IJBT (2017) 01: 16-48

RESEARCH ARTICLE

Anomaly detection in banking operations

Chilukuri K. Mohan[1] · Kishan G. Mehrotra[1]

Abstract This paper presents an overview of anomaly detection algorithms and methodology, focusing on the context of banking operations applications. The main principles of anomaly detection are first presented, followed by listing some of the areas in banking that can benefit from anomaly detection. We then discuss traditional nearest-neighbor and clustering-based approaches. Time series and other sequential data analysis approaches are described. The problems posed by categorical data are also discussed, along with the methods proposed in the literature to address the same. The ensemble methods are presented, followed by mathematical perspectives on anomaly detection.

Keywords: Anomaly detection · Banking · Time-stamped data · Categorical data · Ensemble methods

1 Introduction

Banking operations include many daily, periodic, and aperiodic activities and transactions performed by or affecting numerous stakeholders such as employees, customers, debtors, and external entities. The complex nature of these activities and transactions necessitate constant monitoring to ensure that neither the bank nor its stakeholders are adversely affected by various events that may be malicious, random, or occurring due to inevitable business cycles. Events may unfold over time, and early detection can significantly ameliorate potential ill-effects, and in some cases actively prevent the same. This paper develops a framework for understanding and addressing such events using various anomaly detection algorithms.

Anomalies (or outliers) are variations from expected norms, or from prior data, or from predictions based on process models. Anomalies are contrasted with *inliers* or non-anomalous data. The variations from norms may be exhibited in terms of multiple data points or in patterns of data, rather than individual data points themselves. For instance, a customer's abnormal spending may be indicated by a collection of closely spaced purchases, rather than by a single purchase.

✉ Chilukuri K. Mohan
mohan@syr.edu

Kishan G. Mehrotra
mehrotra@syr.edu

[1] Department of Electrical Engineering and Computer Science, 4-206, Center for Science and Technology Syracuse University Syracuse, NY 13244-4100, USA

In some cases, norms may be derived from the knowledge of experts who have examined past data; in other cases, machine learning or data mining algorithms may have been applied to prior data. In the supervised learning context (also known as the signature-based approach), the data used to build the model is pre-characterized by experts (or prior observations), so that the anomaly detection task merely involves measuring the variation of new data from such models. However, many practical applications involve the unsupervised learning context, in which no prior labels conclusively distinguish anomalous data from normal data.

Unsupervised fraud detection approaches are particularly important since new fraudulent attacks are being invented every day and signature based techniques are unable to detect them. Researchers have also explored semi-supervised problems in which some data identified as belonging to the majority (normal) classes is available, although not all data is labelled.

Classical unsupervised anomaly detection algorithms focus on identifying anomalies from a finite set $\boldsymbol{D}$ of data points $\boldsymbol{x}_{\mathrm{i}}$ that are described using d numerical dimensions or attributes. First, we define a distance measure between two data points, such as the Euclidean distance measure,

$$d(x_i, x_j) = \frac{1}{d}\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2$$

Such a distance measure is often extended to define a related distance measure $d(x_i, S)$ between a data point $\boldsymbol{x}_i$ and a set $\boldsymbol{S}$ of data points, such as the average distance between $\boldsymbol{x}_i$ and all points in S. Often, though not always, the data is first clustered (using an algorithm such as the k-means algorithm), and S is chosen to be the cluster of points nearest to $\boldsymbol{x}_i$. This distance measure is then used to estimate an “outlierness” or *anomaly score* indicator α, such as the distance between a point and its k nearest neighbors, often linearly normalized to ensure that $0 \leq \alpha(\boldsymbol{x}_i) \leq 1$. A point $\boldsymbol{x}_i$ is then defined to be anomalous if $\alpha(\boldsymbol{x}_i)$ exceeds a predetermined threshold, or is among the m highest anomaly scores among points in S, if the goal is to find m such points.

There is a vast literature on such anomaly detection algorithms, as well as on clustering algorithms on which they are based. Markou and Singh [53, 54], Hodge and Jim [37], Agyemang *et al.* [5], Chandola *et al.*[17, 18] and Pimentel et al. [60] provide extensive surveys for particular domains from different point of views, but the focus is mostly on signature-based fraud detection techniques. Sabau [64, 8], Patcha *et al.* [58] also present various anomaly detection techniques based on supervised, unsupervised and clustering methods.

This paper focuses on anomaly detection problems that arise in the banking context, and develops a framework to understand the same, addressing how such problems differ from the classic anomaly detection problems described in the preceding paragraph. In particular, we focus on problems where:

- Some data attributes are categorical (making the formulation of numerical distance measures infeasible or awkward);
- Some domain knowledge is available (so that the problem is neither fully supervised nor unsupervised, not even semi-supervised in the formal sense);

- Data arrives over time (so that anomalousness must be judged in the context of immediately preceding data); and
- May be inherently heterogeneous (so that anomalies of different kinds may appear in the data).

We categorize anomaly detection problems depending on whether the data being examined is a set, a sequence, or a structure of inherently higher dimensionality, leading to three major categories:

1. Point anomaly: One or more anomalous points in a collection of data points;

2. Contextual anomaly: A point that may be anomalous with respect to its neighbors, or to other data points that share a context or some features; and

3. Collective anomaly: A collection of (similar) data points that behave differently from other collections in the entire dataset. An individual data point may not then be considered anomalous by itself, but is considered anomalous in conjunction with other data points in a collection.

The first category is addressed by traditional algorithms based on separating a single outlier data point from others, e.g., whether an item purchased by an accountholder is substantially different from anything purchased by him in the past. The second emerges in analyzing data that arrives over time, e.g., higher credit card expenses may not be anomalous if incurred during a festive month, but may be anomalous during a non-festive month. A collection of numerous credit card purchases during a single day may be considered anomalous, although each purchase by itself cannot be flagged as anomalous; this illustrates the third category, often ignored by the literature.

A major problem in anomaly detection is a lack of effective general purpose anomaly detection techniques because an anomaly detection technique in one domain may not be suitable for other domains; both the normal and abnormal behaviours vary from domain to domain. For example, the techniques used to finding out anomalies in stock exchange transactions may not work well for network traffic analysis, although they both make use of transaction data [45].

Section 2 discusses banking applications where anomaly detection is useful. Section 3 presents traditional anomaly detection approaches, particularly those based on clustering. Time series approaches are sketched in Section 4. Section 5 discusses anomaly detection in the context of categorical data. Ensemble approaches are discussed in Section 6. This is followed by simple mathematical formulations of the problems addressed, in Section 7. Concluding remarks in Section 8 summarizes the discussions.

2 Banking applications of anomaly detection

Many banking applications require the use of anomaly detection methodologies. For instance, credit card and mobile phone fraud occurrences include a variety of scams that steal data and wealth from individuals and organizations. To prevent the misuse of any account, it is necessary to detect any unusual usage pattern by monitoring the customer’s usage pattern and looking for any deviation from prior usage patterns.

Insurance fraud includes cases where criminals manipulate the claim processing system by submitting documents whose contents contain untruths. In insider trading, criminals make use of crucial information before it is made public, to make illicit profits and manipulate stock prices; early detection is important.

Three broad classes of applications can be distinguished: behavior tracking, situation awareness, and fraud. Most of this section, and the focus of most work in this area, is on fraud; we first describe the other two areas.

2.1 Behavior tracking

Customer behavior tracking can provide clues to assist marketing efforts, and to enable advising processes aimed at preventing deterioration of the financial status of any customer. Examples include cases where spending patterns of customers begin to diverge significantly from prior history, as well as information regarding purchase behaviors that may suggest increased probability of specific other purchases in the future, perhaps signaling changes in lifestyle.

Employee behavior tracking can be helpful to identify productive and unproductive patterns of activity, which can directly lead to reinforcing or corrective actions by managers or decision-makers.

In these contexts, we may seek to identify outliers of various kinds, comparing individual behaviors to their own past (long-term) behavior, to the behavior of the peer group to which they belong, and when compared to larger populations of individuals (e.g., all customers, or all employees in a geographical region).

2.2 Situation awareness

Tracking external financial parameters and their changes over time can alert banks about how they can expect their customers and employees to behave in the future. For example, sudden changes in market indicators may lead to predictions about expected withdrawals or spending by customers. Similarly, anomalous events in overseas markets or unexpected fluctuations in currency markets may be expected to influence some day-to-day activities by large institutions that conduct business with a bank.

2.3 Fraud

“Fraud” is a very broad term that is used to describe many kinds of activities, distinguished by the nature of the data and the approaches needed to detect and prevent such activities. Broadly, such activities may be classified into three groups of activities, based on the perpetrators of the same: clients/customers and their representatives, employees/officers of the bank or financial institution, and outsiders.

1. Fraudulent activities by customers or clients include money-laundering, trading in illegal goods, and deliberate attempts to mislead the bank or others *via* multiple transactions. In some cases, an account may be administered by an authorized individual who undertakes a series of transactions resulting in a loss for the primary account-holder or beneficiary. Analysis of transaction sequences over time (and compared to patterns for other accounts) can reveal anomalies that require careful analysis and further investigation.

2. Bank employees have access to enormous amounts of data and are trusted to take care of their customers' accounts with the highest standards of ethics. Unfortunately, some cases have emerged in the past where employees have exploited internal knowledge of the system to obtain personal benefits, or to indulge in risky trading behavior contrary to the mission and principles of their employer. Many such cases can be detected by looking for anomalies in the behaviors (over time) of employees authorized to access customer accounts.
3. Most serious are cases of purchases and cash withdrawals by unauthorized individuals, and by illegal trading of customer information, violating privacy and exposing customers to increased risk. Unsurprisingly, these have attracted the greatest attention in the industry and the research community. In particular, examples of credit card fraud (and data compromises) are considered to be prototypical of situations in which anomaly detection algorithms can help, and we discuss this example in greater detail below.

   When a debit or credit card is illegally used by an outsider, the nature of the purchase (amount, location, timing, etc.) can be compared to other purchases by the legitimate credit card user and flagged as an anomaly. Sometimes, the anomaly may consist in identifying a sudden increase in the (short-term) frequency of purchases (e.g., a flurry of successive transactions, none of which is a large amount). The nature of the item purchased may also signify increased probability of fraud; e.g., online purchases of electronics and expensive clothes may indicate fraud, if the nature of the items purchased is substantially different from prior purchases.

Irrespective of which anomaly detection algorithm is applied, there is always uncertainty in the decision-making, with both false positives (legitimate transactions being flagged as fraudulent) and false negatives (fraudulent transactions that are not caught by the anomaly detection algorithm) occurring with some probability. The direct cost involved in false negatives is easily measurable (the dollar amount), along with the potential for future fraudulent transactions with the same account: catching the first fraudulent transaction can help prevent others. With false positives, on the other hand, we must consider other intangible costs: most importantly, unpleasant interactions with the customer may ensue, with reduced customer satisfaction and potential loss of future business with that customer, as well as members of their organizations and social circles.

