Outlier Detection for Temporal Data: A Survey

Manish Gupta, Jing Gao, *Member, IEEE,* Charu C. Aggarwal, *Fellow, IEEE,* and Jiawei Han, *Fellow, IEEE*

Abstract—In the statistics community, outlier detection for time series data has been studied for decades. Recently, with advances in hardware and software technology, there has been a large body of work on temporal outlier detection from a computational perspective within the computer science community. In particular, advances in hardware technology have enabled the availability of various forms of temporal data collection mechanisms, and advances in software technology have enabled a variety of data management mechanisms. This has fueled the growth of different kinds of data sets such as data streams, spatio-temporal data, distributed streams, temporal networks, and time series data, generated by a multitude of applications. There arises a need for an organized and detailed study of the work done in the area of outlier detection with respect to such temporal datasets. In this survey, we provide a comprehensive and structured overview of a large set of interesting outlier definitions for various forms of temporal data, novel techniques, and application scenarios in which specific definitions and techniques have been widely used.

Index Terms—Temporal outlier detection, time series data, data streams, distributed data streams, temporal networks, spatio-temporal outliers, applications of temporal outlier detection, network outliers

1 Introduction

Outlier detection is a broad field, which has been studied in the context of a large number of application domains. Aggarwal [1], Chandola *et al*. [2], Hodge *et al*. [3] and Zhang *et al*. [4] provide an extensive overview of outlier detection techniques. Outlier detection has been studied in a variety of data domains including high-dimensional data [5], uncertain data [6], streaming data [7]–[9], network data [9]–[13] and time series data [14], [15]. Outlier detection is very popular in industrial applications, and therefore many software tools have been built for efficient outlier detection, such as R (packages 'outliers'[1] and 'outlierD' [16]), SAS[2], RapidMiner[3], and Oracle datamine[4].

The different data domains in outlier analysis typically require dedicated techniques of different types. Temporal outlier analysis examines anomalies in the behavior of the data *across time*. Some motivating examples are as follows.

1. http://cran.r-project.org/web/packages/outliers/outliers.pdf
2. http://www.nesug.org/Proceedings/nesug10/ad/ad07.pdf
3. http://www.youtube.com/watch?v=C1KNb1Kw-As
4. http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/anomalies.htm

- *M. Gupta is with Microsoft, Gachibowli, Hyderabad 500032, India. E-mail: gmanish@microsoft.com.*
- *J. Gao is with the University at Buffalo, State University of New York, Buffalo, NY 14260 USA. E-mail: jing@buffalo.edu.*
- *C. C. Aggarwal is with the IBM T.J. Watson Research Center, Yorktown, NY 10598 USA. E-mail: charu@us.ibm.com.*
- *J. Han is with University of Illinois at Urbana-Champaign, Urbana-Champaign, IL 61801 USA. E-mail: hanj@illinois.edu.*

Manuscript received 26 Feb. 2013; revised 15 Oct. 2013; accepted 27 Nov. 2013. Date of publication 15 Dec. 2013; date of current version 31 July 2014. Recommended for acceptance by J. Pei.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.184

- *Financial Markets:* An abrupt change in the stock market, or an unusual pattern within a specific window such as the *flash crash* of May 6, 2010 is an anomalous event which needs to be detected early in order to avoid and prevent extensive disruption of markets because of possible weaknesses in trading systems.
- *System Diagnosis:* A significant amount of data generated about the *system state* is discrete in nature. This could correspond to UNIX system calls, aircraft system states, mechanical systems, or host-based intrusion detection systems. The last case is particularly common, and is an important research area in its own right. Anomalies provide information about potentially threatening and failure events in such systems.
- *Biological Data:* While biological data is not temporal in nature, the placement of individual amino-acids is analogous to positions in temporal sequences. Therefore, temporal methods can be directly used for biological data.
- *User-action Sequences:* A variety of sequences abound in daily life, which are created by user actions in different domains. These include web browsing patterns, customer transactions, or RFID sequences. Anomalies provide an idea of user-behavior which is deviant for specific reasons (e.g., an attempt to crack a password will contain a sequence of *login* and *password* actions).

This broad diversity in applications is also reflected in the diverse formulations and data types relevant to outlier detection. A common characteristic of all temporal outlier analysis is that *temporal continuity* plays a key role in all these formulations, and unusual *changes*, *sequences*, or *temporal patterns* in the data are used in order to model outliers. In this sense, time forms the *contextual variable* with respect to which all analysis is performed. Temporal

