inspired by soft computing approaches.

Definition 5: RMF (rough membership function)-based outliers [29, 30]. A RMF is defined as follows. Let IS=(U, A, V, f) be an information system, $X \subseteq U$ and $X \neq \Phi$. U is a non-empty finite set of objects, A a set of attributes, V the union of attribute domains, and $f : U \times A \rightarrow V$ a function such that for any $X \in U$ and $a \in A$, $f(x,a) \in V_a$. Let ν be a given threshold value. For any $x \in X$, if $ROF_X(x) > \nu$, x is called a *rough membership function* (*RMF*)*-based outlier* with respect to X in IS, where $ROF_X(x)$ is the rough outlier factor of x with respect to X in IS. The *rough outlier factor* is defined as
$ROF_X(x)$

$$= 1 - \frac{\sum\limits_{j=1}^{m} \left(\mu_X^{A_j}(x) \times |A_j| \right) + \sum\limits_{j=1}^{m} \left(\mu_X^{\{a_j\}}(x) \times W_X^{\{a_j\}}(x) \right)}{2 \times |A|^2} \tag{5}$$

where A=$\{a_1, a_2, \ldots, a_m\}$. $\mu_X^{A_j}(x)$ and $\mu_X^{\{a_j\}}(x)$ are RMFs for every attribute subset $A_j \subseteq A$ and singleton subset $\{a_j\}$ of A, $1 \leq j \leq m$. For every singleton subset $\{a_j\}$, $W_X^{\{a_j\}} : X \rightarrow (0,1]$ is a weight function such that for any $x \in X$, $W_X^{\{a_j\}}(x) = \sqrt{(|\, [x]_{\{a_j\}} \,|)/\,(|U|)}$. $[x]_{\{a_j\}} = \{u \in U : f(u,a_j) = f(x,a_j)\}$ denotes the indiscernibility class of relation IND($\{a_j\}$) that contains element x.

The RMF is $\mu_X^B :\rightarrow (0,1]$ such that for any $x \in X$

$$\mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \tag{6}$$

where $[x]_B = \{u \in U : \forall_a \in B(f(u,a) = f(x,a))\}$ and $B \subseteq A$ denotes the indiscernibility class of relation

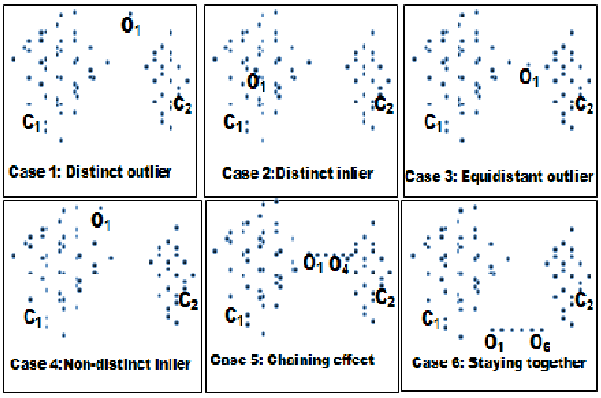

FIGURE 3. Six cases

IND(B) that contains element x.

Rough sets are used in classification system, where we do not have complete knowledge of the system [31]. In any classification task the aim is to form various classes where each class contains objects that are not noticeably different. These indiscernible or indistinguishable objects can be viewed as basic building blocks (concepts) used to build a knowledge base about the real world. This kind of uncertainty is referred to as rough uncertainty. Rough uncertainty is formulated in terms of rough sets.

In fuzzy sets, the membership of an element in a set is not crisp. It can be anything in between yes and no. The concept of fuzzy sets is important in pattern classification. Thus, fuzzy and rough sets represent different facets of uncertainty. Fuzziness deals with vagueness among overlapping sets [32]. On the other hand, rough sets deal with coarse non-overlapping concepts [33]. Neither roughness nor fuzziness depends on the occurrence of an event. In fuzzy sets, each granule of knowledge can have only one membership value for a particular class. However, rough sets assert that each granule may have different membership values for the same class. Thus, roughness appears due to indiscernibility in the input pattern set, and fuzziness is generated due to the vagueness present in the output class and the clusters. To model this type of situation, where both vagueness and approximation are present, the concept of fuzzy-rough sets [33] can be employed.

2.4. Comparison of outlier Detection Approaches

Outlier detection has largely focused on data that is univariate, and data with a known (or parametric or density-based) distribution. These two limitations have restricted the ability to apply outlier detection methods to large real-world databases which typically have many different fields and have no easy way of characterizing the multivariate distribution.

TABLE 1. A General Comparison of Three Outlier Detection Approaches

Approaches	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Distance-based	Yes	Yes	Yes	Yes	Yes	No
Density-based	Yes	Yes	No	Yes	Partially	Partially
Soft computing	Yes	Yes	Yes	Yes	Partially	No

To evaluate the effectiveness of outlier detection methods, we consider six cases over the synthetic data set given in $Fig.$ 3. In these figures, O_i is an object and C_i is cluster of objects. A *distinct outlier* object, (case 1 in $Fig.$ 3) is one that cannot be included in any clusters, whereas a *distinct inlier* object (case 2) is inside of a cluster. An *equidistant outlier* (case 3) is the object which is at equal distance from the clusters, whereas *non-distinct inlier* (case 4) is that object located near the border of a cluster. The *chaining effect* (case 5) represents the objects which are situated in a straight line among the clusters. However, the *staying together (case 6)* effect represents the objects which are outliers on a straight line.

A comparison table of three approaches in the context of these six cases over synthetic data is given in *Table 1.*

3. NETWORK ANOMALY DETECTION

Unusual activities are outliers that are inconsistent with the remainder of data set [34]. An outlier is an observation that lies at an abnormal distance from other values in the dataset. In order to apply outlier detection to anomaly detection in the context of network intrusion detection, it is assumed by most researchers [10] that (1) the majority of the network connections are normal traffic. Only a small fraction of the traffic is malicious. (2) The attack traffic is statistically different from normal traffic. An anomaly detection technique identifies attacks based on deviations from the established profiles of normal activities. Activities that exceed thresholds of the deviations are identified as attacks. Thus, supervised and unsupervised outlier detection methods can find anomaly in network intrusion detection.

Network intrusion detection systems [35] deal with detecting intrusions in network data. The primary reason for these intrusions is attacks launched by outside hackers who want to gain unauthorized access to the network to steal information or to disrupt the network. A typical setting is a large network of computers which is connected to the rest of the world through the Internet. Generally, detection of an intrusion in a network system is carried out based on two basic approaches - signature based and anomaly based. A signature based approach attempts to find attacks based on the previously stored patterns or signatures for known intrusions. However, an anomaly based approach can detect intrusions based on deviations from the previously stored profiles of normal activities, but also, capable of detecting unknown intrusions (or suspicious pattern). The NIDS does this by reading all incoming packets and trying to find suspicious patterns.

3.1. Network Anomalies and Types

Anomaly detection attempts to find data patterns that are deviations in that they do not conform to expected behavior. These deviations or non-conforming patterns are anomalies. Based on the nature, context, behavior or cardinality, anomalies are generally classified into following three categories [10]:

1. *Point Anomalies*- This simplest type of anomaly is an instance of the data that has been found to be anomalous with respect to the rest of the data. In a majority of applications, this type of anomaly occurs and a good amount of research addresses this issue.
2. *Contextual Anomalies*- This type of anomaly (also known as conditional anomaly [36]) is defined for a data instance in a specific context. Generally, the notion of a context is induced by the structure in the data set. Two sets of attributes determine if a data instance belongs to this type of anomaly: (i) *Contextual attributes* and (ii) *Behavioural attributes*. Contextual attributes determine the context (or neighbourhood) for that instance. For example, in stock exchange or gene expression time series datasets, time is a contextual attribute that helps to specify the position of the instance in the entire sequence. However, behavioural attributes are responsible for the non-contextual characteristics of an instance. For example, in a spatial data set describing the average number of people infected by a specific disease in a country, the amount of infection at a specific location can be defined as a behavioural attribute.
3. *Collective Anomalies*- These are collections of related data instances found to be anomalous with respect to the entire set of data. In collective anomaly, the individual data instances may not be anomalous by themselves, however, their collective occurrence is anomalous.