3 Principles of traditional anomaly detection

Traditional anomaly detection algorithms are formulated as solutions to the problem of identifying a small subset $\boldsymbol{s}$ of data points from a given set $\boldsymbol{D}$, such that points in $\boldsymbol{s}$ are significantly different from those in $\boldsymbol{D}\backslash\boldsymbol{s}$ (i.e., points in $\boldsymbol{D}$ but not in $\boldsymbol{s}$). Abstractly, the problem of finding m most anomalous data points in a numerical multi-dimensional data set $\boldsymbol{D}$, with a specific distance measure, $\boldsymbol{d}$, may be viewed as an optimization problem with the goal of finding $\boldsymbol{s}$, a subset of $\boldsymbol{D}$ such that $|\boldsymbol{s}|=m$, maximizing;

$$\sum_{x \in s} \min(\boldsymbol{d}(\,\boldsymbol{x},\boldsymbol{y})|\boldsymbol{y} \in (\boldsymbol{D} \setminus s))\,.$$

Specific algorithms often focus on formulating an anomaly score (or outlierness) function which can be used to compare pairs of points, such that $\alpha\,(\boldsymbol{x}) > \alpha(\boldsymbol{y})$ whenever a data point $\boldsymbol{x}$ is to be considered more anomalous than another data point $\boldsymbol{y}$. We can then present the results of the application of the algorithm in one of two forms, as required:

a. find the m most anomalous points; or
b. find all points whose anomaly score exceeds a specified threshold.

Cost-benefit analysis may dictate further refinements of such a procedure, as in the following example.

Example: Thousands of credit card transactions may occur in a minute, and computational constraints may limit the system's capacity to process more than 100 of them. Then the system should select the top 100 transactions, and a naive approach is to select these based on the anomaly score $\alpha(\boldsymbol{x})$ alone. But other considerations may also be important, such as the false-negative cost $\beta(\boldsymbol{x})$. In addition, the dollar amount of the transaction together with a quantification of other intangible costs (e.g., loss of trust in the system, and follow-up costs after fraud discovery); the cost $\gamma(\boldsymbol{x})$ of following up the transaction as a possible fraud; and the misclassification cost $\delta(\boldsymbol{x}, y)$ for a false positive, where y indicates the customer, since the cost may be higher for some customers (e.g., a large account may be lost by the bank if the CEO of a company is upset that s/he is unable to conduct a transaction while traveling). Rational decision theory now dictates that the anomaly score $\alpha(\boldsymbol{x})$ be translated into a probability $p(\boldsymbol{x})$ that describes the likelihood of the transaction $\boldsymbol{x}$ being fraudulent, which can only be done if a substantial amount of prior data is available, mapping anomaly scores of previous transactions to ground truth (indicating which transactions are fraudulent). Changes to the algorithm necessitate recalibrating the relationship between $\alpha(\boldsymbol{x})$ and $p(\boldsymbol{x})$. Then the expected cost of considering a transaction as fraudulent is $\gamma(\boldsymbol{x}) + \big(1 - p\,(\boldsymbol{x})\big)\delta(\boldsymbol{x},\boldsymbol{y})$ and the expected benefit is $p\,(\boldsymbol{x})\,\beta(\boldsymbol{x})$. Hence the main criterion for selecting transactions for further processing should be:

$$C\,(\boldsymbol{x},\boldsymbol{y}) = \;p(\boldsymbol{x})\beta(\boldsymbol{x}) - \;(\;\gamma(\boldsymbol{x}) + \;\big(1 - p(\boldsymbol{x})\big)\,\delta(\boldsymbol{x},\boldsymbol{y})\,)$$

and the resource-constrained system should select the top 100 transactions, as compared using the overall cost measure $C\,(\boldsymbol{x},\boldsymbol{y})$.

We now consider traditional approaches to measuring the anomaly score of a transaction $\boldsymbol{x}$, which essentially transforms data points into a single numerical measure.

Distance measures Traditional anomaly detection algorithms, applied to points in a d-dimensional Euclidean space with numerical coordinates, usually begin with the Euclidean distance between two points,

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

In many applications involving incommensurate dimensions, it is preferable to use the Mahalanobis distance or the normalized Euclidean distance,

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_i \left(\frac{x_i - y_i}{s_i}\right)^2}$$

where each dimension is separately normalized using the standard deviation s_i for that specific (ith) dimension. For example, if one dimension of a bank account is a customer's income (measured in units of currency), and another dimension is age (measured in unit of time), then the Mahalanobis measure circumvents the strange question of whether a difference in income of one rupee is equivalent to a difference in age of one year, one month, or one second.

The next question to be addressed is how we may evaluate the distance ***d(x, S)*** between a point ***x*** and a set of points ***S***, e.g., in defining what we mean by the distance between a potentially anomalous point and a cluster to which it does not belong. A few alternatives are possible, with definite implications for the notion of anomaly that is based on the point-to-set distance measure:

a) Euclidean distance between the given point ***x*** and the centroid μ of the set ***S***;

b) Mahalanobis distance $\sqrt{(x-\mu)^T\, V^{-1}(x - \mu)}$ where V is the covariance matrix between dimensions for points in ***S***;

c) Minimum Euclidean distance between ***x*** and the points in ***S***, i.e., distance from ***x*** to the nearest point in ***S***; and

d) Minimum Mahalanobis distance between ***x*** and the points in ***S***.

Example: In attempting to determine whether the online purchase of an expensive pair of shoes via a credit card transaction is fraudulent, we may compare it to other similar transactions that are believed to be non-fraudulent. For instance, if the customer had made prior online purchases from the same retailer, or had purchased another pair of shoes, of the same shoe size and similar cost, from another retailer, we may believe that the purchase is non-fraudulent. But if even the most similar prior purchase by that customer was significantly less expensive, or if the "shoe does not fit", then a minimum distance criterion would suggest that this purchase should be flagged with a high anomaly score.

k-nearest neighbor approaches: Nearest neighbor methods are among the simplest anomaly detection algorithms, and these may be used in a few different ways for anomaly detection. First, the distances $d_1, d_2, \ldots, d_k$ of a point ***x*** from its k nearest neighbors are calculated, where $k > 0$ is a predetermined integer. We may then adopt one of the following strategies to consider anomalousness of ***x***:

- A majority ($\left\lceil\frac{k}{2}\right\rceil$) of these distances exceed a predefined threshold;
- The mean $\left(\frac{1}{k}\sum_i d_i\right)$ of these distances exceeds a predefined threshold;
- The smallest ($min_i\ d_i$) of these distances exceeds a predefined threshold.

When the value of k is very small (e.g., k = 1), these approaches may fail to detect anomalies if there is a small cluster of anomalous points that are near each other but very far from a vast majority of other points in the data set. At the other extreme, when *k* is too high, non-anomalous points lying in a small cluster may be inaccurately tagged as anomalous.

Instead of a yes/no anomaly decision at this stage, if the goal is to compute an anomaly score, then the latter should be a quantity that increases with one of the distance measures (mentioned above) e.g.,

$$\alpha(\boldsymbol{x}) = \frac{\boldsymbol{d}(x)}{\boldsymbol{d}_{max}}.$$

which lies between 0 and 1 whenever an upper bound $\boldsymbol{d}_{max}$ is known for $\boldsymbol{d}$. When a data set grows with time, *k*-Nearest Neighbor approaches require that the effect of each new data point on (the neighborhoods of) all other data points must be determined. But no global or local properties of subsets of data are retained or updated, unlike clustering approaches that must retain information such as cluster centroid locations.

3.1 Clustering

A large number of unsupervised anomaly detection algorithms rely on clustering the given data set, identifying points that are outside (or at the boundaries of) clusters, and computing their anomaly scores.

The most well-known clustering algorithm is the k-means algorithm. The essence of the algorithm is that *k* seeds (data points) are randomly chosen to be the candidate cluster centroids, other points are allocated to the clusters associated with these centroids based on a minimum distance-to-centroid criterion, then the centroids are updated by arithmetic averaging within each cluster. The last two steps (allocation of points and updating centroids) are then repeated until results converge, i.e., remain almost unchanged.

The k-means algorithm dominates applications due to its simplicity, despite known problems, e.g., the results of the algorithm are not guaranteed to be optimal, and presenting the same data set in a different sequence may result in a different set of clusters. This algorithm also implicitly assumes that clusters are radially symmetric, which may not be appropriate.

X-means [59] is a variant of k-means; an algorithm which is advantageous over basic k-means algorithm to automatically determine the number of clusters. Chang et al. [19] used X-means algorithm to analyze the behavior changes of online auction fraudsters in Yahoo! Taiwan. Several other variations have also been suggested in the literature.

Issa and Vasarhelyi [42] proposed an anomaly detection method, based on k-means, to identify fraudulent refunds. Thiprungsri et al. [69] applied it to detect fraudulent life insurance claims by identifying small clusters with large beneficiary payment, huge interest amount and long processing. Nhien et al. [48] applied k-means clustering to study anti-money laundering detection.

An important concern is that the anomaly detection context justifies the non-allocation of some points to any cluster. This necessitates a modification to clustering algorithms, where a predetermined distance threshold may be established in order to consider a point to belong to a cluster, so that no cluster is allowed to contain "faraway" points. Variations of the Adaptive Resonance Theory approach [14] allocate such points to new clusters, not requiring a fixed number k of clusters; in the last step, points in very small clusters (with very few points) may be considered to be anomalies.

Asymmetrical cluster formation may be permitted by using a nearest-neighbor clustering approach instead of k-means: a point is then added to a cluster which contains its nearest neighbor, instead of the cluster with the nearest cluster centroid. Extending this approach, k-nearest-neighbor clustering allocates a data point to the cluster containing the largest number of its *k*-nearest neighbors that have been allocated already to clusters. As mentioned earlier, a distance threshold may be used to avoid forcing a data point into a distant cluster.

Another important class of clustering algorithms is *hierarchical*; they construct\dendrograms (trees) that preserve much more of the relative distance relationships than algorithms such as k-means. In the top-down approach, we begin with the entire set of data, and successively split it into two subsets (at each level of the tree) that are maximally distant from each other. This process is continued until subset sizes reach a pre-specified threshold (e.g., a threshold of 1 implies that each leaf node of the tree contains one data point). Alternatively, hierarchies may be constructed bottom-up, starting with singleton sets and successively merging sets that are nearest to each other. When new data arrives over time, the dendrogram may change substantially. For anomaly detection purposes, points whose distances are largest from others are of most interest, i.e., points which are separated early in the top-down approach, or added very late in the bottom-up approach. An anomaly score may hence be computed based on the stage at which a data point is removed (for top-down) or added (for bottom-up) with respect to the current tree.

Commonly used hierarchical clustering algorithms include BIRCH [57], CURE [58], and ROCK [59]. Although the computational complexity of BIRCH is low, it is sensitive to the data ordering and lacks the ability to handle combinations of numerical and categorical data. CURE [58] can recognize arbitrarily shaped clusters better than BIRCH but has higher computational complexity. ROCK can handle categorical data but its computational complexity is higher than CURE.

Rui et al. [69] used a combination of BIRCH [57] and k-means [34] to identify money laundering activities. Panigrahi et al. [70] used DBSCAN [30], a density based clustering algorithm, for credit card fraud detection.

Neural network models called self-organizing [feature] maps (SOM) [49, 50, 7] have also been used for partitioning data space into Voronoi regions, where each node in the SOM is analogous to a cluster centroid in the k-means algorithm, and each region includes the data points that are nearest to that node. Another important aspect of SOMs is that the network nodes are pre-configured into a topological structure (e.g., a two-dimensional mesh) that determines a neighborhood relation different from the distance relation between data points. Node positions are iteratively updated, focusing on the nodes nearest to each data point being presented, as well as their topological neighbors that are also adapted, though to a smaller degree. Variations of this approach, such as Growing Cell Structures [31], permit the number of nodes in the network to increase or decrease in successive iterations, thereby capturing data sets in which clusters appear to be clearly separated, unlike SOMs in which the predefined topological map (nodes with their neighborhood relations) do not change.