1041-4347 © 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

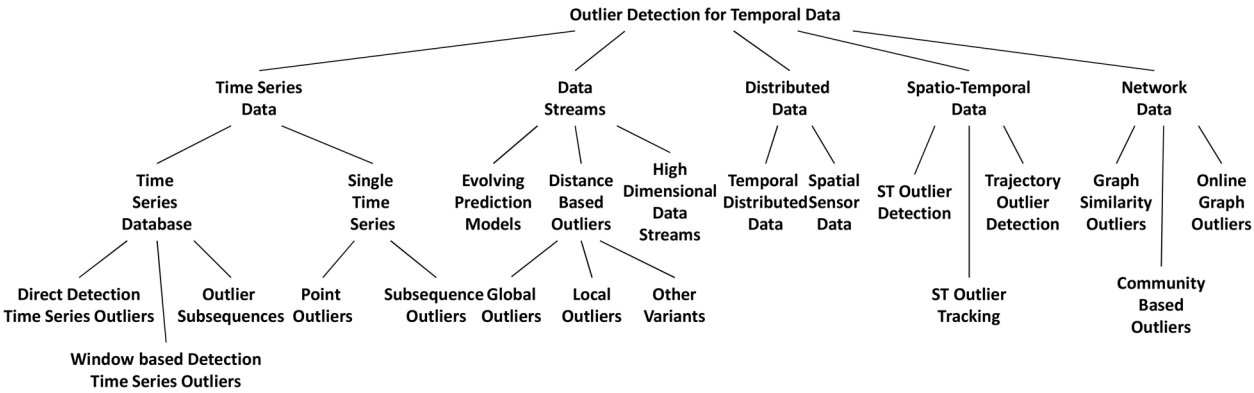

Fig. 1. Organization of the survey.

outlier analysis is closely related to change point detection, and event detection, since these problems represent two instantiations of a much broader field. The problem of *forecasting* is closely related to many forms of temporal outlier analysis, since outliers are often defined as deviations from *expected* values (or *forecasts*). Nevertheless, while forecasting is a useful *tool* for many forms of outlier analysis, the broader area seems to be much richer and multi-faceted.

Different Facets of Temporal Outlier Analysis: Outlier analysis problems in temporal data may be categorized in a wide variety of ways, which represent different facets of the analysis. The area is so rich that no single type of abstract categorization can fully capture the complexity of the problems in the area, since these different facets may be present in an arbitrary combination. Some of these facets are as follows.

- *Time-series vs Multidimensional Data Facet:* In time-series data (e.g., sensor readings) the importance of temporal continuity is paramount, and all analysis is performed with careful use of reasonably small windows of time (the contextual variable). On the other hand, in a multi-dimensional data stream such as a text newswire stream, an application such as *first-story detection*, might not rely heavily on the temporal aspect, and thus the methods are much closer to standard multi-dimensional outlier analysis.
- *The Point vs Window Facet:* Are we looking for an unusual data *point* in a temporal series (e.g., sudden jump in heart rate in ECG reading), or are we looking for an unusual pattern of changes (contiguous ECG pattern indicative of arrythmia)? The latter scenario is usually far more challenging than the former. Even in the context of a multi-dimensional data stream, a single point deviant (e.g., first story in a newswire stream) may be considered a different kind of outlier than an aggregate change point (e.g., sudden change in the aggregate distribution of stories over successive windows).
- *The Data Type Facet:* Different kinds of data such as continuous series (e.g., sensors), discrete series (e.g., web logs), multi-dimensional streams (e.g., text streams), or network data (e.g., graph and social streams) require different kinds of dedicated methods for analysis.
- *The supervision facet:* Are previous examples of anomalies available? This facet is of course common to all forms of outlier analysis, and is not specific to the temporal scenario.

These different facets are largely independent of one another, and a large number of problem formulations are possible with the use of a combination of these different facets. Therefore, this paper is largely organized by the facet of data type, and examines different kinds of scenarios along this broad organization.

Specific Challenges for Outlier Detection for Temporal Data: While temporal outlier detection aims to find rare and interesting instances, as in the case of traditional outlier detection, new challenges arise due to the nature of temporal data. We list them below.

- A wide variety of anomaly models are possible depending upon the specific data type and scenario. This leads to diverse formulations, which need to be designed for the specific problem. For arbitrary applications, it may often not be possible to use off-the-shelf models, because of the wide variations in problem formulations. This is one of the motivating reasons for this survey to provide an overview of the most common combinations of facets explored in temporal outlier analysis.
- Since new data arrives at every time instant, the scale of the data is very large. This often leads to processing and resource-constraint challenges. In the streaming scenario, only a single scan is allowed. Traditional outlier detection is much easier, since it is typically an offline task.
- Outlier detection for temporal data in distributed scenarios poses significant challenges of minimizing communication overhead and computational load in resource-constrained environments.

In this work, we aim to provide a comprehensive and structured overview of outlier detection techniques for temporal data. Fig. 1 shows the organization of the survey with respect to the data type facet. For each data type, we discuss specific problem classes in various subsections. We begin with the easiest scenario for temporal data – discrete time series data (Section 2). However a lot of data gets sampled over very short time intervals, and keeps flowing in infinitely leading to data streams. We will study techniques for outlier detection in streams in Section 3.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

Often times, data is distributed across multiple locations. We study how to extract global outliers in such distributed scenarios in Section 4. For some applications like environmental data analysis, data is available over a continuum of both space and time dimensions. We will study techniques to handle such data in Section 5. Finally, networks can capture very rich semantics for almost every domain. Hence, we will discuss outlier detection mechanisms for network data in Section 6. We will also present a few applications where such temporal outlier detection techniques have been successfully employed in Section 7. The conclusions are presented in Section 8.