To handle the above types of anomalies, various detection techniques have been proposed over the decades. Especially, to handle the point anomaly type, distance-based approaches have been found suitable. However, the effectiveness of such techniques largely depends on the type of data, proximity measure or anomaly score used and the dimensionality of the data. In the case of contextual anomaly detection, both distance-based and density-based approaches have been found suitable. However, like the previous case, in the case of distance-based outlier approaches, the proximity measure used, type of data, data dimensionality and the threshold measure play a vital role. The density-based approach can handle this type of anomaly effectively for uniformly distributed datasets. However, in the case of skewed distributions, an appropriate density threshold is required for handling the variable density situation. Collective anomaly is mostly handled by using density-based approaches. However, in the identification of this type of anomalous patterns, other factors also play crucial role (such as compactness, and single linkage effects)

3.2. Characterizing ANIDS

An *ANIDS* is an anomaly based network intrusion detection system. A variety of ANIDSs have been proposed since the mid 1980s. It is necessary to have a clear definition of anomaly in the context of network intrusion detection. The majority of current research on ANIDS does not explicitly state what constitute anomaly in their study [37]. In a recent survey [19], anomaly detection methods were classified into two classes: generative and discriminative. Generally, an ANIDS is characterized based on the following attributes: (i) nature and type of the input data, (ii) appropriateness of similarity/dissimilarity measures, (iii) labelling of data and (iv) reporting of anomalies. Next, we discuss each of these issues.

3.2.1. Types of Data

A key aspect of any anomaly detection technique is the nature of the input data. The input is generally a collection of data instances or objects. Each data instance can be described using a set of attributes (also referred to as variables, characteristics, features, fields or dimensions). The attributes can be of different types such as binary, categorical or continuous. Each data instance may consist of only one attribute (univariate) or multiple attributes (multivariate). In the case of multivariate data instances, all attributes may be of the same type or may be a mixture of different data types.

3.2.2. Proximity measures

Distance or *similarity measures* are necessary to solve many pattern recognition problems such as classification, clustering, and retrieval problems. From scientific and mathematical points of view, distance is defined as a quantitative degree of how far apart two objects are. A synonym for distance is *dissimilarity*. Distance measures satisfying the metric properties are simply called metric while non-metric distance measures are occasionally called *divergence*. A synonym for similarity is *proximity* and similarity measures are often called *similarity coefficients*. The selection of a proximity measure is very difficult because it depends upon the (i) the types of attributes in the data (ii) the dimensionality of data and (iii) the problem of weighing data attributes. In the case of numeric data objects, their inherent geometric properties can be exploited naturally to define distance functions between two data points. Numeric objects may be discrete or continuous. A detailed discussion on the various proximity measures for numeric data can be found in [38]. Categorical attribute values cannot be naturally arranged as numerical values. Computing similarity between categorical data instances is not straightforward. Several data-driven similarity measures have been proposed [39] for categorical data. The behavior of such measures directly depends on the data. Mixed type datasets include categorical and numeric values. A common practice for clustering mixed datasets is to transform categorical values into numeric values and then use a proximity measure for numeric data. Another approach [27] is to compare the categorical values directly, in which two distinct values result in distance 1 while two identical values result in distance 0.

3.2.3. Data Labels

The labels associated with a data instance denote if that instance is normal or anomalous. It should be noted that obtaining labelled data that is accurate as well as representative of all types of behaviours, is often prohibitively expensive. Labelling is often done manually by a human expert and hence requires substantial effort to obtain the labelled training data set. Several active learning approaches for creating labelled datasets have also been proposed [40]. Typically, getting a labelled set of anomalous data instances which cover all possible type of anomalous behavior is more difficult than getting labels for normal behavior. Moreover, anomalous behavior is often dynamic in nature, e.g., new types of anomalies may arise, for which there is no labelled training data. The KDD CUP '99 dataset [41] is an evaluated intrusion data set with labelled training and testing data. Based on the extent to which labels are available, anomaly detection techniques can operate either in supervised or unsupervised approaches. A supervised approach usually trains the system with normal patterns and attempts to detect an attack based on its non-conformity with reference to normal patterns. In case of the KDD CUP '99 dataset, the attack data are labelled into four classes – DoS (denial of service), R2L (remote to local), U2R (user to root), and probe. This defines the intrusion detection problem as a 5-class problem. If attack data are labelled into n possible classes, we have an $(n + 1)$-class problem at hand. An unsupervised approach does not need a labelled dataset. Once the system identifies the meaningful clusters, it applies the appropriate labelling techniques for identified clusters. A supervised approach has high detection rate (DR) and low false positive rate (FPR) of attack detection compared to an unsupervised approach. Supervised approaches can detect known attacks whereas unsupervised approaches can detect unknown attacks as well.

3.2.4. Anomaly Scores

Detection of anomalies depends on scoring techniques that assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly. Thus the output of such a technique is a ranked list of anomalies. An analyst may choose to either analyse the top few anomalies or use a cut-off threshold to select anomalies. Several anomaly score estimation techniques have been developed in the past decades. Some of them have been represented under the category of *distance-based*, *density-based* and machine learning or soft computing based approach.

A Distance-based anomaly scores

In this section, we introduce some of the popular *distance-based* anomaly score estimation techniques.

A.1 *LOADED (Link-based Outlier and Anomaly Detection in Evolving Data Sets) Anomaly Score* [42] - Assume our data set contains both continuous and categorical attributes. Two data points p_i and p_j are considered linked if they are considerably similar to each other. Moreover, associated with each link is a link strength that captures the degree of linkage, and is determined using a similarity metric defined on the two points. The data points p_i and p_j are linked in a categorical attribute space if they have at least one attribute-value pair in common. The associated link strength is equal to the number of attribute-value pairs shared in common between the two points. A score function that generates high scores for outliers assigns score to a point that is inversely proportional to the sum of the strengths of all its links. To estimate this score efficiently, ideas from frequent itemset mining are used. Let I be the set of all possible attribute-value pairs in the data set M. Let $D = \{d : d \in PowerSet(I) \wedge \forall_{i,j:i \neq j} d_i \cdot attrib \neq d_j \cdot attrib\}$ be the set of all itemsets, where an attribute only occurs once per itemset. The score function for a categorical attribute is defined as:

$$Score_1(p_i) = \sum_{d \subseteq p_i} \left(\frac{1}{|d|} |sup(d) \leq s \right) \qquad (7)$$

where p_i is an ordered set of categorical attributes. $sup(d)$ is the number of points p_i in the data set where $d \subseteq p_i$, otherwise known as *support* of itemset d. $|d|$ is the number of attribute-value pairs in d. s is a user-defined threshold of minimum support or minimum number of links.

A point is defined to be linked to another point in the mixed data space if they are linked together in the categorical data space and if their continuous attributes adhere to the joint distribution as indicated by the correlation matrix. Points that violate these conditions are defined to be outliers. The modified score function for mixed attribute data is as follows:

$$Score_2(p_i) = \sum_{d \subseteq p_i} \left(\frac{1}{|d|} |(C_1 \vee C_2) \wedge C_3 \ is\ true \right) \tag{8}$$

where $C_1 : sup(d) \leq s$, C_2 : at least δ% of the correlation coefficients disagree with the distribution followed by the continuous attributes for point p_i, and C_3 : C_1 or C_2 hold true for every superset of d in p_i. Condition C_1 is the same condition used to find outliers in a categorical data space using $Score_1(p_i)$. Condition C_2 adds continuous attribute checks to $Score_1(p_i)$. Condition C_3 is a heuristic and allows for more efficient processing because if an itemset does not satisfy conditions C_1 and C_2, none of its subsets are considered.