Deng et al. [24] proposed a clustering model VKSOM combining Self Organizing Map and k-means clustering for fraud detection in financial statements. The model enjoys the benefit of unsupervised self-learning SOM [55]; the k-means clustering algorithm is applied to the results of SOM.

When the density of data distribution varies in different parts of the data space, then the anomaly detection algorithms that rely on clustering may not perform well; a data point may be far away from its neighbors from the perspective of the high-density region, but not from the perspective of the low density region. This issue has been addressed by algorithms such as LOF [12] and its variants COF [68] and INFLO [43], which compute an anomaly score for a point ***x*** that is sensitive to the density of the neighborhood of ***x***, e.g., by appropriately normalizing the distance measure. The relevant question is the following: how far is a point from its neighbors, when compared to the distances of those neighbors from their own neighbors? Of course, the answer is not obvious if a point is “in between” a high density and a low density region, so that the nearest neighbors of the point differ in their local density.

Instead of directly addressing density using inter-point distance, rank-based anomaly detection algorithms such as RBDA [40] formulate an approach drawn from sociological considerations in identifying loners and socially marginalized people: *Do the friends of* **x** *consider* **x** *to be their friend?* For anomaly detection, this question is posed in terms of a relative ranking of nearest neighbors, e.g., the third nearest neighbor of ***x*** has rank 3 with respect to ***x.*** Given *k>0*, if the *k* nearest neighbors of point ***x*** are in set S, then we can compute the sum of the ranks of ***x*** with respect to each element of S. A true outlier would have a relatively poor rank (i.e., high value) with respect to most of its nearest neighbors, even if it is in the immediate neighborhood of one or two other outliers.

3.2 Learning with available class information

When some information about normal or anomalous data is available, anomaly detection has been modeled as follows:

1. Two-class classification (supervised learning) – Two classes of data, normal and abnormal, are assumed to be available. Almost by definition, there would be very few representatives of the anomaly class, which makes learning difficult. To mitigate this class imbalance, a cost-sensitive learning algorithm may be applied that assigns a much higher cost for misclassifying an anomalous observation than for misclassifying a normal observation [28]. However, determining these costs is not a trivial task and often demands an expert’s knowledge.
2. When representatives of only the normal class are given, an algorithm must be trained to recognize them; any points that cannot be assigned to the normal class are considered to be anomalous [8]. This is also known as one-class classification problem. This approach does not need a prior cost specification, and is hence of great interest.
3. When unlabeled data is utilized to complement labeled data and both types of data are used for training a predictive algorithm; the learning is known as semi-supervised learning, as discussed by Zhu [74] and Elkan and Noto [27].

In supervised anomaly detection, labeled data is available based on past history, and this can be advantageously used for anomaly detection. The simplest approach would be to evaluate a new data point with respect to its distances from the labeled elements of “normal” and “abnormal” classes. Points that are near the labeled normal data are considered to be normal, and those near the labeled abnormal data are considered anomalous. As before, the newly assigned label to the data point in question may depend on a majority of labels from among *k* nearest neighbors. Several additional tools are also available to address this problem when viewed as a two-class problem, e.g., Support Vector Machines (SVMs) [11, 38] and Feedforward neural networks [55] or Radial-basis-function (RBF) Neural Networks [33, 57, 26] can be applied to such data.

As per empirical findings, different anomaly detection algorithms work well for different data sets. This suggests the use of ensemble as well as multi-objective approaches. Ensemble methods apply multiple individual anomaly detection algorithms, and combine the anomaly scores obtained by each; more details are given in a later section. Multi-objective methods instead obtain a collection of data points that constitute the non-dominated elements, from the anomaly scores obtained using different component algorithms. In this context, a data point $\boldsymbol{x_1}$ weakly dominates $\ $\boldsymbol{x_2}$ if $\boldsymbol{x_2}$ is not considered more anomalous than $\boldsymbol{x_1}$ by any component algorithm; also, $\boldsymbol{x_1}$ strongly dominates $\boldsymbol{x_2}$ if $\boldsymbol{x_1}$ weakly dominates $\boldsymbol{x_2}$ but $\boldsymbol{x_2}$ does not weakly dominate $\boldsymbol{x_1}$. Among the most successful multi-objective algorithms are evolutionary algorithms such as NSGA-II [23]; the end result could, however, be a large set of non-dominated data points, and the selection of a smaller subset may involve the application of other heuristics, such as a preference for data points (from the non-dominated set) that are considered anomalous by a majority of the component algorithms.

4 Anomaly detection with time-stamped data sequences

So far, we have discussed the problem of anomaly detection for sets of data points, with no implicit ordering among them. However, many practical problems involve data that arrive over time, and are hence in a strict temporal sequence; treating the data as a set (ignoring the time-stamp) loses information essential to the problem. Treating the time-stamp as just another dimension (on par with other relevant dimensions such as dollar amounts) can only confuse the matter. The occurrence of a set of (other) attribute values at a specific time instant can mean something quite different from the same attribute values occurring at another time, depending on the immediately preceding values; these dependencies require explicitly modeling time as a special aspect of the data, and treating the data as a sequence rather than a set. Hence, anomaly detection for time-sequenced data requires algorithms that are substantially different from those discussed in the preceding section. Outlier detection for temporal data has been extensively surveyed by [20, 36]. Several anomaly detection techniques have been proposed for symbolic data sequences, and a recent survey is provided by Chandola, et al. [18].

Some researchers use model-based methods to find abnormal sub-sequences. First a model is generated to predict the behavior of the time series. Using this model, the predicted values are calculated and compared with the observed values. The cumulative score of the differences is defined as the anomaly score of each observed data object. These models include Regression, Auto-Regression [32], ARMA [56], ARIMA [56], and Support Vector Regression [67, 52]. These methods were mainly designed for individual outlier detection, not for abnormal subsequence detection, and the results are impacted significantly by model parameters or distributions of data sets. These deficiencies limit their practical application.

Applications of time series anomaly detection have emerged in many fields. For financial data, example time series are shown in Figure 1, representing the stock prices of the oil and gas companies for 2010 to 2012 in which the behavior of one series (dashed red line) is different from the rest.

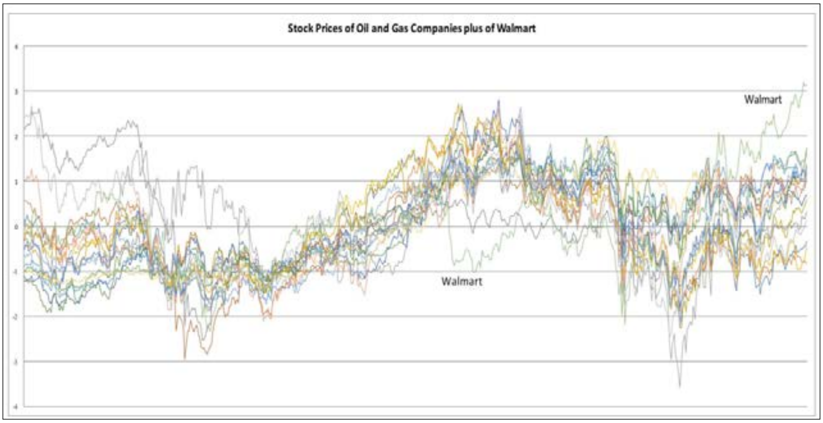


Fig. 1 Stock prices for some oil and gas companies and an outlier series. The anomalous series is stock price of Walmart (see right top corner, series in light green color).

Anomaly detection algorithms for data sequences fall into two categories, *viz.*, Procedural and Transformation approaches, as discussed below:

- In procedural algorithms, such as Regression and Hidden Markov Models (HMMs), a parametric model is built using the training data and an anomaly score is assigned to a test time series with associate probability.
- In the transformation approach, the data is transformed prior to anomaly detection. Transformations can be categorized into three major types:
    1. Aggregation approach focuses on dimensionality reduction by aggregating consecutive values.
    2. Discretization approach converts numerical attributes into a small number of discrete values, in order to utilize symbolic sequence anomaly detection algorithms and to reduce computational effort.
    3. Signal processing based transformations (e.g., Fourier and Wavelets) transform the data to a different space, and reduce the dimensionality of the data.

4.1 Finding anomalies within a sequence

Several kinds of anomalies have been categorized in the context of data sequences (over time). We first consider anomalies observed within a single (possibly multivariate) time series:

- The first is the occurrence of an *event* or *point anomaly*, characterized by a substantial variation in the value of a data point from preceding data points. For example, in the credit card fraud detection context, if a purchase is associated with a much higher cost than prior purchases by the same customer, an anomaly would be signaled. Sometimes, additional problem dimensions and refined analysis are important to avoid false negatives, e.g., the purchase over the Internet of a shirt for a hundred dollars may be flagged as anomalous even though the customer has made other purchases of higher cost; to characterize this transaction as an anomaly, prior data needs to be analyzed regarding the cost of prior purchases by this customer of clothing items over the Internet.

- Sometimes, it is important to detect anomalous sub-sequences within a given time series called *discords*, defined by Keogh et. al [47, 48] as “the subsequences of a longer time series that are maximally different from the rest of the sequence”. The problem is difficult because the exact length of the anomalous subsequence may be unknown. For example, in detecting cardiac arrhythmias from an electrocardiograph (ECG), no individual reading may be out of range, but the sequence of a collection of successive values may not be consistent with the regular and periodic form of the data. We expect a certain (previously observed) regular waveform to be repeated, and variation from this expectation constitutes an anomaly. Another example consists of overseas money transfers; past history might indicate these occur about once a month for a customer (e.g., to financially support the customer’s family living abroad), representing a simple periodic sequence over time, but the occurrence of multiple such transactions within a month then constitutes an anomaly. Keogh *et al.* [48] proposed a method to convert a series to multiple shorter subsequences, and used suffix trees to create an index structure for discretized sub-sequences; anomaly scores were computed by comparing trees generated by each subsequence. They also proposed an optimized dynamic time warping and SAX representation for time series [46, 47]. Wei *et al.* [71] proposed a time series BITMAP method based on comparison between the frequencies of SAX words of current data and past data.
- Sometimes the individual values may be within an acceptable range, but the rate of change over time may be anomalous, and we refer to this as a *rate anomaly*. For instance, in the credit card fraud detection problem, the balance (total amount owed by the customer) may suddenly increase within a day, due to a large number of small transactions made during a short amount of time; this should signal an anomaly even if each individual transaction is unremarkable, and the total balance remains within the range of prior monthly expenditures.
- Rate anomalies may be generalized to *contextual anomalies*, wherein a data point is anomalous with respect to the immediately preceding values, though not with respect to the entire range of possible values from past history. For instance, a customer's purchase transaction may be from a different country or city than the same customer’s immediately preceding transaction that occurred a short while ago; this would be inconsistent with the expectation that the customer could not have suddenly travelled a large distance within a short period of time. Some false positives could be prevented by paying attention to other features of the context, e.g., if the prior purchase was from an airport, then there is a higher probability that the next purchase could be from a different geographical location.