2 Time Series Outlier Detection

A significant amount of work has been performed in the area of time series outliers. Parametric models for time series outliers [15] represents the first work on outlier detection for time series data. Several models were subsequently proposed in the statistics literature, including autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), vector autoregression (VARMA), CUmulative SUM Statistics (CUSUM), exponentially weighted moving average, etc. While a detailed exposition is beyond the scope of this survey, we will provide an overview of the key ideas in this topic, especially from a computer science perspective. We direct the reader to [17]–[19] for further reading from the statistics point of view. In this section, we will focus on two main types of outlier detection techniques for time series studied in the data mining community. The first part concerns techniques to detect outliers over a database of time series, whereas the second part deals with outliers within a single time series.

2.1 Outliers in Time Series Databases

Given a time series database, we will discuss methods to identify a few time series sequences as outliers, or to identify a subsequence in a test sequence as an outlier. An outlier score for a time series can be computed directly, or by first computing scores for overlapping fixed size windows and then aggregating them. We discuss these techniques in this subsection.

2.1.1 Direct Detection of Outlier Time Series

Given: A database of time series
Find: All anomalous time series

It is assumed that most of the time series in the database are normal while a few are anomalous. Similar to traditional outlier detection, the usual recipe for solving such problems is to first learn a model based on all the time series sequences in the database, and then compute an outlier score for each sequence with respect to the model. The model could be *supervised* or *unsupervised* depending on the availability of training data.

Unsupervised Discriminative Approaches

Discriminative approaches rely on the definition of a similarity function that measures the similarity between two sequences. Once a similarity function is defined, such approaches cluster the sequences, such that within-cluster similarity is maximized, while between-cluster similarity is minimized. The anomaly score of a test time series sequence is defined as the distance to the centroid (or medoid) of the closest cluster. The primary variation across such approaches, are the choice of the similarity measure, and the clustering mechanism.

Similarity Measures: The most popular sequence similarity measures are the simple *match count based sequence similarity* [20], and the normalized length of the longest common subsequence (LCS) [21]–[24]. The advantage of the former is its greater computational efficiency, whereas the latter can adjust to segments in the sequences containing noise, but is more expensive because of its dynamic programming methodology.

Clustering Methods: Popular clustering methods include k-Means [25], EM [26], phased k-Means [27], dynamic clustering [24], k-medoids [21], [22], single-linkage clustering [28], clustering of multi-variate time series in the principal components space [29], one-class SVM [30]–[33] and self-organizing maps [34]. The choice of the clustering method is application-specific, because different clustering methods have different complexity, with varying adaptability to clusters of different numbers, shapes and sizes.

Unsupervised Parametric Approaches

In unsupervised parametric approaches, anomalous instances are not specified, and a summary model is constructed on the base data. A test sequence is then marked anomalous if the probability of generation of the sequence from the model is very low. The anomaly score for the entire time series is computed in terms of the probability of each element. Popular models include Finite state automata (FSA), Markov models and Hidden Markov Models (HMMs).

FSA can be learned from length l subsequences in normal training data. During testing, all length l subsequences can be extracted from a test time series and fed into the FSA. An anomaly is then detected if the FSA reaches a state from where there is no outgoing edge corresponding to the last symbol of the current subsequence. FSAs have been used for outlier detection in [23], [35]–[37]. The methods to generate the state transition rules depend on particular application domains.

Some Markov methods store conditional information for a fixed history size=k, while others use a variable history size to capture richer temporal dependencies. Ye [38] proposes a technique where a Markov model with k=1 is used. In [39], [40], the conditional probability distribution (CPD) are stored in probabilistic suffix trees (PSTs) for efficient computations. Sparse Markovian techniques estimate the conditional probability of an element based on symbols within the previous k symbols, which may not be contiguous or immediately preceding to the element [41], [42].

HMMs can be viewed as *temporal dependency-oriented* mixture models, where hidden states and transitions are used to model temporal dependencies among mixture components. HMMs do not scale well to real life datasets. The training process may require judicious selection of the model, the parameters, and initialization values of the parameters. On the other hand, HMMs are interpretable and theoretically well motivated. Approaches that use HMMs for outlier detection include [23], [43]–[46].

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

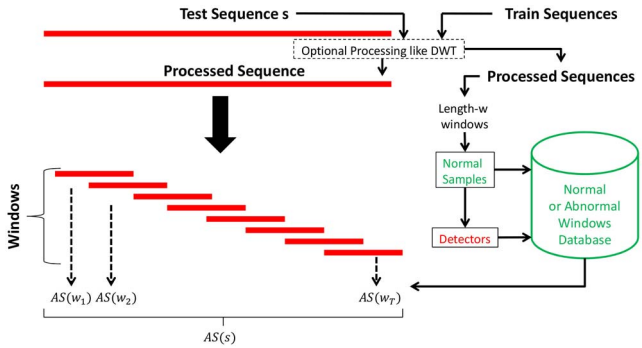

Fig. 2. Window-based time series outlier detection.