A.2 *RELOADED (REduced memory LOADED) Anomaly Score* [43]- An anomalous data point can be defined as one that has a subset of attributes that take on unusual values given the values of the other attributes. When all categorical attributes of a data point have been processed, the anomaly score of the data point is computed as a function of the count of incorrect predictions and the violation score as below:

$$AnomalyScore[P_i] = \frac{\left(\frac{\sum_{j=1}^{m} i - W_j}{i} \right)}{m} + \frac{V_\tau}{mn^2} \tag{9}$$

where W_j is the cumulative number of incorrect predictions of categorical attribute j for the previous i data points. There are m categorical attributes and n continuous attributes. V_τ is cumulative violation score of point P_i.

B Density-based anomaly scores

Here, we introduce a few *density-based* anomaly score estimation techniques.

B.1 *ODMAD (Outlier Detection for Mixed Attribute Datasets) Anomaly Score* [27] - This score can be used in an approach that mines outliers from data containing both categorical and continuous attributes. The ODMAD score is computed for each point taking into consideration the irregularity of the categorical values, the continuous values, and the relationship between the two spaces in the dataset. A good indicator to decide if point X_i is an outlier in the categorical attribute space is the score value, $Score_1$, defined below:

$$Score_1(X_i) = \sum_{d \subseteq X_i \wedge supp(d) < \sigma \wedge |d| \leq Max} \frac{1}{supp(d) \times |d|} \tag{10}$$

where X_i is a data point with m_c categorical attributes in a dataset D. Let T be the set of all possible combinations of attribute and value pairs in the dataset D. Let S be the set of all sets d so that an attribute occurs only once in each set d. Then,

$$S = \{d : d \in PowerSet(T) \wedge \forall\, l, k \in d, l \neq k\} \tag{11}$$

where l and k represent attributes whose values appear in set d. $|d|$ represents length of set d. δ is a user-defined threshold. A point can be an outlier if it contains single values that are infrequent or sets of values that are infrequent. A categorical value or a combination of values is infrequent if it appears less than δ times in dataset. Max is a user-defined length of attribute set.

In the case of mixed attribute datasets a modified score is defined for the data points that share the same categorical value as well as similar continuous values as below:

$$Score_2(X_i) = \frac{1}{|a \ \in \ X_i^c|} \times \sum_{\forall a \in \ X_i^c} cos\,(X_i^q, \mu_a) \tag{12}$$

where X_i is a data point containing m_c categorical values and m_q continuous values. X_i^c and X_i^q are respectively the categorical and the continuous parts of X_i. Let a be one of the categorical values of X_i^c that occurs with support $supp(a)$. Let a subset of the data that contains the continuous vectors corresponding to the data points that share categorical value a be $\{X_i^q : a \in X_i^c, i = 1 \ldots n\}$, with a total of $supp(a)$. The mean vector of this set, μ_a, is below:

$$\mu_a = \frac{1}{supp(a)} \times \sum_{i=1 \wedge a \in X_i^c}^{n} X_i^q. \tag{13}$$

Also, the cosine similarity between a point X_i^q and mean μ_a is given below:

$$cos\,(X_i^q, \mu_a) = \sum_{j=1}^{m_q} \left(\frac{x_{ij}^q}{\|X_i^q\|} \times \frac{\mu_{aj}}{\|\mu_a\|} \right) \tag{14}$$

where $\|X\|$ represents the L_2-norm of vector X. $Score_2(X_i)$ is a score for each point X_i; for all categorical values a contained in X_i^c, this is the summation of all cosine similarities for all categorical values a divided by the total number of values in the categorical part of X_i, X_i^c. As minimum cosine similarity is 0 and maximum is 1, the data points with similarity close to 0 are more likely to be outliers.

C Machine learning or soft-computing based anomaly scores
This section presents a few machine learning or soft computing based anomaly score esimation techniques.

C.1 *RNN (Replicator Neural Network) Outlier Detection Anomaly Score* [44] - This score has been used in feed-forward multi-layer perceptron network approaches to anomaly detection. The Outlier Factor δ_i of the i-th data record is the measure of outlierness. δ_i is defined by the average reconstruction error over all features (variables):

$$\delta_i = \frac{1}{n}\sum_{j=1}^{n}(x_{ij} - o_{ij})^2 \qquad (15)$$

where x_{ij}'s are reconstruction data instances, o_{ij} are reconstruction output instances and n is the number of features over which the data is defined. The reconstruction error is used as anomaly score.

C.2 *GMM (Gaussian Mixture Model) Anomaly Detection* [36] - Assume each data instance d is represented as $[x, y]$. If U is the contextual data and V the behavioural data, the mapping function $p(V_j|U_i)$ 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. The anomaly score for a test instance d is given as:
Anomaly Score

$$= \sum_{i=1}^{n_U} p(x \in U_i) \sum_{j=1}^{n_V} p(y \in V_j)\ p(V_j|U_i) \qquad (16)$$

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 that a sample point x is generated from the mixture component U_i while $p(y \in V_j)$ indicates the probability that a sample point y is generated from the mixture component V_j.

C.3 *Markov Chain Model Anomaly Score* [45] - This model is used to represent a temporal profile of normal behavior in a computer and network system. The Markov chain model of the normal profile is learned from historic data of the system's normal behavior. The observed behavior of the system is analysed to infer the probability that the Markov chain model of the normal profile supports the observed behavior. A low probability of support indicates an anomalous behavior that may result from intrusive activities. The likelihood $P(S)$ of sequence S is given as:

$$P(S) = q_{S_1}\prod_{t=2}^{|S|} p_{S_{t-1}S_t} \qquad (17)$$

where q_{S_1} is the probability of observing the symbol S_1 in the training set and $p_{S_{t-1}S_t}$ is the probability of observing the symbol S_t after S_{t-1} in the training set. The inverse of $P(S)$ is the anomaly score for given sequence S.

A general comparison of the effectiveness of various anomaly/outlier scores reported in the previous subsections can be made based on parameters such as detection approaches used (density and distance), attribute types of data and applications under considerations. A summary of comparisons are given in *Table 2*.

3.2.5. Datasets Used

Network intrusion detection is a problem of handling of high dimensional mixed type data. Most approaches considered here are able to handle categorical, numerical or mixed type high dimensional data.

Most techniques are evaluated based on KDD Cup 1999 intrusion detection dataset. However, this dataset has several limitations such as (i) the dataset is not unbiased and (ii) it is a purified dataset that does not contain fragment data.

Several acedemic and research laboratories, commercial organizations have generated unbiased intrusion datasets to evaluate IDSs and associated datasets. Prototype IDS testing platforms have been developed by University of California at Davis [46] and IBM Zurich [47]. A rigorous and extensive IDS testing was performed by MIT Lincoln Laboratory [48]. The Air Force Research Laboratory [49] has also been involved in IDS testing in a complex hierarchical network environment. The MITRE Corporation [50] investigated the characteristics and capabilities of network base IDS.

4. EXISTING OUTLIER DETECTION APPROACHES FOR NETWORK ANOMALY DETECTION

Outlier detection is a critical task in many safety critical environments as outliers indicate abnormal running conditions from which significant performance degradation may result. We can categorise and analyse a broad range of outlier detection methodologies as either supervised or unsupervised approaches.