- Other aspects of a data sequence may be considered anomalous because of multiple attribute values, requiring prior analysis of normative patterns that adequately describe the data sequence. For example, a customer's history may indicate that certain kinds of purchases are made singly and not in groups of successive transactions, so that multiple successive clothing purchases over the Internet may be considered anomalous, even though each such purchase is normal. Another customer's prior history, on the other hand, may indicate that it is normal for that customer to make 1-5 successive clothing purchases in the same day, so the occurrence of anomalies for that customer need to be determined differently.

Detection of an abnormal sub-sequence can be recast as the problem of comparing each sub-sequence of a given length *w* to other sub-sequences (also of length *w*) in the time series of length n. More precisely: given a time series X, the set of extracted sub-sequences,$X_w = \{X_{p,w} ;\ 1 \le p \le (n - w + 1)\}$, consists of $X_{1,w} = \{ x(1), x(2), \dots \dots . . x(w)\};\ \ X_{2,w} = \{ x(2), x(3), \dots \dots . . x(w + 1)\}; . . . .$ $\dots;\ \ X_{n-w+1,w} = \{ x(n - w + 1), x(n - w + 2), \dots \dots . . x(n)\}$. One example is shown in Figure 2. We must determine if any $X_{i,w}$ is substantially different from the others, e.g., by evaluating whether its average distance to its k nearest neighbors is much larger than is the average distance for other sub-sequences.

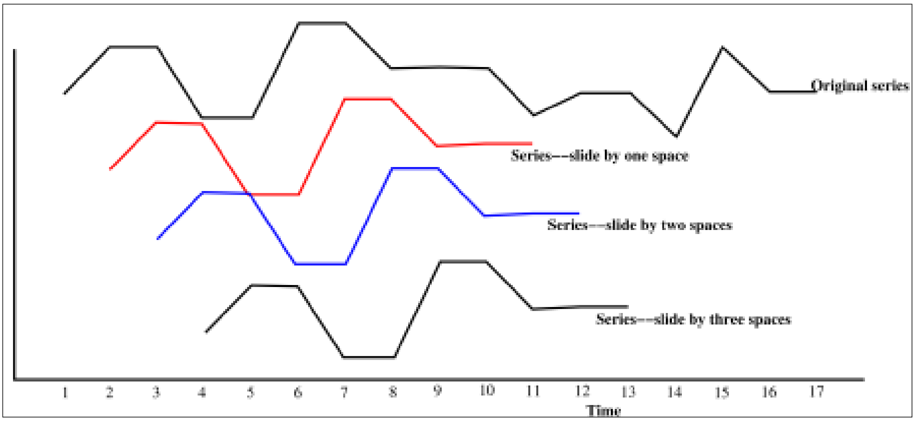


Fig. 2 Illustration for sliding window concept, with *w=10*

We observe that any sequence $X_{\mathrm{i,w}}$ is substantially similar to $X_{i+1,w}$, since consecutive series share most elements, and are likely to be the nearest neighbors with only a small distance between them. This *self-match* problem makes it difficult to find anomalous sub-sequences, addressed by comparing a sub-sequence only to others with no overlap, as in Keogh, *et al.* [46]. Hence, we consider the set of series in X_w, finding the k nearest neighbor subsequences that do no overlap with each sub-sequence $X_{p,w} \in X_w$. Several methods have been proposed to improve computational efficiency; e.g., to find the nearest neighbor of a subsequence [63] in X_w, we may use Euclidean distance and instead of $k>1$ nearest neighbors, we may use only the nearest neighbor, i.e., $k=1$.

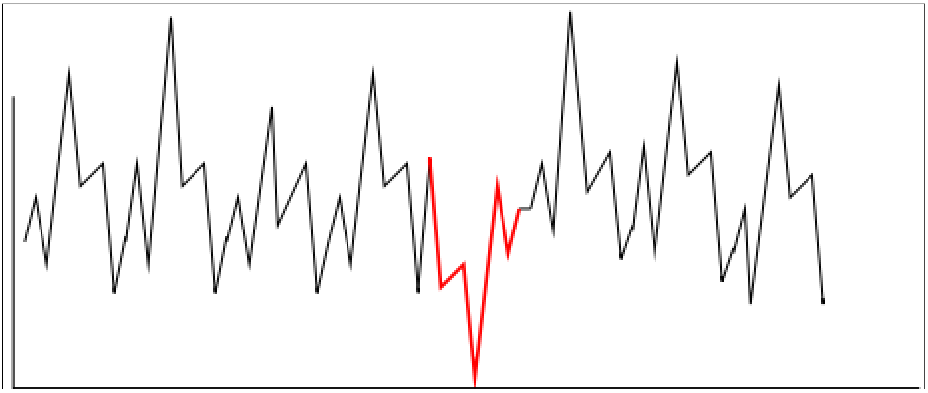


Fig. 3 Sub-sequence frequency – Red bold line represents a sub-sequence considered abnormal since it appears only once in this series, whereas other sub-sequence patterns occur much more frequently

Another approach, called *SAX words based frequency ratio (SAXFR)*, is based on assuming that the occurrence frequencies of abnormal sub-sequences are far lower than the frequencies of normal sub-sequences if length *w* is properly chosen. An example is shown in Figure 3. If we calculate the ratio between frequencies of SAX words generated from an abnormal sub-sequence with frequencies of these words in the entire series, then the ratio computed for an abnormal sub-sequence is much higher than the ratio computed for a normal series. The anomalousness measure is then defined to be α_{SAXFR} (X) = average of the reciprocal of the frequencies of SAX words contained in X.

For example, if three SAX words *abc, aac, abd* are generated from an abnormal subsequence, and these words appear *2, 3, 2* times respectively within the entire series, then the corresponding reciprocals are *1/2, 1/3, 1/2*. The average value can be calculated for such reciprocal frequencies, in this case, it is α_{SAXFR} (X)= *(0.50 + 0.33 + 0.50) / 3 = 0.44*. For another sub-sequence each of whose SAX words appear 10 times, this average would instead be 0.1, a much lower value, indicating that the sub-sequence is not anomalous.

To ensure that a SAX word is properly set to capture the outlierness of sub-sequence, the window size of SAX is set to *(w/2)*. If the window size of each word in SAX is too short, then the shorter sub-sequence represented by each SAX word may not be anomalous. If window size is too long, then the number of SAX words obtained in each sub-sequence (of length *w*) is less, and this may impact the results of the SAXFR approach.

The advantages of SAXFR are:

- Since the ratio is compared between a sub-sequence and the entire series, there is no need for nearest neighbor computation, nor for additional parameters.
- Frequencies can be pre-computed over the entire sequence, and sub-sequence SAX frequencies can be updated incrementally, so the computation is very fast.

In many algorithms, it may be difficult for users to find the right choices for parameter values. Another problem is that most algorithms focus on a single outlierness measure. Huang *et al*. [41] combine different measures, proposing an algorithm that requires little domain knowledge and can detect multiple abnormal sub-sequences in one run;

it also requires fewer parameters from users. Huang et al. proposed to use three measures – (1) SAXBAG, (2) STREND, and (3) DIFFSTD, described in Section 4.2, in the Multiple Measure Based Abnormal Subsequence Detection (MUASD) algorithm.

4.2 Finding anomalies between sequences

Another class of anomalies involves inter-time series anomalies, where multiple time sequences are available, of which a small number exhibit substantially different behavior. In other words, we are interested in determining whether the behavior of an individual time series differs substantially from most of those in a set of time series. For example, we may like to determine whether the sequence of overseas transactions being made by a customer is unusually high compared to other customers.

In many cases, the entire set of time series may be very large, with very little cohesion, so that it is difficult to identify anomalies with respect to the entire set. However, it may be possible to group the time series into clusters within each of which there is substantial similarity, so that a time series would be compared to those in the most similar subset. The grouping may be either on the basis of data attributes, or similarity between time series. For example, the number of overseas transactions made by a customer may be much higher than those made by most others with the same balance amount or income; this would be anomalous, even though it is normal for many other (wealthier) customers to make similar numbers of overseas transactions in the same period of time.

In the aforementioned cases, we encounter several problems. One of the most critical among them is the determination of an appropriate distance measure; it is not easy to determine the best similarity or distance measures that can be used for different types of time series. Euclidean distance is highly sensitive to outliers and also cannot be used when the time series are of different lengths. Another concern is the presence of random noise, naturally inherent in many time series; distinguishing noise from anomalies is a challenging task. Additional complications arise due to practical considerations, e.g., the length of the series may be very large and, consequently, the computational complexity of the proposed algorithm is prohibitively large.

Suppose we are given a time series data set $D = \{ x_i\,(t) \mid 1 \leq\ t \leq n; i = 1, 2, \ldots\ldots, m\}$, where $x_i\,(t)$ represents the data object of the i^{th} time series at time t, n is the length of the time series, and m is the number of time series in the dataset. The goal is to calculate $O(x_i)$, the outlierness of a series x_i, and $O_{threshold}$ a threshold such that if x_i is an anomalous series, then $O(x_i) \geq O_{threshold}$; *otherwise* $O(x_i) < O_{threshold}$.

The first step is to obtain a compact representation to capture important information contained in the series. Three main categories of approaches have been identified, by Ding *et al.* [25] and Chandola *et al.* [17], to reduce the dimensionality of time series: *model based*, *data-adaptive*, and *non-data adaptive* approaches. The compact representations obtained using these approaches must then be compared using suitable distance measures, which have been categorized into three groups [25]:

- The lock-step measures are based on one-to-one mapping between two-time series. Examples include the Cross-Euclidean distance (EUC) [29], the Cross-Correlation Coefficient-based measure, defined as

$\sqrt[2]{2(\,1 - corrcoef(d_x - \,d_y))}$ [10], SameTrend (STREND), and standard deviation of differences (DIFFSTD).

The elastic measures are based on one to many mapping between two time series, e.g., Dynamic Time Warping (DTW) [9, 44], and Edit Distance on Real sequence (EDR) [21]. For example, the similarity measure used by DTW addresses differences in two series, possibly due to acceleration or deceleration during the course of measurements, compounded by time lags caused by real-life processes such as the sensor measurements upstream and downstream in a manufacturing process. A time series *X* may be very similar to the series *Y*, except that they are out of sync in some places in the time domain. DTW attempts to accommodate such discrepancies in *X* and *Y* series. Consider *C(t, l)* = | *x(t)* - *y(l)*| which represents the cost of aligning *x(t)* with *y(l)*. Then, the goal of DTW is to

$$minimize \sum\sum C(t,l).$$

This can be accomplished by using the classical dynamic programming algorithm developed to align two sequences.

- A time series can be transformed into another space; measures in the transformed space include TQuEST [1] distance, and Spatial Assembling Distance SpADe [22]. Other examples include Symbolic Aggregate approXimation (SAX) proposed by Keogh and Lin [46] with and without sliding window (SAXSL and SAXNW respectively); SAX with bag-of-pattern (SAXBAG) [51], Discrete Wavelet Transform [16], and Discrete Fourier Transform [29].