Unsupervised OLAP Based Approach

Besides traditional uni-variate time series data, richer time series are quite popular. For example, a time series database may contain a set of time series, each of which are associated with multi-dimensional attributes. Thus, the database can be represented using an OLAP cube, where the time series could be associated with each cell as a measure. Li *et al*. [47] define anomalies in such a setting, where given a probe cell c, a descendant cell is considered an anomaly if the trend, magnitude or the phase of its associated time series are significantly different from the expected value, using the time series for the probe cell c.

Supervised Approaches

In the presence of labeled training data, the following supervised approaches have been proposed in the literature: positional system calls features with the RIPPER classifier [48], subsequences of positive and negative strings of behavior as features with string matching classifier [34], [49], neural networks [50]–[54], Elman network [53], motion features with SVMs [55], bag of system calls features with decision tree, Naïve Bayes, SVMs [56]. Sliding window subsequences have also been used as features with SVMs [57], [58], rule based classifiers [59], and HMMs [44].

2.1.2 Window-Based Detection of Outlier Time Series

Given: A database of time series
Find: All anomalous time windows, and hence anomalous time series

Compared to the techniques in the previous subsection, the test sequence is broken into multiple overlapping subsequences (windows). The anomaly score is computed for each window, and then the anomaly score (AS) for the entire test sequence is computed in terms of that of the individual windows. Window-based techniques can perform better localization of anomalies, compared to the techniques that output the entire time series as outliers directly. These techniques need the window length as a parameter. Windows are called fingerprints, pattern fragments, detectors, sliding windows, motifs, and n-grams in various contexts. In this methodology, the techniques usually maintain a normal pattern database, but some approaches also maintain a negative pattern or a mixed pattern database. Fig. 2 shows a general sketch of the method.

Normal Pattern Database Approach

In this approach, normal sequences are divided into size w overlapping windows. Each such window subsequence is stored in a database with its frequency. For a test sequence, subsequences of size w are obtained, and those subsequences that do not occur in normal database are considered mismatches. If a test sequence has a large number of mismatches, it is marked as an anomaly [44], [49], [51], [53], [60]. Rather than looking for exact matches, if a subsequence is not in the database, soft mismatch scores can also be computed [20], [61]–[63].

Besides contiguous window subsequences, a lookahead based method can also be used for building a normal database [64]. For every element in every normal sequence, the elements occurring at distance 1,2,. . ., k in the sequence are noted. A normal database of such occurrences is created. Given a new test sequence, a lookahead of the same size k is used. Each pair of element occurrence is checked with the normal database, and the number of mismatches is computed.

Negative and Mixed Pattern Database Approaches

Besides the dictionaries for normal sequences, anomaly dictionaries can also be created [34], [50], [65]–[67]. All normal subsequences of length w are obtained from the input string. Next, all sequences of size w not in the set are considered detectors or negative subsequences. Detectors can be generated randomly or by using some domain knowledge of situations that are not expected to occur in normal sequences. A test sequence is then monitored for presence of any detector. If any detector matches, the sequence can be considered an outlier.

2.1.3 Outlier Subsequences in a Test Time Series

Given: A database of time series D and a test time series t
Find: An outlier subsequence (or a pattern) p in t

The anomaly score for a pattern p can be computed as the difference between the frequency of pattern p in test time series and the expected frequency in database D. Keogh *et al*. [68] define a soft match version of the problem where the frequency of pattern p in the database D is defined using the largest number l such that every subsequence of p of length l occurs at least once in D. Another form of soft matching is defined in [69], where rather than an exact match of pattern p, any permutation of p is also considered a match. To make the computations efficient, the TARZAN algorithm was proposed which exploits suffix trees [68], [70], [71]. Also Gwadera *et al*. [72], [73] have proposed Interpolated Markov Models (IMM) to efficiently compute the match score of a pattern or its permutations within any time series.

2.2 Outliers Within a Given Time Series

Given a single time series, one can find particular elements (or time points) within the time series as outliers, or one can also find subsequence outliers. In this subsection, we will discuss techniques for both these cases.

2.2.1 Points as Outliers

Given: A time series t
Find: Outlier points in t

Various methodologies have been proposed to find outlier points for a time series. A large number of prediction models and profile based models have been proposed. An

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

information-theoretic compression based technique has also been proposed to find "deviants". We will discuss these techniques below. Apart from these, a number of clustering and classification approaches described in Section 2.1.1, can also be used to detect point outliers.

Prediction Models

The outlier score for a point in the time series is computed as its deviation from the predicted value by a summary *prediction model*. The primary variation across models, is in terms of the particular prediction model used.

Given a time series, one can predict the value at time t as a median of the values in the size-$2k$ window from $t-k$ to $t+k$ [74], or as an average of all the points in the cluster that the value at time t maps to [75], or using regression models. Single-layer linear network predictor (or AR model) has been used in a large number of studies including [75]. Other prediction models include Multilayer perceptron (MLP) predictor [75] and support vector regression [76]. Mixture transition distribution (MTD) have been proposed for outlier detection for general non-Gaussian time series [77]. Tsay *et al*. [78] propose a vector ARIMA model to identify additive outliers, innovation outliers, level shifts, and temporary changes from multi-variate time series. Besides individual points, multiple outliers can also be discovered using prediction models, e.g., using re-weighted maximum likelihood estimates [79] or using Gibbs sampling and block interpolation [80].