4.1. Supervised Approaches

The supervised approaches are essentially supervised classification and require pre-labelled data, tagged as

TABLE 2. Comparison of Anomaly Score

Author & Year	**Score Formula**	**Approach (Density /Distance/soft-computing)**	**Data type**	**Applications**
Ye, 2000	$P(S) = q_{S_1} \prod_{t=2}^{\lvert S \rvert} p_{S_{t-1} S_t}$	Soft-computing based approach using HMM [45]	Mixed Type data	1-order Markov chain modelling [45] for contextual anomaly detection.
Hawkins, 2002	$\delta_i = \frac{1}{n} \sum_{j=1}^{n} (x_{ij} - o_{ij})^2$	Soft-computing based approach in RNN [44]	Categorical data	One-class Anomaly Detection [44].
Ghoting, 2004	$Score_1(p_i) = \sum_{d \subseteq p_i} \left(\frac{1}{\lvert d \rvert} \lvert sup(d) \leq s \right)$	Distance-based approach	Categorical data	Capturing dependencies using links for categorical data in LOADED [42].
Ghoting, 2004	$Score_2(p_i) = \sum_{d \subseteq p_i} \left(\frac{1}{\lvert d \rvert} \lvert (C1 \vee C2) \wedge C3 \ is \ true \right)$	Distance-based approach	Mixed type data	Handling mixed attribute data in LOADED [42].
Otey, 2005	$Anomaly\ Score[P_i] = \left(\frac{\sum_{j=1}^{m} i - W_j}{i} \right) / m + \frac{V_\tau}{mn^2}$	Distance-based approach	Mixed type data	Discriminate outliers for categorical and continuous attributes in RELOADED [43].
Song, 2007	$Anomaly\ Score = \sum_{i=1}^{n_U} p(x \in U_i) \sum_{j=1}^{n_V} p(y \in V_j)\ p(V_j \lvert U_i)$	Soft-computing based approach in GMM-CAD [36]	Mixed type data	Reduction of contextual anomaly to point anomaly [36].
Koufakou, 2010	$Score_1(X_i) = \sum_{d \subseteq X_i \wedge supp(d) < \sigma \wedge \lvert d \rvert \leq Max} \frac{1}{supp(d) \times \lvert d \rvert}$	Density-based approach	Categorical data	Categorical score finding in ODMAD [27] for categorical data.
Koufakou, 2010	$Score_2(X_i) = \frac{1}{\lvert a \in X_i^c \rvert} \times \sum_{\forall a \in X_i^c} cos\left(X_i^q, \mu_a\right)$	Density-based approach	Mixed type data	Continuous score finding in ODMAD [27] for mixed attribute data.

normal or abnormal. These approaches can be used for on-line classification, where the classifier learns the classification model and then classifies new exemplars as and when required against the learned model. If the new exemplar lies in a region of normality it is classified as normal, otherwise it is flagged as an outlier. Classification algorithms require a good spread of both normal and abnormal labelled data.

4.1.1. Statistical Methods

The general approach to solve the outlier detection problem using statistical methods is based on the construction of probabilistic data models and the use of mathematical methods of applied statistics and probability theory. With a statistical approach to outlier detection, a system learns the behavior of users, applying metrics or measuring methods. As the system is running, the outlier or anomaly detector is constantly measuring the deviation of the present behavior profile from the original. The LNKnet software package was developed to simplify the application of most important statistical, neural network, and machine learning pattern classifier [51] on network connection data.

Methods for detecting outliers based on the regression analysis are included among statistical methods. Regression analysis consists of finding a dependence of one random variable (or a group of variables) Y on another variable (or a group of variables) X. Specifically, the problem is formulated as that of examining the conditional probability distribution $Y|X$. Among regression methods for outlier analysis, two approaches are distinguished. In the framework of the first approach, the regression model is constructed with the use of all data; then, the objects with the greatest errors are successively, or simultaneously, excluded from the model. This approach is called a *reverse search*. The second approach consists of constructing a model based on a part of data and, then, adding new objects followed by the reconstruction of the model. Such a method is referred to as a *direct search* [52]. The model is extended by adding the most appropriate objects, which are the objects with least deviations from the model constructed. The objects added to the model in the last turn are considered outliers. A regression method based on support vector regression (SVR) and particle swarm optimization algorithm (PSOA) is given in [53] for pattern analysis of intrusion detection. Basic disadvantages of the regression methods are that they greatly depend on assumptions about the error distribution and need a prior partition of variables into independent and dependent ones.

4.1.2. Decision Tree based Approaches

These approaches begin with a set of cases or examples, and create a tree data structure that can be used to classify new cases. Each case is described by a set of attributes (or features) which can have numeric or symbolic values. Associated with each training case is a label representing the name of a class. Each internal node of the tree contains a test, the result of which is used to decide what branch to follow from that node.

For example, a test might ask *is x > 4 for attribute x* ? If the test is true, then the case processes down the left branch, and if not then it follows the right branch. The leaf nodes contain class labels instead of tests. In classification mode, when a test case (which has no label) reaches a leaf node, a decision tree method such as C4.5 [54] classifies it using the label stored there.

Decision tree learners use a method known as divide and conquer to construct a suitable tree from a training set. The divide and conquer algorithm partitions the data until every leaf contains cases of a single class, or until further partitioning is impossible because two cases have the same values for each attribute but belong to different classes. Consequently, if there are no conflicting cases, the decision tree will correctly classify all training cases. This so-called *overfitting* is generally thought to lead to a loss of predictive accuracy in most applications.

Overfitting can be avoided by a stopping criterion that prevents some sets of training cases from being subdivided (usually on the basis of a statistical test of the significance of the best test), or by removing some of the structure of the decision tree after it has been produced.

The possibilistic decision tree [55] has been used for a intrusion detection system. Traffic data is captured by performing simulated attacks on Intelligent Electronic Devices (IEDs). Data is obtained for two types of genuine user activity and two types of common malicious attacks on IEDs. The genuine user activity includes, casual browsing of IED data and downloading of IED data while a Ping flood Denial of Service (DoS) and password crack attack are performed for malicious attacks. Classification is done using possibilistic decision trees for the logarithmic histogram of the time difference between the arrival of two consecutive packets. It obtains a continuous valued possibilistic decision tree and its cut points. It also includes the use of mean distance metrics to obtain the possibility distribution for the real attack data.

4.1.3. Soft computing Method - Roughset Approach

The rough set [29] philosophy is based on the assumption that with every object of the universe there is associated a certain amount of information (data, knowledge), expressed by means of its attributes. Objects having the same description are indiscernible. The basic idea of rough sets is discussed in *Sub-section 2.3.*

In [6], RST (Rough Set Theory) and SVM (Support Vector Machine) are used to detect intrusions. First, RST is used to preprocess the data and to reduce the dimensions. Next, the features selected by RST are sent to the SVM model to learn and test respectively. The method is effective in decreasing the space density of data and has low false positive rate and good accuracy of detection. The major advantage of rough set theory is that it does not need any preliminary or additional information about data, such as a probability distribution. The main problems that can be solved using rough set theory include data reduction (i.e., elimination of superfluous data), discovery of data dependencies, estimation of data significance, generation of decision (control) algorithms from data, approximate classification of data, discovery of similarities or differences in data, discovery of patterns in data, and discovery of cause-effect relationships.

4.1.4. Proximity-based Approaches

Proximity-based techniques are simple to implement and make no prior assumptions about the data distribution model. However, they suffer high computational cost as they are founded on the calculation of the distances between all records. The computational complexity is directly proportional to both the dimensionality of the data m and the number of records n. The k-nearest neighbour (k-NN) algorithm is suitable for outlier detection; it calculates the nearest neighbours of a record using a suitable distance calculation metric such as Euclidean distance or Mahalanobis distance. A definition of proximity based outlier $D^k(p)$ from [24] is discussed in *Definition 4* in *Sub-section 2.1.*

A partition-based outlier detection algorithm first partitions the input points using a clustering algorithm, and computes lower and upper bounds on D^k for points in each partition. It then uses this information to identify the partitions that cannot possibly contain the top n outliers and prunes them. Outliers are then computed from the remaining points (belonging to unpruned partitions) in a final phase.

PAIDS (Proximity assisted intrusion detection) [56], is an approach for identifying the outbreak of unknown worms. PAIDS does not rely on signatures. Instead, it takes advantage of the proximity information of compromised hosts. PAIDS operates on an orthogonal dimension with existing IDS approaches and can thus work collaboratively with existing IDSs to achieve better performance. The effectiveness of PAIDS with trace-driven simulations has a high detection rate and a low false positive rate.

The major limitation of this approach is selectiing an appropriate proximity measure, especially for the high dimensional mixed type data. Also, developing a heuristic method for appropriate threshold selection is a difficult task.

4.1.5. Kernel Function based approach

As has been found in the case of distance-based methods, defining an appropriate proximity measure for heterogeneously structured data is a difficult task. To overcome this limitation, methods based on kernel functions can be used [57]. The kernel of a function

f is the equivalence relation on the function's domain that roughly expresses the idea of equivalence as far as the function f can tell.