Many variations on these approaches have been proposed in recent years. For example, Wei *et al.* [71] proposed an algorithm to find unusual shapes by using SAX representation to speed up the process for selecting the best candidates; Lin *et al.* proposed using the distance between *bag-of-pattern* [51] based on the frequency of each unique SAX word, which can be applied for anomalous time series detection, Keogh *et al.* [44] suggested indexing techniques for dynamic time warping by using a lower bounding measure, which can be used for detecting distance measures of time series, Protopapas [62] proposed an outlierness measure based on the average of correlations between time series. Chan [16] showed that Euclidean distance in the Haar-wavelet transformed domain can be effective for finding time series matches, and reported that this approach outperformed the Discrete Fourier Transformation proposed by Faloutsos [29], and Bu *et al.* [13] applied Haar wavelet transform on time series and then built an augmented trie to find the top *k* discords in the time series database.

In Table 1, we summarize the advantages and disadvantages associated with using some of the distance measures. We observe that no single measure is capable of capturing different types of perturbations that may make a series anomalous, hence multiple measures should be considered together. For the best utilization of limited computational resources, we may select measures that are orthogonal to each other, to minimize redundancy. One approach minimizes redundancy by selecting three measures that are least correlated with each other: (a) SAXBAG, (b) Same Trend (STREND), and (c) Standard deviation of differences between two time series

(DIFFSTD); described below. These measures capture different aspects of the time series, and a combination of these gives a comprehensive measure of how isolated is a time series from others in the comparison set. SAXBAG captures behavior of a time series using a histogram of possible patterns, STREND identifies the degree of synchronization of a series compared with another series, and DIFFSTD measures the amount of deviation, as illustrated in Figure 4. Combining these three metrics produces a comprehensive and balanced distance measure that is more sensitive than individual measures.

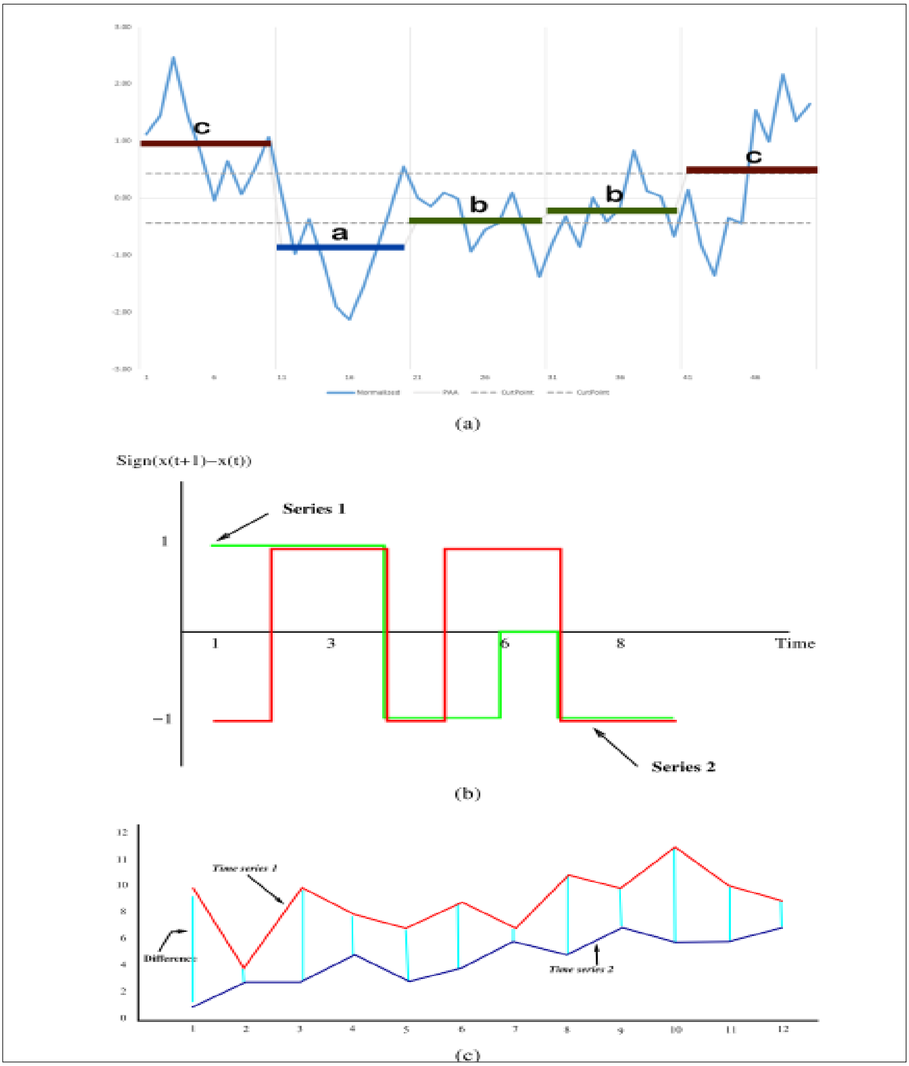
- SAXBAG (proposed by Lin *et al.* [51]: Given a time series, a sub-sequence of size *l* is obtained using a sliding window. Each sub-sequence is reduced to *w*-dimensional discrete representation in two steps. First, the sub-sequence is further divided into $u = \frac{l}{w}$ equal size sub-subsequences and the average value of the data falling within a sub-subsequence is evaluated. In the second step, each average is discretized to a symbol, from a predetermined set of symbols, using equal probability intervals approach. Figure 4(a) illustrates these concepts using three symbols: *a, b, and c.*
- STREND: For each i, we calculate the difference $x_t' = x_i(t+1) - x_i(t), t \in [1 \ldots\ldots n-1]$and define:

$$S_{i,j}(t) = \begin{cases} 1 & if\ x_i'\ (t).x_j'\ (t) > 0 \\ -1 & if\ x_i'\ (t).x_j'\ (t) < 0 \\ 0 & otherwise \end{cases}$$

  Clearly, *S* $S_{i,j}(t)$ indicates whether or not $x_i(t)$ and $x_j(t)$ change in the same direction at time *t*. The aggregate measure, over the entire length, *n*, of the time series is evaluated as,

$$dist\ (i,j) = 1 - \sum_{t\ \in[1\ldots.n-1]} S_{i,j}(t)/n - 1 \qquad (1)$$

- DIFFSTD is the standard deviation of differences between two time series, i.e., if $\delta_{i,j}(t) = \|x_i(t) - x_j(t)\|$, and $\mu_{i,j} = \sum_t \frac{\delta_{i,j}(t)}{n}$, then the new distance is defined as $dist\left(x_i, x_j\right) = \sqrt{(\sum_t(\delta_{i,j}(t) - \mu_{i,j}\ (t))^2/n}$. This measure is widely used in *pairs trading* in the financial field, which monitors the performance of correlated securities.



(a)

(b)

(c)

Fig. 4 Illustrations for three key measures (a) Illustrates how a SAX word, in a sliding window, is generated. SAXBAG counts the frequencies of such words in the word-sequence; (b) illustrates STREND (table under the figure shows $S_{i,j}(t)$'s); and (c) illustrates DIFFSTD (vertical lines show the differences between two time series x_1 *and* x_2).

Measures	**Pros**	**Cons**
EUC[29] CCF [10] DIFFSTD	Easy to implement; Computationally efficient	Lock-step measure; Normal series with time lagging cause problems
Dynamic Time Warping [44]	Elastic measure; Comparison with time lagging	Small deviations may not be detected
Discrete Fourier Transform [29] (DFT)	Good in detecting anomalies in frequency domain; Normal series with time lagging do not cause problem	Small deviations may not be detected; Cannot detect anomalies with time lagging
Discrete Wavelet Transform [16] (DWT)	Good in detecting anomalies in frequency domain	Small deviations may not be detected; Sensitive to time lagging
SAX with sliding window [46] (SAXSL)	Tolerates noise, as long as its standard deviation is small	May not detect abnormal subsequence of shorter length than feature window size; Normal series with time lagging can result in false positives
SAX without sliding window [46] (SAXNW)	Tolerates noise, as long as its standard deviation is small	May not detect abnormal subsequence of shorter length than feature window size; Small deviations may not be detected Normal series with time lagging can result in false positives
SAX with bag-of-pattern[51] (SAXBAG)	Tolerates noise, as long as its standard deviation is small; Normal series with time lagging do not cause problem	Cannot detect anomalies with time lagging; Cannot detect anomalous series with similar frequencies but different shapes

Best results were obtained by using the following refinements:

- Elements of each time series are first normalized to have mean 0 and standard deviation 1.
- Normalization was performed by dividing observations by trimmed mean (excluding 5% on either end).
- Higher weights are assigned to the more effective measures, determined using RBDA.
- Deciding whether a point is anomalous is finally performed by thresholding the combined anomaly score, or selecting the highest values.

Sometimes data arrives incrementally, and online anomaly detection algorithms are required. Some distance measures can be updated incrementally, e.g.,

- **DIFFST**: The variance of differences between series *i* and *j* at time *n* can be calculated as:

$$\text{dist}_f\ (i,j)=\frac{n \times ssq(i,j) - (\ sqs(i,j))^2}{n{\times}(n-1)}, \qquad (2)$$

where $ssq(i\ ,j) = \sum_{t=1}^{n}(x_i(t) - x_j(t))^2$ and $sqs(i, j)=\sum_{t=1}^{n}(x_i(t) - x_j(t))$ (3)

The numerator in Equation 2 can be updated for the $(n + 1)$th observations by adding $(x_i(n + 1) – x_j(n + 1))^2$ and $(x_i(n + 1) – x_j(n + 1))$ to $ssq(i, j)$ and $sqs(i\ , j)$ respectively.

- **STrend**: Let $x_i'\ (n) = x_i(\ n) – x_i(n – 1)$. Then, by definition,

$$S_{i,\,j}(\ n\ ) = \begin{cases} 1\ \ if\ x_i'\ (n).x_j'\ (n) > 0\ \ or\ \ x_i'\ (n) = x_j'\ (n) = 0 \\ \qquad\qquad 0\ otherwise \end{cases}$$

Consequently,

$$\text{dist}_t\ (i, j)=\frac{\sum_{t=2}^{n} S_{i,j}}{n-1} \qquad (4)$$

Therefore, to update this value using the $(n + 1)$th observation, we modify the numerator by adding the last trend $S_{i,\ j}(\ n\ +\ 1)$, and accordingly modify the denominator as well.

- **SAXBAG**: converts the data segment in the sliding window of size w to a single SAX word, and then counts the frequency f_i of each word. When data at time $n + 1$ is observed, a new SAX word will be generated based on the sequence

  $x_i\ (n + 2 – w), x_i\ (n + 3 – w), x_i\ (n + 4 – w),\ldots\ldots\ldots, x_i\ (\ n +1)$.

  The stored data set can be updated to account for the new SAX word.

- **Normalization and Assignment of Weights:** We normalize $dist_f\ \ (i,\ j)$, $dist(i,j)$, and $dist_s(i, j)$ to $dist_f'(i, j)$, $dist_t'(i, j)$, and $dist_s'(i, j)$, respectively. Then $\text{weight}_l'(i)$ values are also calculated.