There are many variants of this technique. Outlier-aware variants of these prediction models estimate model parameters and outliers together [81]–[84]. In multivariate time series, prediction could be made for all constituent time series. In [85], to compute outliers for multivariate time series, testing for outliers is done only in some smartly selected projection directions rather than testing the multivariate series directly to compute outliers.

Profile Similarity based Approaches

These approaches maintain a normal profile and then compare a new time point against this profile to decide whether it is an outlier. For example, for multiple OS performance metric time series, the Tiresias system [86] maintains a normal profile and also a variance vector. Any new data point is compared both with the normal profile and the variance vector to compute its anomaly score. Here the profile is the actual smoothed time series data from past data. In [87], a neural network is used to maintain the normal profile and an estimation is made for the next value in the sensor stream based on this profile. We will discuss more profile based techniques in Section 3 in the context of data streams.

Deviants

Deviants are outlier points in time series from a minimum description length (MDL) point of view [88]. If the removal of a point P from the time sequence results in a sequence that can be represented *significantly* more succinctly than the original one, then the point P is a deviant. These information-theoretic models explore the space-deviation tradeoff by fixing the deviation, rather than fixing the space, as in conventional models. Thus the problem is to find points whose removal results in a histogram representation with a lower error bound than the original, even after the number of buckets has been reduced to account for the separate storage of these deviant points. Jagadish *et al*. [88] propose a dynamic programming mechanism to solve the problem. Muthukrishnan *et al*. [89] make the observation that for any bucket, the optimal set of k deviants within the bin always consists of the l highest and remaining $k-l$ lowest values for some $l \leq k$. Then, they propose an approximation to the dynamic programming-based solution, that maintains a partial solution only for a few interspersed indexes of the time series rather than for each value.

2.2.2 Subsequences as Outliers

Given: A time series t

Find: Outlier subsequences in t

In the previous subsection, we looked at techniques to identify *point* outliers. In this subsection, we will visit mechanisms to identify outlier *subsequences* in time series.

Given a time series T, the subsequence D of length n beginning at position l is said to be the discord (or outlier) of T if D has the largest distance to its nearest non-overlapping match [90]. The brute force solution is to consider all possible subsequences $s \in S$ of length n in T and compute the distance of each such s with each other non-overlapping $s' \in S$. Top-K pruning can be used to make this computation efficient. Subsequence comparisons can be smartly ordered for effective pruning using various methods like heuristic reordering of candidate subsequences [91], locality sensitive hashing [92], Haar wavelet and augmented tries [93], [94], and SAX with augmented trie [95]. To compute the distance between subsequences, most methods use Euclidean distance while Compression based Dissimilarity Measure (CDM) is used as a distance measure in [96]. Yankov *et al*. [97] solve the problem for a large time series stored on the disk.

Chen *et al*. [98] define the subsequence outlier detection problem for an unequal interval time series which is a time series with values sampled at unequal time intervals. For such a time series, a pattern is defined as a subsequence of two consecutive points. A pattern p is called an anomaly, if there are very few other patterns with the same slope and the same length. To identify anomalies at multiple scales, the Haar transform is used. Haar transform is also used in [99] to identify multi-level trends and anomalies.

Besides the aforementioned definitions of outlier subsequences, some more variants have also been discussed in the literature. For example, in [100] a lead and a lag window are defined as adjacent subsequences. The two windows could be of any length. The subsequence represented by the lead window is an anomaly, if its similarity with the lag window is very low. They measure the similarity using chaos bitmaps. As another example, Zhu *et al*. [101] compute subsequences of length w with a very high aggregate value as outliers. They use a novel data structure called *Shifted Wavelet Tree* to solve the problem.

3 Outlier Detection for Stream Data

Compared to static data, streaming data does not have a fixed length. Streams can be a time-series or multidimensional. Temporal dependencies for multidimensional streams, are used differently than in time-series. Rather

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

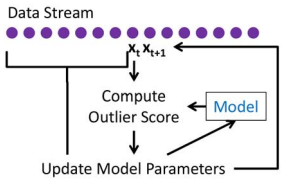

Fig. 3. Stream anomaly detection.

than using time-series *forecasting* methods, these methods are closer to conventional multidimensional models, but with a temporal component, which accounts for temporal drift, and deviations.

3.1 Evolving Prediction Models

Given: A multidimensional data stream s
Find: Outlier points in s

Evolving prediction models are models in which the parameters or components of the model are updated as new data arrives in order to better capture the normal trends in the data. In this subsection, we will discuss a few such models.

Online Sequential Discounting

Yamanishi *et al*. [102], [103] present SmartSifter which employs an online discounting learning algorithm to incrementally learn the probabilistic mixture model, with a decay factor to account for drift. The outlier score for a data point is computed in terms of the probabilistic fit value to the learned mixture model. Such mixture models are used for conventional multidimensional data, but without the adjustments for incremental updates and temporal decay. Fig. 3 shows an illustration of the method.