Definition 6: Let X and Y be sets and let f be a function from X to Y. Elements x_1 and x_2 of X are equivalent if $f(x_1)$ and $f(x_2)$ are equal, i.e., they are the same element of Y [58]. Formally: $f : X \rightarrow Y$

$$ker(f) = f\left\{(x_1, x_2) \in X \times X : f(x_1) = f(x_2)\right\}. \quad (18)$$

The kernel function $K(x, y)$ can be expressed as a dot product in a high dimensional space. If the arguments to the kernel are in a measurable space X, and if the kernel is positive semi-definite, i.e.,

$$\sum_{i,j} K(x_i, x_j)c_i c_j \geq 0$$

for any finite subset $f(x_1, \ldots, x_n)$ of X and subset $f(c_1, \ldots, c_n)$ of objects, there exists a function $\varphi(x)$ whose range is in an inner product space of possibly high dimension, such that

$$K(x, y) = \varphi(x) \cdot \varphi(y). \quad (19)$$

The KPCA (Kernel Principal Component Analysis) [59] is a real time IDS. It is composed of two parts. First part is used for on-line feature extraction. The second part is used for classification. Extracted features are used as input for classification. With an adaptation of the kernel function *kernel-trick* (*Equation 19*) KPCA extracts on-line non-linear features. Here, Least Squares Support Vector Machines (LS-SVM) [60] is used as a classifier. SVMs typically solve problems by quadratic programming (QP). Solving QP problem requires complicated computational effort and has high memory requirement. LS-SVM overcomes by solving a set of linear equations.

4.1.6. Kernel Function using Fuzzy Approach

This is a fuzzy clustering method in the feature space for the multi-class classification problem. The approach [14] searches for one common cluster containing images of all objects from the original space. In this case, the membership degree of an object image with respect to the fuzzy cluster in the feature space may be viewed as a typicalness degree of the object, i.e., a measure opposed to outlierness. Objects with a low typicalness degree (less than a threshold determined by the user) are considered outliers. It should be noted that the modification of the threshold (i.e., the modification of the outlier factor criterion) does not require model reconstruction, which is the case when the distance-based algorithms are used.

Petrovskiy introduces a fuzzy kernel-based method for real-time network intrusion detection [61]. It involves a kernel-based fuzzy clustering technique. Here, network audit records are vectors with numeric and nominal attributes. These vectors are implicitly mapped by means of a special kernel function into a high dimensional feature space, where the possibilistic clustering algorithm is applied to calculate the measure of typicalness and to discover outliers.

4.1.7. Distance-based outlier detection approach

This approach trains classifiers and computes covariance matrices incrementally. Therefore, the decision whether a given point is an anomaly or not is based only on the previously processed data points. For example, in the *RELOADED* algorithm [43] for each point in the data set, and for each categorical attribute d of that data point, an appropriate classifier is trained. That classifier, in turn, is used to predict the appropriate value of d. If the prediction is wrong, the count of incorrect predictions is incremented. Next, continuous attributes of the data point are used to incrementally compute the covariance matrix corresponding to the attribute-value pair d. The cumulative violation score of the data point is incremented. An anomaly score for *RELOADED* is given in *Equation 9*.

VAHD (Variable-length Average Hamming Distance) [62] is a distance-based anomaly detection method. The method operates in two stages (i) building up a normal variable-length pattern database and (ii) detecting such pattern(s) in real environment. In the first stage, it builds a normal profile by collecting system calls of some interested process(es) and by extracting interested patterns. In the second stage, this profile is used to monitor system behavior and calculate the average Hamming distance (AHD) between them, which determines the strength of an anomalous signal. If it exceeds a user defined threshold value, a suspicious event is assumed. The method has some advantages, such as high accuracy and real-time detection.

One needs to choose a threshold for anomaly score in order to discriminate between outliers and normal points. This can be done by incrementally computing the mean and standard deviation of the anomaly scores and by flagging any point as an outlier if it is more than s standard deviations greater than the current mean. Here, deriving the standard deviation is difficult. Also, estimating distance over a combined numeric and categorical attribute domain with proper weightage is a difficult task.

4.1.8. Signal processing based approach

Signal processing techniques can be applied to identify network anomalies, and to study network characteristics such as routing and congestion. There are two signal processing based approaches: wavelet based approach and cognitive packet network based approach.

A Wavelet based approach

In a wireless sensor network (WSN), a large number of sensor nodes are distributed over a large area. The sensor nodes are endowed with wireless communication capabilities for sensing and processing. A measurement that significantly deviates from the normal pattern of sensed data detects an outlier in WSN. In [17], a detailed overview of the existing outlier detection techniques for the WSN is given. A survey of wavelet based network anomaly detection approaches in the context of WSN can be found in [20]. In recent research, another significant work [63] can be found for anomaly detection by identifying outlier based on wavelets. In addition, research reported in [64, 65, 66, 67, 68] present significant work dealing with automatic network response to unexpected events and improvement in quality of service (QoS).

B Cognitive packet network based approach

The cognitive packet network (CPN) architecture uses an adaptive routing protocol that attempts to address the stability and reliability of network services by rerouting their traffic as necessary. Network worms are self-replicating and self-propagating malicious applications. They can exploit system vulnerabilities of operating systems and spread through networks causing significant damage by reducing system performance. Worms can be considered anomalies in network traffic. Research related to detection of attacks, particularly those by network worms, using CPN is found in [69, 70, 71, 72, 73]. Recent research [74, 75] also report on the development of self-aware computer networks (SAN) based on CPN. Such networks are capable of detecting and reacting to intrusions adaptively.

4.1.9. Density-based outlier detection approach

A density-based approach uses an outlier factor as a measurement of being an outlier. In the LOF [28] algorithm, a local outlier factor (LOF) is used as a measurement for finding a sample as outlier. This local outlier factor is computed from the sample's nearest neighbour objects rather than from the entire set of data as a whole. LOF is the mean value of the ratio of the density distribution estimate in the neighbourhood of the object analysed to the distribution densities of its neighbours. LOF is computed using *Equation 2*.

Another density-based anomaly detection method is found in [76]. An important advantage of this method is its capability to update normal profile of system usage pattern dynamically. It models the system usage pattern based on features of program behavior. When system usage pattern changes, new program behaviours are inserted into old profiles by density-based incremental clustering. It uses DBSCAN [77] to generate the initial clusters for normal program behavior profiles. The profiles are updated by modifying DBSCAN's clusters using incremental clustering. This method has incremental detection quality and a much lower false alarm rate.

4.2. Unsupervised Approaches

These approaches determine outliers with no prior knowledge of the data. They use a learning approach analogous to unsupervised clustering. The approaches process the data as static distributions, pinpoint the most remote points, and flag them as potential outliers. Once a system possesses a sufficiently large dataset with good coverage, it can compare new items with the existing data.

4.2.1. Statistical Method

These approaches are developed from the view point of statistical learning theory. They attempt to detect outliers in an on-line process through the on-line unsupervised learning of a probabilistic model of the information source. A score is given to an input based on the learned model, with a high score indicating a high possibility of being a statistical outlier. An off-line process of outlier detection uses batch-detection in which outliers can be detected only after seeing the entire dataset. The on-line setting is more realistic than the off-line one when one deals with the tremendous amount of data in network monitoring. An example of such a method is SmartSifter (SS) [78]. SS is able to detect 79% intrusions in the top 3%, and 81% intrusions in the top 5% of the KDDCUP 1999 dataset.

4.2.2. Graph Theoretic approach

These approaches generate many classification trees from the original data using a tree classification algorithm. After the forest of trees is formed, a new object that needs to be classified is sent down each of the trees in the forest for classification. It finds outliers whose proximities to all other cases in the entire data are generally small. An example is the random forests [21] algorithm. The random forests algorithm generates many classification trees from the original data. If cases k and n are in the same leaf of a tree, their proximity is increased by one. Finally, the proximities are normalized by dividing by the number of trees.