- Finally, the anomaly score for the i^{th} series is calculated as follows:

$$\text{A}_i = \sqrt{\sum_{l\in\{s,t,f\}}(dist_l'(i))^2 \times \text{weight}_l'(i)}.$$

A simple weighted distance can be used for fast online detection, without considering rank:

$$dist_{msm}(i, j)=\sqrt{\frac{\sum_{l\in\{s,t,f\}}(dist_l'(i,j))^2}{3}} \qquad (5)$$

Then, the anomaly score for i^{th} series, $\text{A}_i\ (k)$, can defined as the average distance to its k nearest neighbors:

$$\text{A}_i\ (\ k\ )= \frac{\sum_{j\in N_k(i)} dist_{msm}(i,j)}{|N_k(i)|} \qquad (6)$$

Now, select any k neighbors of i, and let A'_i (k) denote the average $dist_{msm}(i, j)$ over them. Then the average distance of k-nearest-neighbors of i must be *less* or equal to average distance of any k neighbors of i, so:

$$A_i (k) \leq A'_i (k) \tag{7}$$

Huang *et al.* [39] propose OMUDIM, a faster version of MUDIM, whose key steps are as follows:

1. Find a threshold λ such that $A_i(k) < \lambda$ implies i is not an anomalous series; then any A'_j $(k) < \lambda$ also implies j is not an anomalous series either; thus most of the non-anomalous series can be excluded from anomaly score calculations. To find an estimate of the threshold, λ, we apply the following:
2. Sampling procedure: We calculate $A_i(k)$'s for $i \in S$, where S contains a small fraction (α) of the elements in D, randomly selected. Then λ is chosen to equal the value of $A_i(k)$ which is at the top ($\beta \times 100$)th percentile in descending order.
3. For $x_i \in D - S$, maintain a binary max heap consisting of $dist_{msm}(i, j)$ where j's are selected k neighbors of i. If the average of these k neighbors is less than λ, series i is declared as non-anomalous. Else $dist_{msm}(i, j)$ is calculated for next selected value of j, and the heap is updated by keeping only the smallest k values of $dist_{msm}$. The anomalousness of series i is tested using the above criterion. This process stops if at any stage, the series is found to be non-anomalous or no j is left.
4. Calculate the anomaly scores of all potential anomalous series (found in Step 2) and find the anomalous series, if any.
5. The above steps are repeated once new observations arrive.

This algorithm outperformed three other online detection algorithms based on (a) Euclidean distance, (b) Dynamic Time Warping (DTW), and (c) Autoregressive (AR) approach, proposed by [17], [46] and [32] respectively. The first two of these methods calculate a measure of anomalousness of a time series by (i) finding the k nearest neighbors of the series, and (ii) using the average distance of these k neighbors. The third method constructs a global AR model for all series and then measures the anomaly score at time t as the gap between the observed value and the predicted value.

The MUDIM approach is efficient and detects anomalous series as soon as it begins to drift away from the other (non-anomalous) series, a substantial advantage over other anomaly detection algorithms for time series. This approach can handle data uncertainty very well, and its online version does not require any training data sets. Compared with other methods, it requires less domain knowledge.

5 Anomaly detection with categorical data

Categorical (or nominal) data pose special problems for the formulation of suitable anomaly detection algorithms. The biggest issue is that numerical distance measures cannot easily apply; the simplest distance measure for a single categorical attribute is binary-valued: the distance is 1 if the two values of an attribute are different, and 0 if identical. Transforming a d-valued categorical attribute into d binary attributes does not make any substantial difference to this problem.

Many practical problems involve such data, in which some of the variables are nominal or categorical, i.e., not numeric. For example, a credit application for a bank may state an occupation which is relevant to decision-making; numeric encoding of the same can only confuse data analysis procedures, since an occupation value of "7" may be as different from "8" as it is from "1". Hence, anomaly detection algorithms that rely on distance computations must not attempt to reduce categorical variables to numerical values.

One approach for anomaly detection with categorical data is based on data mining approaches that focus on frequent item-set identification; the terminology comes from the grocery shopping analogy, in which shoppers make transactions, and each transaction involves the purchase of a certain set of items. Data mining analysis can help determine which sets of items tend to be purchased together, with actionable implications for the merchant or grocery store. Well-known algorithms [4, 3] have been developed to identify frequent item-sets from large numbers of transactions.

From the anomaly detection perspective, we are interested in transactions that contain infrequent item-sets, i.e., items that are not usually purchased together, and are often purchased separately. An anomaly score can readily be constructed, based on this heuristic, high if $f(\boldsymbol{x},\boldsymbol{y})/f(\boldsymbol{x})f(\boldsymbol{y})$ is small, where f indicates the frequency of purchase of that item [REF]. This principle also applies to other contexts in which some variables are categorical. Another perspective is to evaluate the normalized conditional (marginal) probabilities of item-sets.

Another well-founded approach is based on an information compression perspective, relying on a minimum description length (MDL) methodology [34]. First, each transaction can be considered to be covered by a collection of item-sets, of varying frequency in the set of transactions. A coding approach is utilized which assigns codewords (symbol sequences) to various item-sets, with frequent item-sets represented as the shortest codewords, as in the KRIMP algorithm [65]. A transaction that consists only of very frequent item-sets will hence have a much shorter codeword representation than a transaction that consists of randomly assorted items. Thus, the anomaly score of a transaction can be considered to be proportional to the length of its representation using item-set codewords. A small number of items missing from the item-sets can be tolerated; Smets and Vreeken propose a – fault-tolerant cover computation algorithm, which attempts to use as few item-sets as possible to describe a transaction [66].

For example, consider a database with three categorical variables and six transactions (data points in the three-dimensional space): *abx, abx, abx, abx, acx, acy*, where each letter {*a, b, c, x, y*} indicates a different value for one of the three categorical variables. The code assigned to the most frequent item-set, *abx*, is 0, of length 1 bit. The code assigned to the next most frequent item-set, *ac,* is 10, of length 2 bits. The code assigned to *x* is 110, of length 3 bits, and the code assigned to *y* is 111, also of length 3 bits. Now, we can determine the length of the minimal cover for a given transaction using these codes. For example, *acx* requires 2 bits (for *ac*) + 3 bits (for *x*) = 5 bits, indicating that it is far more anomalous than the transaction *abx* which requires only 1 bit.

This approach was further extended in the parameter-free CompreX algorithm [6] that uses multiple dictionaries (compression tables with codewords), and identifies transactions with high compression cost as anomalies.

6 Ensemble methods for anomaly detection

Several anomaly detection algorithms have been developed, and have been fine-tuned to work well in specific data distributions. Ensemble approaches attempt to combine multiple individual algorithms in order to obtain better results over a larger class of problems and data distributions. The two main classes of ensemble approaches [2] are:

1. Sequential: The results obtained by one algorithm are refined by the application of another algorithm.

2. Parallel (or Independent): Each algorithm is applied to the data set, and the results obtained using multiple algorithms are combined. The MUDIM approach discussed in the previous section is an example of this approach, for time series problems.

These approaches work best if different algorithms being utilized are orthogonal to each other, e.g., one catches the anomalies that the other does not.

6.1 Sequential ensembles

The principle of successive refinement is applied by setting algorithm parameters to permit the first algorithm to act as a “coarse sieve” that minimizes false negatives but permits a greater number of false positives, whereas each successive algorithm reduces the number of false positives. Zhiruo *et al.* [72] have developed new sequential ensemble algorithms based on the concept that anomaly detection performs better on a subsample of the dataset, and propose to use the algorithms which have higher diversity among themselves as the base algorithms for a better combination result. Iterative sequential learning over base algorithms processes data in multiple passes, with each round of execution providing a better understanding of the dataset. As a result, the final result will provide a more refined result than obtainable using a single iteration.

Boosting is a well-known example of iterative sequential learning. In particular, the *AdaBoost* algorithm [30] has gained substantial popularity for classification problems. But this approach has not been explored much in unsupervised anomaly detection, since labeled training data are unavailable. Zhiruo *et al.* [72] propose a novel adaptive learning algorithm for unsupervised outlier detection, which uses the score output of the base algorithm to determine the hard-to-detect examples, and iteratively resamples more points.

Random forests constitute ensemble decision-making with multiple randomly generated decision trees, improving performance over single decision trees and other approaches for classification problems. Guha *et al.*[35] have shown that random trees can be used for anomaly detection in a semi-supervised context. However, parameters must be chosen based on empirical learning, and performance suffers for when the problem dimensionality increases. Zhiruo *et al.* [73] have analyzed the impact of parameters used in random trees from both empirical and theoretical points of view, and proposed new algorithms to solve the problem of anomaly detection over high-dimensional data, partitioning the feature space into similar clusters, then building random trees separately.

Future work must address how to extend ensemble methods to streaming data, especially in the presence of concept drift. Since high computational costs are required for iterative computations, it is also important to develop algorithms that reduce computational effort.

6.2 Independent ensembles

The literature on information fusion, e.g., from multiple heterogeneous sensors [70, 15], provides a contrast between data-level, feature-level, and decision-level fusion, providing useful analogies for ensemble methods for anomaly detection.

- Data-level: A single algorithm is applied to data obtained from multiple sensors. This is not an ensemble approach.
- Feature-level: Preliminary analysis identifies important features that are then analyzed together, in the data fusion context. Analogously, we may consider the anomaly scores, obtained from different individual anomaly detection algorithms, as features that need to be combined by averaging the scores, or considering the maximum of the scores, which can then be ranked.
- Decision-level: The final results from various algorithms are combined, in this approach. For anomaly detection, the *ranks* obtained for each point using different anomaly detection algorithms may be considered to be their final decisions; formally, if $\alpha_i\,(x_j)$ is the anomaly score for the data point x_j using the i^{th} algorithm (applied to the data set D), then the corresponding *rank* is defined as follows:

$$r_i(x_j) = |D| - \left|\left\{x_k \middle| \alpha_i(x_k) < \alpha_i(x_j)\right\}\right|.$$

For example, if the data set contains a thousand elements, and all their anomaly scores are different, then the element ranked 1 is the one such that 999 elements have lower anomaly scores, the element ranked 2 is the one such that 998 elements have lower anomaly scores, etc. This definition implies that the higher value is used if elements have the same rank, e.g., if two elements have the same highest anomaly score, then both would have to be ranked 2. In the decision-level approach, these ranks have to be combined; this may be done using one of the following approaches:

1. Min-rank, defined as $r_{min}(x_j) = \min_i r_i(x_j)$. Thus, all the items ranked 1 by any algorithm receive the final min-rank of 1.
2. Aggregate-rank, defined as $r_{sum}\,(x_j) = \sum_i r_i(x_j)$. This is equivalent to an averaging of the ranks.

Each of these may be sorted, resulting in a linear ordering, from which a final rank may be extracted; we use ρ to denote such a function, so that $(x_i) \in \{1 \ldots \ldots .\, |D|\}$ when the data set is D, and (x_i) > (x_j) iff either $\alpha(x_i) < \alpha(x_j)$ for a single or composite anomaly score function , or $r(x_i)$> $r(x_j)$ if *r* is a single or composite rank function.

Example: Consider a dataset with a thousand points *D*= $\{x_1,.....,x_{1000}\}$, to which three anomaly detection algorithms are applied with the following respective anomaly scores for some of the points as shown in Table 2; each column (other than the first) corresponds to the data points $\{x_1, ...,x_5\}$, and we assume that all $\alpha_i$values equal 0 for the remaining points in *D*, i.e., they are not considered to be anomalous even to the least degree by any algorithm. We observe from Table 2 that somewhat different results are obtained (shown in the ρ values) by different ensemble combination methods, although they usually agree in which elements are considered most (x_2)or (x_5) least anomalous.

7 Mathematical perspectives

Many algorithms for anomaly detection have been implemented, as discussed above. We now focus on simple mathematical formulations of the problems being addressed, providing a basis to evaluate competing algorithms that may not directly address a mathematical goal; some earlier definitions are repeated for clarity.