For categorical variables, they propose the Sequentially Discounting Laplace Estimation (SDLE) algorithm. In SDLE, cells are created by partitioning the space of all values of all categorical variables. The probability of a symbol in a cell is the number of occurrences of that symbol in that cell divided by the total data points with the Laplace smoothing. When a new data point arrives, the count for all the cells are adjusted with temporal discounting and appropriate Laplace smoothing is applied.

For continuous variables, they propose two models: an independent model and a time series model. The independent model is a Gaussian mixture model in the parametric case and a kernel mixture model in the non-parametric case. For learning the Gaussian mixture model (GMM), they provide the Sequentially Discounting EM (SDEM) algorithm which is essentially an incremental EM algorithm with discounting of effect of past examples. For learning the kernel mixture model, Yamanishi *et al*. provide Sequentially Discounting Prototype Updating (SDPU) algorithm where the coefficients of the mixture and the variance matrix are fixed and so it iteratively learns only the means of the kernels or the prototypes. For learning the time series model, they provide the Sequentially Discounting AR (SDAR) algorithm which learns the AR model parameters iteratively with time discounting.

The online discounting behavior is also used to update the "normal" distribution in SRI's Emerald system [104], by giving more weight to recent data. Online discounting has also been used for mixed attribute data streams, to maintain the frequency of an itemset (a set of attribute values) [11], where a point is called an outlier if it shares the attribute values with none or very few other points. Exponentially greater importance (another form of online discounting) is given to the recent points to learn a polynomial function in StreamEvent [8] which is an algorithm to efficiently compute outliers from multiple synchronous streams, though this model is based on the time-series scenario.

Dynamic Cluster Maintenance

Other than the online discounting methods, large number of methods use dynamically maintained cluster models for computing outliers from data streams. For example, normalized length of the longest common subsequence (LCS) has been used as the sequence similarity measure for dynamic clustering by Sequeira *et al*. [24]. In the text domain, online clustering methods were proposed in [105] to detect outliers.

Dynamic Bayesian Networks

Sometimes updating the parameters of the model is not enough. The model can be modified by itself to incorporate drifts in the data stream. Hill *et al*. [106] present an approach that uses Dynamic Bayesian networks which are Bayesian networks with network topology that evolves over time, adding new state variables to represent the system state at the current time t. They present two strategies for detecting anomalous data: Bayesian credible interval (BCI) and maximum a posteriori measurement status (MAP-ms). The first method uses an HMM model, where Kalman filtering is used to sequentially infer the posterior distributions of hidden and observed states as new measurements become available from the sensors. The posterior distribution of the observed state variables is then used to construct a Bayesian credible interval for the most recent set of measurements. Any measurements that fall outside of the p% Bayesian credible interval can be classified as anomalous. In the second method, a 2-layered DBN is used. The status (e.g., normal/anomalous) of each sensor measurement is also modeled as a hidden state. The maximum a posteriori estimate, (i.e., the most likely value given the posterior distribution) of the hidden state variable, indicating the measurement status, can be used to classify the sensor measurements as normal or anomalous.

3.2 Distance-Based Outliers for Sliding Windows

Given: A data stream s
Find: Distance based outliers in any time window of s

Besides defining outliers for data streams using prediction models, one can also compute distance based outliers for data streams at any time instant. Given a set of points, a point o is called a $DB(k, R)$ distance outlier if there are less than k points within distance R from o [107]. Given a data stream, at any time point, one can discover outliers from the set of points lying within the current sliding window. Distance based outliers can be discovered both in a global as well as in a local sense on points within the current sliding window. Fig. 4 shows a 1-dimensional stream dataset with two time windows (t_3 to t_{18} and t_7 to t_{22}). Consider the window at time t_{18}. If we consider a point with $k < 4$ points within distance R as outliers, $o9$ (with k=4 neighbors $o5$, $o10$, $o14$ and $o15$) and $o11$ (with k=4 neighbors $o3$, $o4$, $o6$ and $o13$) are examples of inliers, while $o8$ (with only 2

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

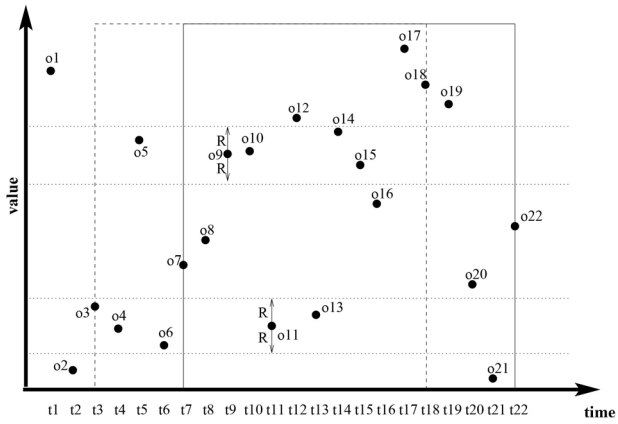

Fig. 4. Distance-based outliers for sliding windows [108].

neighbors $o7$ and $o16$) and $o17$ (with only 1 neighbor $o18$) are examples of outliers.