The random forests algorithm, outliers can be defined as the cases whose proximities to other cases in the dataset are generally small. Outlierness can be calculated over proximities. $class(k) = j$ denotes that k belongs to class j. $prox(n, k)$ denotes the proximity between cases n and k. The average proximity from case n in class j to case k (the rest of data in class j) is computed as:

$$\bar{P}(n) = \sum_{class(k)=j} prox^2(n, k). \tag{20}$$

The raw outlierness of case n is defined as: $N/\bar{P}(n)$, where N denotes the number of cases in the dataset. In each class, the median and the standard deviations of all raw outlierness values are calculated. The median is subtracted from each raw outlierness value. The result of the subtraction is divided by the standard deviation to get the final outlierness. If the outlierness of a case is large, the proximity is small, and the case is determined an outlier.

The random forests algorithm provides relatively higher detection rates where the false positive rates are low on KDDCUP 1999 dataset. For example, the detection rate is 95% when the false positive rate is 1%. When the false positive rate is reduced to 0.1%, the detection rate is still over 60%.

4.2.3. Clustering

These approaches attempt to detect both either single point outliers or cluster-based outliers, and can assign each outlier a degree of being an outlier. *LDBSCAN (local-density-based spatial clustering of applications with noise)* [79] is a cluster-based outlier detection algorithm.

LDBSCAN randomly selects one core point which has not been clustered, and then retrieves all points that are local density reachable from the chosen core point to form a cluster. It does not stop until there is no unclustered core point.

Cluster-based outlier detection has been applied to the Backbone Anomaly Detection System for CSTNET, an Internet service provider for all the institutes of Chinese Academy of Sciences [79]. The Backbone Anomaly Detection System continuously monitors the input and output throughput of about 300 network nodes of CSTNET. Each node generates its average throughput record every five minutes, so the Backbone Anomaly Detection System checks the node state $300 \times 12 = 3600$ times each hour. Under normal circumstances, the throughput of a certain node forms a cluster. But during the period of an abnormal event, the throughput exhibits temporal locality, i.e., it forms a new cluster which is different from history. Using cluster-based outlier detection, the system generates 10 alerts per hour. According to the feedback of the network administrators in CSTNET, cluster-based outlier detection generates accurate alerts.

We compare the outlier detection approaches discussed in the previous subsections based on parameters such as detection approach (supervised or unsupervised), methods (statistical, proximity based, kernel function), input data (training data required or not), proximity measure used (e.g., Euclidean distance), parametric or not, attribute types of data. See *Table 3* for how the methods compare against one another. Supervised methods require training data, but unsupervised methods do not. Except a few, most supervised methods use numeric data. However, unsupervised methods are capable of handling mixed type high dimensional data. All the methods in *Table 3* can handle high dimensional data.

5. RESEARCH ISSUES AND CHALLENGES

Outlier detection is an extremely important problem with direct application in various domains. It involves exploring unseen spaces. A key observation in outlier detection is that it is not a well-formulated problem. The nature of the data, the nature of the outliers, the constraints and the assumptions collectively constitute the problem formulation. Some outlier detection techniques are developed in a more generic fashion and can be ported to various application domains while others directly target a particular application domain. In many cases, the data structures used for faster detection, the proximity measure or anomaly detection formula used and the capability of handling higher dimensional data (may be mixed type) for any distribution pattern dictate which method outperforms the others. In outlier detection, the developer should select an algorithm that is suitable for their data set in terms of the correct distribution model, the correct attribute types, the scalability, the speed, any desired incremental capabilities to allow new exemplars to be handled and the modelling accuracy. The developer should also consider which of the fundamental approaches is suitable for their problem.

The distance-based techniques do not make assumptions about the data since they compute the distance between each pair of points. The distance measuring techniques can be based on numeric, categorical or mixed type data. It is difficult to have a single proximity measure that can handle the numeric, categorical or mixed type attribute for any dimensionality and for any number of instances. Datasets that consist of one type of attribute, i.e., only numerical attributes or categorical attributes can be directly mapped into numerical values. However, the mapping of categorical attributes to numerical attributes is not a straightforward process and greatly depends on the mapping used. On the other hand, density-based methods estimate the density distribution of the data points based on attributes or parameters and then identify outliers as those lying in regions of low density. Like distance-based methods, the design of an appropriate density-based outlier detection method is a challenging task. Density-based methods are also based on distance computations which can be inappropriate for categorical data, and again not straightforward. In addition, high-dimensional data is almost always sparse, which creates problems for density-based methods. If indiscernible or indistinguishable objects occur, rough set techniques can provide good solutions for outlier detection. For objects with uncertainties, fuzzy-rough techniques may be suitable for outlier detection. However, soft computing [80] based approaches allow tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost.

TABLE 3. Outlier Detection Methods: A General Comparison

Approach	Author, Year	Methods Used	Method Class	Training Dataset	Proximity Measure	Parametric/ nonParametric/Both	Numeric/ Categorical /Mixed type
Supervised	Hadi, 1992	Regression Analysis [52]	Statistical	Training Data	No	Parametric	Numeric
	Ramaswamy, 2000	Partition based [24]	Proximity based	No	Euclidean/ Mahalanobis	No	Numeric
	Petrovskiy, 2003	Fuzzy Approach [14]	Kernel Function	No	Membership Degree	nonParametric	Numeric
	Pawlak, 1995	Roughset [29]	Soft computing	Training Data	Membership Function	nonParametric	Numeric
	Quinlan, 1993	C4.5 [54]	Decision Tree	Training Data	Information entropy	Parametric	Numeric
	Matthew, 2005	RELOADED [43]	distance-based	Training Data	Euclidean Distance	nonParametric	Mixed Type
	Kriegel, 2000	LOF [28]	Density-based	Training Data	LOF	Parametric	Mixed Type
Unsupervised	Kenji, 2004	SmartSifter [78]	Statistical	No	No	Both	Mixed
	Zhang, 2006	Random forests [21]	Graph Theoretic approach	No	Gini Index	Parametric	Mixed
	Duan, 2008	LDBSCAN [79]	Clustering	No	Euclidean distance	nonParametric	Numeric

In addition, for all types of attacks, all features are not equally predictive [81]. Considering various factors, one can conclude that a combined approach based on distance, density or soft computing, can provide required robustness and scalability for outlier detection. Here, pre-processing of the data set is essential to identify responsible features among categorical attributes. For continuous attributes, a distance-based approach is suitable, but the right threshold value is needed for differentiation of data points. Thus, a faster incremental method capable of handling high dimensional mixed type data with high detection rate and reduced false positives is still called for. The major issues in outlier detection are as follows.

- Defining a normal region which encompasses every possible normal behavior is very difficult. Oftentimes normal behavior evolves over time and an existing notion of normal behavior may not be sufficiently representative in the future.
- 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.
- Often the data contains noise similar to the actual outliers and hence is difficult to distinguish and remove.
- Availability of labelled data for training/validation is often a major issue when developing an outlier detection technique.
- Typically, soft computing embraces several computational intelligence methodologies, including artificial neural networks, fuzzy logic, evolutionary computation, and probabilistic computing. These methods neither are independent of one another nor compete with one another. Rather, they work in a co-operative and complementary way. In this context, establishing an appropriate soft computing method for outlier detection is a challenging task.
- Nowadays, most network attacks are distributed. To handle such attacks a distributed outlier detection technique is essential.
- Among existing outlier detection approaches, none is capable of handling the outlier detection problem individually to a satisfactory level. Especially, to achieve a high detection rate with a low false positive alarm rate for the network domain, a cost effective ensemble approach may be more suitable.
- Several scoring functions have been proposed over the decades for numeric, categorical or mixed type data. However, the design of an appropriate scoring function that can handle all these types of data in the presence of noise still remains a challenge.
- Often, it has been observed that supervised or unsupervised anomaly detection based on clustering approaches alone cannot handle all network traffic data effectively. In such cases, for time window based real time attack detection, integration of an appropriate outlier detection technique, either in a post processing task or in support of simultaneous processing, may be useful.

6. CONCLUSION

This paper has attempted to establish the significance of outlier detection in anomaly identification. A comprehensive survey of various distance-based, density-based and soft computing based outlier detection techniques has been provided in this paper. In addition, we report on and analyse various outlier detection techniques under supervised and unsupervised approaches. Based on our review, we observe that the notion of outlier is different for different application domains. Thus, development of an effective outlier detection technique for mixed-type and evolving network traffic data, especially in the presence of noise, is a challenging task. However, for future research, outlier detection method should be tested on real network data collected using tools such as flow-tools [82] and dataset like the MITRE [50] data set.