7.1 Prediction

Let *f* be a mathematical model for an unknown process that generates values for an unknown variable $\mathcal{Y}$, given a collection of known variables $\boldsymbol{x}$. Then | $f(\boldsymbol{x})-\mathcal{Y}$ | measures the amount of deviation between the predicted and actual values of $\mathcal{Y}$.

Table 2 Example of anomaly scores and ranks for five points using three anomaly detection algorithms; the ρ values give the final results of applying the ensemble operations in various ways.

	x_1	x_2	x_3	x_4	x_5
α_1	0.9	0.8	0.7	0.6	0.5
α_2	0.9	0.9	0.9	0.4	0.4
α_3	0.2	0.9	0.5	0.7	0.3
α_{max}	0.9	0.9	0.9	0.7	0.5
$\rho_{\alpha max}$	3	3	3	4	5
α_{sum}	2.0	2.6	2.1	1.7	1.2
$\rho_{\alpha sum}$	3	1	2	4	5
r_1	1	2	3	4	5
r_2	3	3	3	5	5
r_3	5	1	3	2	4
r_{min}	1	1	3	2	4
ρ_{rmin}	2	2	4	3	5
r_{sum}	9	6	9	11	14
ρ_{rsum}	3	1	3	4	5

This leads to defining anomalies as those observed points ($\boldsymbol{x}, \mathcal{Y}$) for which | $f(\boldsymbol{x})$ $-\mathcal{Y}$| is maximized.

7.2 Classification

Let X_1 and X_0 be two sets of data points, of which the former are known to be "normal" and the latter are known to be anomalous.
A classifier C is first to be trained on this data set, minimizing the number of classification errors, i.e., minimizing

$$|\{\boldsymbol{x} : \boldsymbol{x} \in X_1 \text{ and } C(\boldsymbol{x}) = 0\}| + |\{\boldsymbol{x} : \boldsymbol{x} \in X_0 \text{ and } C(\boldsymbol{x}) = 1\}|$$

To facilitate learning using algorithms such as error back-propagation for feedforward neural networks, often it is preferred to minimize the mean squared error, proportional to,

$$\sum_{x_i \in |\mathrm{X}_1|} (1 - f(\boldsymbol{x}_\mathrm{i}))^2 + \sum_{x_i \in |\mathrm{X}_0|} (f(\boldsymbol{x}_\mathrm{i}))^2$$

where f describes the function implemented by the model after learning, presumed to be constrained such that $0 \leq f(\boldsymbol{x}_\mathrm{i}) \leq 1$ for all data points $\boldsymbol{x}_\mathrm{i} \in X_1 \cup X_0$ **.** To avoid penalizing points whose values are close to the target values (0 or 1), a minor variation of the above is the task of minimizing

$$\sum_{x_i \in |\mathrm{X}_1| \,\&\, f(x_\mathrm{i}) < 1-\epsilon} (1 - \epsilon - f(\boldsymbol{x}_\mathrm{i}))^2 + \sum_{x_i \in |\mathrm{X}_0| \& f(x_\mathrm{i}) > \epsilon} (f(\boldsymbol{x}_\mathrm{i}) - \epsilon)^2$$

where $0 < \epsilon < 1$. Since many different models may be constructed for a given dataset, the trade-off between model complexity C_M and error magnitude is achieved by introducing a regularization term, i.e., minimizing,

$$E + \lambda C_M$$

where E is the (misclassification or mean squared or other) error term described above, C_M may be measured as the sum of the number of trainable parameters in the model or the sum of the magnitudes of the model parameter values, and the regularization parameter $\lambda > 0$ describes the permitted tradeoff.

7.3 Clustering

For the purposes of anomaly detection, the goal of clustering is to group data points together, which is quite different from that of partitioning the data space into disjoint regions. We need to address two distinct considerations, discussed below:

1. When does a data point belong to a cluster?
   The goal is to associate each data point with a cluster which is at the least distance. If an *a priori* decision is made to cluster the data set $\boldsymbol{D}$ into k clusters, then the goal is to determine a function C such that $C(\boldsymbol{x}_\mathrm{i}) \in \{1, . . . , k\}$, minimizing

$$d(C, D) = \sum_{x_i, x_j : C(x_i)=C(x_j)} d(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

However, k (the number of clusters) is not given to us in an anomaly detection task, requiring us to address the following question:

2. How many *(k)* clusters should we use?
   This question may be answered in many ways.

- A maximum threshold θ may be specified on the distance between a point and the cluster to which it is assigned; we must then determine a function C such that $C(x_i) \in \{1, 2, 3, \ldots\}$, minimizing $d(C, D)$, but subject to the constraint that $C(\boldsymbol{x}_i) = C(\boldsymbol{x}_j)$ implies $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \theta$

- A threshold may instead depend on the point $\boldsymbol{x}_i$ in question, e.g., the local density in a region of the data space containing $\boldsymbol{x}_i$, so that we replace θ in the previous constraint by a function of the distances between points in a neighborhood $N(\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$, such as,

$$\theta(\boldsymbol{x}_i) = \eta \frac{\sum_{x_j \in N(x_i)} \sum_{x_k \in N(x_j)} \mathbf{d}(x_j, x_k)}{|N(x_j)|\,|N(x_i)|}$$

  where $0 < \eta < 1$. The definition of the neighborhood may consist of a fixed number of nearest points, e.g., such that $|N(\boldsymbol{x}_i)| = n$, a predetermined size, and $\boldsymbol{x}_j \in N(\boldsymbol{x}_i) \& \boldsymbol{x}_k \notin N(\boldsymbol{x}_i)$ implies $\boldsymbol{d}(\boldsymbol{x}_i, \boldsymbol{x}_k) \geq \boldsymbol{d}(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

- Instead, we may impose a criterion based on inter-cluster distances, e.g., minimizing the ratio of the mean intra-cluster to the inter-cluster distance, i.e.,

$$R = \frac{\frac{1}{k}\sum_{i=1}^{k} \boldsymbol{d}(C_m)}{\frac{1}{k(k-1)}\sum_{i \neq j} \boldsymbol{d}(C_i, C_j)}$$

where the intra-cluster distance for the mth cluster is defined as

$$\boldsymbol{d}(C_m) = \frac{\sum_{x_i, x_j : C(x_i)=C(x_j)} \boldsymbol{d}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{|\{\boldsymbol{x}_j : C(\boldsymbol{x}_j) = m\}|}$$

and the inter-cluster distance between the i^{th} and j^{th} clusters could be defined in different ways, e.g.,

$$\boldsymbol{d}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y})$$

or

$$\boldsymbol{d}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y})}{|C_i| . |C_j|}$$

In an incremental approach, e.g., where clusters are successively merged, a greedy algorithm is usually followed, i.e., continuing to merge clusters until the ratio R begins to increase, or until R exceeds a pre-specified threshold.

7.4 Decision-theoretic optimization

Although minimization or maximization of some measure is inherent in the formulations given above (viz., prediction, classification, or clustering), we may directly pose the anomaly detection task as an optimization task, and address the same using a global optimization algorithm, obtaining potentially “better” solutions, with the caveat that we then expect greater computational expense. Each such formulation provides a standard against which the quality of results obtained by a specific algorithm can be evaluated.

We now consider the decision-theoretic task, where false positive cost $C^{+}(x_i)$ is associated with mislabeling a non-anomalous data point x_i, and false negative cost $C^{-}(x_i)$ is associated with mislabeling an anomalous data point x_i. The outputs of an anomaly detection algorithm are interpreted to suggest that only the points in A $\subset D$ are anomalous. Ground truth may be available suggesting that a point x_i is an anomaly with probability $p(x_i)$.

Then the goal is to minimize the total cost of misclassification for the algorithm, i.e.,

$$\sum_{x_i \in A} C^{+}(x_i)\left(1-p(x_i)\right)+\sum_{x_i \in D \setminus A} C^{-}(x_i)\ p(x_i)$$

If the anomaly detection algorithm provides anomaly scores $\alpha(x_i)$ in the [0,1] interval, then the choice of a threshold θ (above which a point is to be considered an anomaly) is obtained by minimizing the total cost obtained by the choice of θ, i.e.,

$$\underset{\theta}{argmin}\ \sum_{x_i:\,\alpha(x_i)\geq\theta} C^{+}(x_i)\left(1-p(x_i)\right)+\sum_{x_i:\,\alpha(x_i)<\theta} C^{-}(x_i)\ p(x_i)$$

If, on the other hand, the goal is to infer probabilities from the anomaly scores, then a calibration function f_a must be learned using the data points X for which such probabilities are available (from the ground truth), perhaps minimizing mean squared error $\sum_{x_i \in X}(f(\alpha(x_{\mathrm{i}}))-p(x_{\mathrm{i}}))^2/|X|$ using a neural network or a support vector machine.

8 Concluding remarks

Anomaly detection problems arise in many applications and fields of study, and have been addressed by researchers using traditional statistical tools, data mining approaches, and problem-specific methodologies. Recent years have seen the development of many different algorithms for anomaly detection, and there is considerable potential in exploring their applications in banking and finance. This paper has conducted an overview of various classes of anomaly detection problems and algorithms, focusing on those that are relevant to the context of banking, such as customer and employee behavior tracking and various cases of fraudulent activities.

Traditional unsupervised anomaly detection algorithms have been based on nearest neighbor and clustering procedures, applied to multidimensional numerical data; the choice of distance measures is then critical. Heterogeneous density distributions and asymmetric cluster shapes hinder their applicability, but have been successfully addressed by more recently developed algorithms, which focus on local neighborhoods and relative proximity relationships between nearest neighbors. In some applications, ground truth data may be available, permitting the application of supervised or semi-supervised learning algorithms which use the available data at least to characterize normal data, against which anomalous data may be contrasted and identified.

An entirely different set of problems arises with time-stamped data streams that arrive over time. We then consider the identification of anomalies within a single stream, as well as compare one data stream against others to identify stream-level anomalies. Within a stream, we may distinguish among various special kinds of anomalies such as point anomalies, discords, rate anomalies, contextual anomalies, and others. Between multiple data streams, we may compare individual data streams against subsets of the entire set of streams for which data is available. Algorithms that accomplish these tasks include procedural algorithms such as Regression and Hidden Markov Models, as well as approaches based on transforming data to a different space that is more amenable to the application of anomaly detection algorithms.

These methodologies need substantial modification when the data is categorical or nominal, i.e., data attributes are not numeric. Data mining approaches may then be applicable, identifying frequent item-sets, with anomalies being defined as the infrequent ones, subject to some normalization procedures to account for the possibility that the items (within the infrequent item-sets) themselves occur rarely. Information compression methodologies based on the minimum description length principle have also been explored; transactions with high compression cost are then considered to be anomalies.

We then discussed different ways of combining multiple anomaly detection algorithms, so that we can identify anomalies that may be detected by some (but not all) individual algorithms.

Finally, we presented mathematical formulations of the anomaly detection problems, providing standards for solution quality against which specific anomaly detection algorithms can be evaluated. In addition to prediction, clustering, and classification perspectives, we also introduce decision-theoretic considerations that take asymmetric misclassification costs into consideration.