3.2.1 Distance-Based Global Outliers

As the sliding window for a stream moves, old objects expire and new objects come in. Since objects expire over time, the number of preceding neighbors of any object decreases. Therefore, if the number of succeeding neighbors of an object is less than k, the object could become an outlier depending on the stream evolution. Conversely, since any object will expire before its succeeding neighbors, inliers having at least k succeeding neighbors will be inliers for any stream evolution. Such inliers are called safe inliers (e.g., in Fig. 4, for k=3, $o9$ is a safe inlier as it has 3 succeeding neighbors ($o10$, $o14$, $o15$) while $o11$ is not, as it has only 1 succeeding neighbor ($o13$)).

Angiulli *et al*. [108] propose an exact algorithm to efficiently compute distance outliers using a new data structure called Indexed Stream Buffer (ISB) which supports a range query. Further, they also propose an approximate algorithm which uses two heuristics: (a) It is sufficient to retain in ISB only a fraction of safe inliers, (b) Rather than storing the list of k most recent preceding neighbors, it is enough to store only the fraction of preceding neighbors, which are safe inliers to the total number of safe inliers.

Yang *et al*. [109] propose that maintaining all neighbor relationships across time may be very expensive. Therefore, abstracted neighbor relationships can be maintained. However, maintaining such cluster abstractions is expensive too. Hence, they exploit an important characteristic of sliding windows, namely the "predictability" of the expiration of existing objects. In particular, given the objects in the current window, the pattern structures that will persist in subsequent windows can be predicted by considering the objects (in the current window) that will participate in each of these windows only. These predicted pattern structures can be abstracted into "predicted views" of each future window. They propose an efficient algorithm which makes use of the predicted views to compute distance based outliers.

The problem of distance-based outlier detection for stream data can also be solved using dynamic cluster maintenance, as has been done for similar problems in [110], [111].

3.2.2 Distance-Based Local Outliers

Local Outlier Factor (LOF) is an algorithm for finding anomalous data points by measuring the local deviation of a given data point with respect to its neighbours [112]. The LOF algorithm developed for static data can be adapted to the incremental LOF problem (i.e., the stream setting) in three ways: (a) periodic LOF, i.e., apply LOF on entire data set periodically, or as (b) supervised LOF, i.e., compute the k-distances, local reachability density (LRD) and LOF values using training data and use them to find outliers in test data or as (c) iterated LOF where the static LOF algorithm can be re-applied every time a new data record is inserted into the data set. A better approach is proposed in [113], where the incremental LOF algorithm computes LOF value for each data record inserted into the data set and instantly determines whether the inserted data record is an outlier. In addition, LOF values for existing data records are updated if needed. Thus, in the insertion part, the algorithm performs two steps: (a) insertion of new record, when it computes the reachability distance, LRD and LOF values of a new point; (b) maintenance, when it updates k-distances, reachability distance, LRD and LOF values for affected existing points.

3.3 Outliers in High-Dimensional Data Streams

Zhang *et al*. [114] present Stream Projected Outlier deTector (SPOT), to deal with outlier detection problem in high-dimensional data streams. SPOT employs a window based time model and decaying cell summaries to capture statistics like Relative Density and Inverse Relative Standard Deviation of data points in the cell from the data stream. Sparse Subspace Template (SST), a set of top sparse subspaces obtained by unsupervised and/or supervised learning processes, is constructed in SPOT. These SSTs are the subspaces which are explored to find projected outliers effectively. Multi-Objective Genetic Algorithm (MOGA) is employed as an effective search method in unsupervised learning for finding outlying subspaces from training data. SPOT is also capable of online self-evolution, to cope with evolving dynamics of data streams.

4 Outlier Detection for Stream Data in Distributed Scenarios

Zhang *et al*. [4] provide a detailed survey for outlier detection techniques on wireless sensor networks. The main challenges for outlier detection in a distributed setting are as follows.

- Resource constraints: Energy, memory, computational capacity and communication bandwidth are all scarce.
- High communication cost: Communication cost is orders of magnitude greater than the computation costs.
- Distributed streaming data: Processing data online coming at different rates from multiple distributed sensors is non-trivial.
- Dynamics: Dynamic network topology, frequent communication failures, mobility and heterogeneity

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

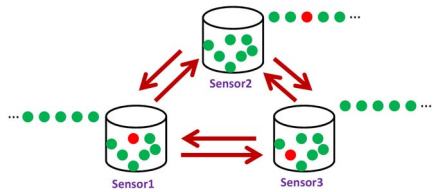

Fig. 5. Distributed data streams scenario.

of nodes are a few dynamic features of such datasets which are challenging to handle.
- Large-scale deployment: Traditional outlier detection algorithms are not easily scalable.
- Identifying outlier sources: It is important to make distinctions between errors, events and malicious attacks.

In a distributed stream setting, points are distributed across various nodes (sensors). Each sensor has an associated stream of incoming points and the aim is to find top few distance based outliers based on the global data (Fig. 5). In a distributed *spatio-temporal* setting, the position of sensors is also important when computing outliers. We discuss both settings in this section.