REFERENCES

[1] H Liu, W. J., S Shah (2004) On-line outlier detection and data cleaning. *Computers and Chemical Engineering*, **28**, 1635–1647.

[2] Mitchell, T. M. (1997) *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA.

[3] Vapnik, V. N. (1995) *The nature of statistical learning theory*. Springer-Verlag, New York, USA.

[4] Gelenbe, E. (2007) Dealing with software viruses: A biological paradigm. *Information Security Technical Report*, **12(4)**, 242–250.

[5] Heady, R., Luger, G., Maccabe, A., and Servilla, M. (1990) The architecture of a network level intrusion detection system. Technical Report NM 87131. Computer Science Department, University of New Mexico, Albuquerque Mexico.

[6] Chen, R. C., Cheng, K. F., and Hsieh, C. F. (2009) Using rough set and support vector machine for network intrusion detection. *International Journal of Network Security & Its Applications (IJNSA)*, **1(1)**, 1–13.

[7] Roesch, M. (1999) Snort-lightweight intrusion detection for networks. *Proceedings of the 13th Conference on Systems Administration (LISA-99), Seatlle, WA, USA November 7-12*, pp. 229–238. USENIX, Seattle, Washington.

[8] Daniel, B., Julia, C., Sushil, J., and Ningning, W. (2001) Adam: a testbed for exploring the use of data mining in intrusion detection. *SIGMOD Rec.*, **30**, 15–24.

[9] Lee, W. and Stolfo, S. J. (1998) Data mining approaches for intrusion detection. *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, San Antonio, Texas, USA, Jan., pp. 6–6. USENIX.

[10] Chandola, V., Banerjee, A., and Kumar, V. (2009) Anomaly detection : A survey. *ACM Computing Surveys (CSUR)*, **41**, 15:1–58.

[11] Padmanabhan, J. and Easwarakumar, K. S. (2009) Traffic engineering based attack detection in active networks. *Lecture Notes in Computer Science (LNCS)*, **5408**, 181–186.

[12] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., and Srivastava, J. (2003) A comparative study of anomaly detection schemes in network intrusion detection. *Proceedings of the 3rd SIAM International Conference on Data mining*, San Francisco, CA, May 1-3, 2003, pp. 25–36. SIAM.

[13] Hodge, V. and Austin, J. (2004) A survey of outlier detection methodologies. *Artificial Intelligence*, **22**, 85–126.

[14] Petrovskiy, M. I. (2003) Outlier detection algorithms in data mining systems. *Programming and Computer Software*, **29**, 228–237.

[15] 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**, 45–84.

[16] Chandula, V., Banerjee, A., and Kumar, V. (2007) Outlier detection: A survey. Technical Report TR 07-017. Dept of CSE,University of Minnesota, USA.

[17] Zhang, Y., Meratnia, N., and Havinga, P. (2010) Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Survey & Tutorials*, **12(2)**, 159 – 170.

[18] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2011) Rodd: An effective reference based outlier detection technique for large datasets. *LNCS-CCIS*, **133, Part I**, 76–84.

[19] Ng, B. (2006) Survey of anomaly detection methods. Technical Report UCRL-TR-225264. Lawrence Livermore National Laboratory, University of California, California USA.

[20] Kaur, G., Saxena, V., and Gupta, J. P. (2010) Anomaly detection in network traffic and role of wavelets. *Proc. of 2nd IEEE International Conference on Computer Engineering and Technology (ICCET 2010)*, Chengdu, China, 16-18 April, pp. V7: 46–51. IEEE.

[21] Zhang, J. and Zulkernine, M. (2006) Anomaly based network intrusion detection with unsupervised outlier detection. *IEEE International Conference on Communications (ICC)*, June, pp. 2388–2393. IEEE Xplore, Istanbul.

[22] Hawkins, D. (1980) *Identification of outliers*. Chapman and Hall, London.

[23] Knorr, E. M. and Ng, R. T. (1998) Algorithms for mining distance-based outliers in large datasets. *Proceedings of the 24th Int'l. Conf. on Very Large Data Bases*, New York USA, Sep., pp. 392–403. Morgan Kaufmann.

[24] Ramaswamy, S., Rastogi, R., and Shim, K. (2000) Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, **29**, 427–438.

[25] Knorr, E. M. and Ng, R. T. (1999) Finding intensional knowledge of distance-based outliers. *Proceedings of the 25th International Conference on Very Large Data Bases, VLDB'99*, Edinburgh, Scotland, UK, 7-10 Sep., pp. 211–222. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[26] Breunig, M. M., Kriegel, H., Ng, R. T., and Sander, J. (2000) Lof: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD international conference on management of data*, Dallas, Texas, United States, May, pp. 93–104. New York: ACM Press.

[27] Koufakou, A. and Georgiopoulos, M. (2010) A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. *Data Mining and Knowledge Discovery*, **20**, 259–289.

[28] Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J. (2000) Lof: Identifying density-based local outliers. *ACM SIGMOD*, **29**, 93–104.

[29] Pawlak, Z., Grzymala-Busse, J., and Ziarko, W. (1995) Rough sets. *Communications of the ACM*, **38**, 88–95.

[30] Jiangab, F., Suia, Y., and Caoa, C. (2008) A rough set approach to outlier detection. *International Journal of General Systems*, **37**, 519–536.

[31] Sarkar, M. and Yegnanarayana, B. (1998) Fuzzy-rough membership functions. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, San Diego, CA , USA, Oct., pp. 2028–2033. IEEE Xplore.

[32] Bezdek, J. C. and Pal, S. K. (1992) *Fuzzy Model for Pattern Recognition*. Eds. IEEE Press, Newyork USA.

[33] Duboia, D. and Prade, H. (1990) Rough-fuzzy sets and fuzzy-rough sets. *International Journal of General Systems*, **17**, 191–209.

[34] Barnett, V. and Lewis, T. (1994) *Outliers in Statistical Data*. John Wiley, Chichester, New York, USA.

[35] Teodoro, P. G., Verdejo, J. D., Fernandez, G. M., and Vazquez, E. (2009) Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computer and Security*, **28**, 18–28.

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

[37] Tavallaee, M., Stakhanova, N., and Ghorbani, A. A. (2010) Towards credible evaluation of anomaly-based intrusion detection methods. *IEEE Transactions on System, Man, and Cybernetics Part C: Applications and Reviews*, **40(5)**, 516–524.

[38] Cha, S.-H. (2007) Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Science*, **1 (4)**, 300–307.

[39] Boriah, S., Chandola, V., and Kumar, V. (2008) Similarity measures for categorical data: A comparative evaluation. *Proceedings of the 8th SIAM International Conference on Data Mining*, Atlanta, Georgia, USA, Apr., pp. 243–254. Society for Industrial and Applied Mathematics(SIAM).

[40] Settles, B. (2009) Active learning literature survey. Computer Sciences Technical Report 1648. University of Wisconsin–Madison.

[41] Lippman, R. P., Fried, D. J., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Wyschogrod, S. W. D., Cunningham, R. K., and Zissman, M. A. (2000) Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. *Proceedings of DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00*, Hilton Head, SC , USA, 25-27 Jan., pp. 12–26 Vol 2. IEEE Xplore.

[42] Ghoting, A., Otey, M. E., and Parthasarathy, S. (2004) Loaded: Link-based outlier and anomaly detection in evolving data sets. *Proceedings of the 4th IEEE International Conference on Data Mining*, Brighton, UK, Nov., pp. 387–390. IEEE Computer Society.

[43] Otey, M. E., Parthasarathy, S., and Ghoting, A. (2005) Fast lightweight outlier detection in mixed-attribute data. Technical Report OSU-CISRC-6/05-TR43. Department of Computer Science and Engineering, The Ohio State University, Ohio, United States.