Although considerable work has been accomplished in all the areas discussed above, extensive empirical results exploring the application of anomaly detection algorithms to banking applications are few. This is a promising area for future study, using real datasets against which different algorithms can be evaluated. The need for such work is critical especially due to the increasing proliferation of cyber-crimes directed at the banking and financial industry, affecting millions of customers, and threatening to drastically diminish the trust in existing financial infrastructure, systems and platforms.

References

1. Abfalg, J., Kriegel, H. P., Kröger, P., Kunath, P., Pryakhin, A., & Renz, M.: Similarity search on time series based on threshold queries. In: Proceedings of the EDBT. 276-294 (2006)
2. Aggarwal, C. C.: Outlier analysis. In Data mining. Springer, 237-263 (2015)
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. Advances in Knowledge Discovery and Data Mining. **12**(1), 307–328 (1996)
4. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. of the 20th Int. Conf. on Very Large Data Bases. VLDB. **1215**, 487–499 (1994)
5. Agyemang, M., Barker, K., Alhajj., R.: A comprehensive survey of numeric and symbolic outlier mining techniques. Intelligent Data Analysis. **10**(6), 521–538 (2006)

6. Akoglu, L., Tong, H., Vreeken, J., Faloutsos., C.: Fast and reliable anomaly detection in categorical data. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. 415–424, ACM (2012)
7. Amari., S.: Field theory of self-organizing neural nets. IEEE Transactions on Systems, Man, and Cybernetics, **(5)**, 741–748 (1983)
8. Amer, M., Goldstein, M., and Abdennadher, S.: Enhancing one class support vector machines for unsupervised anomaly detection. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. 8–15, ACM (2013)
9. Donald, J., Berndt, Clifford, J.: Using dynamic time warping to find patterns in time series. In: AAAI Working Notes of the Knowledge Discovery in Databases Workshop, 359–370 (1994)
10. Bonanno, G., Lillo, F., Mantegna., R.N.: High-frequency cross-correlation in a set of stocks. Quantitative Finance. 96–104 (2001)
11. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 144–152. ACM (1992)
12. Breunig, M.M., Kriegel, H., Ng, R.T., Lof, J.S.: Identifying density-based local outliers. In ACM sigmod record, **29**, 93–104, ACM (2000)
13. Bu, Y., Leung, T., Fu, A.W., Keogh, E., Pei, J., Wat, S.M.: Finding top-k discords in time series database. In: Proceedings of the 2007 SIAM International Conference on Data Mining. 449–454 (2007)
14. Carpenter, G.A., Grossberg, S.: Adaptive resonance theory. In Encyclopaedia of Machine Learning and Data Mining, 1–17, Springer (2016)
15. Chair, Z. and Varshney, P.K.: Optimal data fusion in multiple sensor detection systems. IEEE Transactions on Aerospace and Electronic Systems, **(1)**, 98–101(1986)
16. Chan, K.P. and Fu, A.W.C.: Efficient time series matching by wavelets. In: Proceeding of the 15th International Conference on Data Engineering, Sydney, Australia, 126-127 (1999)
17. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection. ACM Computing Surveys, **41**(3), 1–58 (2009)
18. Chandola, V, Banerjee, A., Kumar, V.: Anomaly detection for discrete sequences: A survey. IEEE Transactions on Knowledge and Data Engineering, **24**(5), 823–839 (2012)
19. Chang, W., Chang, J.: Using clustering techniques to analyze fraudulent behavior changes in online auctions. In: Proceedings of the International Conference on Networking and Information Technology, 34–38, IEEE (2010)
20. Cheboli, D.: Anomaly detection of time series. PhD thesis, University of Minnesota (2010)
21. Chen, L, Raymond, Ng.: On the marriage of lP-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases. 792–803 (2004)
22. Chen, Y., Nascimento, M.A., Chin, B., Anthony, O., Tung, K. H.: Spade: On shape-based pattern detection in streaming time series. In: Proceedings of the 23rd International Conference on Data Engineering, Istanbul, Turkey. 786–795 (2007)
23. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation. **6**(2),182–197 (2002)
24. Deng, Q., Mei, G.: Combining self-organizing map and k-means clustering for detecting fraudulent financial statements. In: Proceedings of the International Conference on Granular Computing, 126–131. IEEE (2009)
25. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. In: Proceedings of the VLDB Endowment, 1542–1552 (2008)
26. Du, K., Swamy, MNS.: Radial basis function networks. In Neural Networks and Statistical Learning. 299–335, Springer (2014)
27. Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery And Data Mining. 213–220, ACM (2008)
28. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection. In: Applications of Data Mining In Computer Security, 77–101, Springer (2002)
29. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In Proceedings of the ACM SIGMOD International Conference on Management of data, New York, NY, USA, 419 – 429, ACM (1994)
30. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of online learning and an application to boosting. In: Proceedings of the European Conference On Computational Learning Theory, 23–37, Springer (1995)
31. Fritzke, B.: Growing cell structures a self-organizing network for unsupervised and supervised learning. Neural networks. **7**(9), 1441–1460 (1994)

32. Fujimaki, R., Yairi, T., Machida, K.: An anomaly detection method for spacecraft using relevance vector learning. Advances in Knowledge Discovery and Data Mining, 785–790 (2005)
33. Ghosh, S., Reilly, D.L.: Credit card fraud detection with a neural-network. In: Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, **3**, 621–630, IEEE (1994)
34. Grünwald, P.D.: The minimum description length principle. MIT Press (2007)
35. Guha, S., Mishra, N., Roy, G., Schrijvers, O.: Robust random cut forest based anomaly detection on streams. In: proceedings of the International Conference on Machine Learning, 2712–2721 (2016)
36. Gupta, M., Gao, J., Aggarwal, C.C., Han, J.: Outlier detection for temporal data: A survey. IEEE Transactions on Knowledge and Data Engineering, **26**(9), 2250–2267, (2014)
37. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review, **22**(2), 85–126 (2004)
38. Hsu, C., Chang, C., Lin, C. et al.: A practical guide to support vector classification. (2003)
39. Huang, H., Mehrotra, K., Mohan, C.K.: An online anomalous time series detection algorithm for univariate data streams. In: Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. 151–160. Springer (2013)
40. Huang, H., Mehrotra, K., Mohan, C.K.: Rank-based outlier detection. Journal of Statistical Computation and Simulation, **83**(3) ,518–531(2013)
41. Huang, H., Mehrotra, K., Mohan, C.K.: Detection of anomalous time series based on multiple distance measures. In: Proceedings of the 28th International Conference on Computers and Their Applications, Honolulu, Hawaii, USA, (2013)
42. Issa, H., Vasarhelyi, M.A., Application of anomaly detection techniques to identify fraudulent refunds, (2011)
43. Jin, W., Tung, A. KH., Han, J., Wang, W.: Ranking outliers using symmetric neighbourhood relationship. Advances in Knowledge Discovery and Data Mining, 577–593, Springer (2006)
44. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **3**, 358–386 (2005)
45. Keogh, E., Folias, T.: The ucr time series data mining archive, (2002)
46. Keogh, E., Lin, J.: Hot SAX: Efficiently finding the most unusual time series subsequence. In: Proceedings of the 5th IEEE International Conf. on Data Mining, Houston, Texas, 226 – 233 (2005)
47. Keogh, E., Lin, J., Fu, A.: Hot sax: Efficiently finding the most unusual time series subsequence. In: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 8, IEEE (2005)
48. Keogh, E., Lin, J., Lee, S., Herle, H.V.: Finding the most unusual time series subsequence: Algorithms and applications. Knowledge and Information Systems, **11**(1), 1–27 (2007)
49. Kohonen, T: The self-organizing map. In: Proceedings of the IEEE, **78**(9),1464–1480 (1990)
50. Kohonen, T.: Essentials of the self-organizing map. Neural Networks, **37**, 52–65 (2013)
51. Lin, J., Keogh, E., Li, W., Lonardi, S.: Experiencing sax: A novel symbolic representation of time series. Data Mining and knowledge discovery, **15**(2), 107 (2007)
52. Ma, J., Theiler, J., Perkins, S.: Accurate on-line support vector regression. Neural computation. **15**(11), 2683–2703 (2003)
53. Markou, M., Singh, S. Novelty detection: a review part 1: Statistical approaches. Signal processing. **83** (12), 2481–2497 (2003)
54. Markou, M., Singh, S.: Novelty detection: a review part 2: Neural network based approaches. Signal Processing. **83**(12), 2499–2521 (2003)
55. Mehrotra, K., Mohan, C.K., Ranka, S.: Elements of artificial neural networks. MIT Press (1997)
56. Moayedi, H.Z., Masnadi-Shirazi, MA.: Arima model for network traffic prediction and anomaly detection. In: Proceedings of the International Symposium on Information Technology, ITSim.**4**, 1–6. IEEE (2008)
57. Park, J., Sandberg, I.W.: Universal approximation using radial basis-function networks. Neural computation. **3**(2), 246–257 (1991)
58. Patcha, A., Park, J: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks. **51**(12), 3448–3470 (2007)
59. Pelleg, D., Moore, A.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Proceedings of the 17th International Conf. on Machine Learning, 727–734, Morgan Kaufmann (2000)
60. Marco, AF., Pimentel, Clifton, DA., Clifton, L., Tarassenko, L.: A review of novelty detection. Signal Processing. **99**, 215–249 (2014)
61. Pincombe, B.: Anomaly detection in time series of graphs using arma processes. Asor Bulletin. **24** (4), 2 (2005)

62. Protopapas, P., Giammarco, JM., Faccioli, L., Struble, MF., Dave, R., and Alcock, C.: Finding outlier light curves in catalogues of periodic variable stars. Monthly Notices of the Royal Astronomical Society. **369** (2), 677–696 (2006)
63. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, b., Zhu,Q., Zakaria, J., and Keogh, E. : Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 262–270 (2012)
64. Sabau, A.S.: Survey of clustering based financial fraud detection research. Informatica Economica. **16** (1), 110 (2012)
65. Siebes, A., Vreeken, J., Leeuwen, M.V.: Item sets that compress. In: Proceedings of the SIAM International Conference on Data Mining. 395–406 (2006)
66. Smets, K., Vreeken, J.: The odd one out: Identifying and characterising anomalies. In: Proceedings of the SIAM international conference on data mining. 804–815 (2011)
67. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing. **14**(3), 199–222 (2004)
68. Tang, J., Chen, Z., Fu, A.W., Cheung, D.: A robust outlier detection scheme for large data sets. In: Proceedings of the 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Citeseer (2001)
69. Thiprungsri, S., Vasarhelyi, M.A.: Cluster analysis for anomaly detection in accounting data: An audit approach (2011)
70. Varshney, P.K.: Distributed detection and data fusion. Springer Science & Business Media (2012)
71. Li, W., Nitin, K., Lolla, V.N., Keogh, E.J., Lonardi, S., Ratanamahatana, C.: Assumption-free anomaly detection in time series. In: SSDBM, **5**, 237–242 (2005)
72. Zhao, Z., Mehrotra, K.G., Mohan, C.K.: Ensemble algorithms for unsupervised anomaly detection. In: Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. 514–525, Springer (2015)
73. Zhao, Z., Mehrotra, K.G., Mohan, C.K.: Card. Evolution of space-partitioning forest for anomaly detection. In: Proceedings of the Workshop on Genetic Programming Theory and Practice. (2017)
74. Zhu, X. Semi-supervised learning literature survey (2005)