[44] Hawkins, S., He, H., Williams, G., and Baxter, R. (2002) Outlier detection using replicator neural networks. *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, London, UK, Sep., pp. 170–180. Springer-Verlag.

[45] Ye, N. (2000) A markov chain model of temporal behavior for anomaly detection. *Proceedings of the 2000 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, June, pp. 171–174. IEEE Xplore.

[46] Puketza, N., Chung, M., Olsson, A. R., and Mukherjee, B. (1997) A software platform for testing intrusion detection systems. *IEEE Software*, **14(5)**, 43–51.

[47] Debar, H., Dacier, M., Wespi, A., and Lampart, S. (1998) An experimentation workbench for intrusion detection systems. Technical report. RZ 2998(93044) Research Division, IBM, New York, NY.

[48] McHugh, J. (2000) Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security*, **3(4)**, 262–294.

[49] Durst, R., Champion, T., Witten, B., Miller, E., and Spagnuolo, L. (1999) Testing and evaluating computer intrusion detection systems. *Communication ACM*, **42(7)**, 53–61.

[50] Aguirre, S. J. and Hill, W. H. (1997) Intrusion detection fly-off: Implications for the united states navy. Technical report. Sept. 1997, MITRE, MTR 97W096 McLean, Virginia.

[51] Lippmann, R. and Kukolich, L. (1993). Lnknet user's guide. MIT Lincoln Laboratory.

[52] Hadi, A. S. (1992) A new measure of overall potential influence in linear regression. *Computational Statistics & Data Analysis*, **14**, 1–27.

[53] Tian, W. and Liu, J. (2009) Intrusion detection quantitative analysis with support vector regression and particle swarm optimization algorithm. *Proceedings of the 2009 International Conference on Wireless Networks ICWN'09*, Shanghai, China, 28-29 December, pp. 133–136. IEEE Xplore.

[54] SALZBERG, S. L. (1994) C4.5: Programs for machine learning. *Machine Learning*, **16**, 235–240.

[55] Premaratne, U., Ling, C., Samarabandu, J., and Sidhu, T. (2009) Possibilistic decision trees for intrusion detection in iec61850 automated substations. *Proceedings of the 2009 International Conference on Industrial and Information Systems (ICIIS)*, Sri Lanka, Dec., pp. 204–209. IEEE Xplore.

[56] Zhuang, Z., Li, Y., and Chen, Z. (2009) Paids:a proximity-assisted intrusion detection system for unidentified worms. *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*, Seattle,Washington, 20-24 July, pp. 392–399. IEEE.

[57] Scholkopf, B. and Smola, A. J. (2000) *Learning with Kernels*. The MIT Press, Cambridge, Massachusetts, London, England.

[58] Sewell, M. (2009). Kernel methods. Department of Computer Science, University College London.

[59] Kim, B. and Kim, I. (2006) Kernel based intrusion detection system. *Proceedings of the 4th Annual ACIS International Conference on Computer and Information Science (ICIS'05)*, Jeju Island, South Korea, 16-16 July, pp. 13–18. IEEE Computer Society.

[60] Suykens, J. A. K. and Vandewalle, J. (1999) Least squares support vector machine classifiers. *Neural Processing Letters*, **9**, 293–300.

[61] Petrovskiy, M. (2003) A fuzzy kernel-based method for real-time network intrusion detection. *Proceeding of the 3rd International Workshop of Innovative Internet Community Systems (IICS)*, Leipzig, Germany, 19-21 June, pp. 189–200. Springer.

[62] Du, Y., Zhang, R., and Guo, Y. (2010) A useful anomaly intrusion detection method using variable-length patterns and average hamming distance. *Journal of Computers*, **5(8)**, 1219–1226.

[63] Hey, L. and Gelenbe, E. (2009) Adaptive packet prioritisation for wireless sensor networks. *Procedings of Next Generation Internet Networks*, Aveiro, Portugal, 1-3 July, pp. 1–7. IEEE Xplore.

[64] Lent, R., Abdelrahman, O. H., Gorbil, G., and Gelenbe, E. (2010) Fast message dissemination for emergency communications. *Proceedings of PerCom Workshop on Pervasive Networks for Emergency Management (PerNEM'10)*, Mannheim, Germany, March 29-April 02 2010, pp. 370–375. IEEE, NY, USA.

[65] Ngai, E., Gelenbe, E., and Humber, G. (2009) Information-aware traffic reduction for wireless sensor networks. *Proceedings of the 34th Annual IEEE Conference on Local Computer Networks (LCN 2009) October 20-23, Zurich, Switzerland*, pp. 451–458. IEEE Zurich, Switzerland.

[66] Gelenbe, E. and Ngai, E. (2008) Adaptive qos routing for significant events in wireless sensor networks. *Proceedings of the 5th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS'08)*, Atlanta, GA, USA, 29 September- 2 October 2008, pp. 410–415. IEEE, New York, NY, USA.

[67] Gelenbe, E. and Ngai, E. (2010) Adaptive random re-routing for differentiated qos in sensor networks. *The Computer Journal*, **53(7)**, 1052–1061.

[68] Gelenbe, E. and Ngai, E. (2008) Adaptive random re-routing in sensor networks. *Proceedings of the Annual Conference of ITA (ACITA '08) September 16-18*, London, UK, pp. 348–349. Imperial College London, UK.

[69] Sakellari, G. and Gelenbe, E. (2010) Demonstrating cognitive packet network resilience to worm attacks. *Procedings of the 17th ACM Conference on Computer and Communications Security (CCS 2010)*, Chicago, IL, USA, 4-8 October 2010, pp. 636 – 638. ACM, New York, NY, USA.

[70] Sakellari, G. and Gelenbe, E. (2009) Adaptive resilience of the cognitive packet network in the presence of network worms. *Procedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management*, Bucharest, Romania, 11-12 May, pp. 16:1–16:14. NATO Research & Technology Organisation.

[71] Sakellari, G., Hey, L., and Gelenbe, E. (2008) Adaptability and failure resilience of the cognitive packet network. *Presented at the Demo Session of the INFOCOM2008*, Phoneix, AZ, USA, 15-17 April. IEEE, New York, NY, USA.

[72] Oke, G., Loukas, G., and Gelenbe, E. (2007) Detecting denial of service attacks with bayesian classifiers and the random neural network. *Procedings of Fuzz-IEEE 2007*, London, UK, 23-26 July, pp. 1964–1969. IEEE, New York, NY, USA.

[73] Gelenbe, E. and Loukas, G. (2007) A self-aware approach to denial of service defence. *Computer Networks*, **51(5)**, 1299–1314.

[74] Gelenbe, E. (2011) Self-aware networks. *McGraw-Hill 2011 Yearbook of Science & Technology*, **To appear**, Manuscript ID YB11–0175, 2011.

[75] Gelenbe, E. (July 2009) Steps towards self-aware networks. *Communications of the ACM*, **52(7)**, 66–75.

[76] Ren, F., Hu, L., Liang, H., Liu, X., and Ren, W. (2008) Using density-based incremental clustering for anomaly detection. *Proceedings of 2008 International Conference on Computer Science and Software Engineering*, Wuhan, Hubei, China, 12-14 December, pp. 986–989. IEEE computer society.

[77] Ester, M. and Kriegel, H. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on knowledge discovery and Data mining*, Portland, Oregon, August 2-4, pp. 226–231. AAAI Press.

[78] Yamanishi, K., ichi Takeuchi, J., 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.

[79] Duan, L., Xu, L., Liu, Y., and Lee, J. (2008) Cluster-based outlier detection. *Annals of Operations Research*, **168**, 151–168.

[80] Zadeh, L. A. (1994) Fuzzy logic, neural networks, and soft computing. *Communications, ACM*, **37**, 77–84.

[81] Kayacik, H. G., Heywood, A. N. Z., and Heywood, M. I. (2005) Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets. *Proceedings of the 3rd Annual Conference on Privacy, Security and Trust*, Halifax, NS, Canada, Oct. Dalhousie University.

[82] Staniford, S., Hoagland, J. A., and McAlerney, J. M. (2002) Practical automated detection of stealthy portscans. *Journal of Computer Security*, **10**, 105–136.