4.1 Distributed Temporal Data

Given: A set of sensors each with an associated data stream
Find: Distance/density based outliers based on global data
Sharing Local Outliers and Other Data Points: Let us first discuss an algorithm [115] for a distributed static setting, where each sensor holds a set of points and tries to compute the set of the global top-K outliers. An outlier detection algorithm A is available which outputs a ranking function R. R is anti-monotonic wrt data and R should be smooth, e.g., the distance to the k^{th} nearest neighbor, the average distance to the k nearest neighbors, etc. Consider a point x in dataset P. Consider a subset P_0 of P. If P_0 is the smallest subset such that $R(x, P) = R(x, P_0)$ then P_0 is called the support set of x over P.

Based on the above basic concepts, we discuss the algorithm flow as follows. Each sensor exchanges messages with its neighbors. Each sensor can apply A to compute top-K outliers for its knowledge where the *knowledge* of the sensor is its own points plus the points received in messages. Sensor p_i should send to p_j all of p_i's current outliers and their supports. If, for any of these points x, p_i cannot be certain that p_j has x (i.e. neither previous messages from p_i to p_j nor messages from p_j to p_i contain x), then x must be sent to p_j. Second, p_i may have points which would effect outliers previously sent by p_j, but these may not be accounted for in the first part. It suffices for p_i to send the support of all of the outliers in message from p_j to p_i which were not in that message. In a streaming setting, when a new point is sampled, data changes at the local sensor itself. This requires that the same calculation is made as in the case of a change in knowledge of the sensor due to receiving a message. If the algorithm needs to only consider points which were sampled recently (i.e. employ a sliding window), this can be implemented by adding a time-stamp to each point when it is sampled. Each node can retire old points regardless of where they were sampled and at no communication cost at all.

Sharing Local Outliers Only: Since transmitting the entire data or even the model to every node could be expensive, Otey *et al*. [116] propose that only the local outliers be exchanged between nodes. If all nodes agree that a point is an outlier, then we can assume that the point is a global outlier. The sites only communicate when some user-specified event occurs. Examples of such events include a user's query for the global outliers, when a node finishes processing a fixed number of points, or when a node finds a fixed number of outliers. Clearly, since there is no global computation, this is an approximate algorithm.
Sharing Local Outliers and Data Distributions: In this approach, data distributions are also shared along with the local outliers. The sensor network is organized in the form of a hierarchy. Kernel density estimators are used to approximate the sensor data distribution. Once the data distributions are estimated, the density of the data space around each value can be computed, and thus outliers can be determined. For distance based outliers, the goal is to identify, among all the sensor readings in a sliding window, those values that have very few close neighbors. The global distance based outliers are computed by sending the local model and the local outliers to the parent nodes. For density based method, Subramaniam *et al*. [117] propose the Multi Granularity Deviation Factor (MDEF) metric. A value is flagged as an outlier, if its MDEF is statistically significantly different from that of the local averages. To estimate global MDEF based outliers, each node sends the new observations to its parent with a probability f, and then the parent node communicates the new model to the lowest level of nodes through the intermediate nodes. The hierarchical architecture of a sensor network has also been proposed by Palpanas *et al*. [118]. The distribution of the sensed data is assumed to be the kernel density estimators. When the observed data points do not fit this model, they are considered as outliers.

4.2 Distributed Sensor Data Streams With Spatial Considerations

For some applications, the location of the sensors is also important. In such settings where data is available across space and time, one can find faulty sensors.
Given: A set of sensors each with (x,y) coordinates and an associated data stream
Find: Outlier sensors and outlier regions

In this setting, each sensor is associated with a data stream based on a part of a source signal received at its location. Each sensor thus receives data which is a combination of the source signal plus some noise where a part of the noise component captures the outlierness in the signal. The outlier detection process then consists of four phases: clustering, temporal outlier detection, removal of the spatial variation, and spatial outlier detection. In the beginning, sensors form a cluster and elect their cluster-head. After that, the cluster-head receives data from sensors, carries out most of the computation tasks, and broadcasts results to sensors. In the next phase, each sensor detects and recognizes any existing temporal outliers. This task can be realized by making all sensors operate simultaneously in parallel. In the third phase, it is important to remove any

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

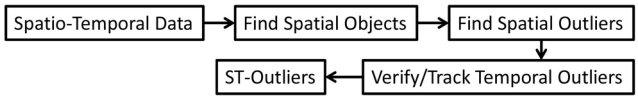

Fig. 6. ST-outlier detection framework.

geometrical effect caused by the sensor distribution. The signal power reduces in proportion to the distance from the source. If this bias is not accounted for properly, sensors near the source that receive more energy might be falsely regarded as outliers, even though they are not. If the data has some bias, the distribution might be heavily tailed with outliers located near the tail of the distribution. In that case, the min and max values are removed as outliers iteratively using the property of α-stable distributions. The last phase is the spatial outlier detection using the variogram method [119, p. 69-83]. Since this method generally shows the spatial variance between sensors, outliers can be considered as those ones that deviate remarkably from the majority of the data. Note that this method [120] assumes that the outliers are uncorrelated in time and space, and can be modeled as an α-stable distribution.

Yet another way to use the location of sensors is to consider neighbors within distance r. Given spatio-temporal data from a sensor network, a sensor s can be considered as an outlier for a time window w, if the value at sensor s at the end of the window has less than k measurements from all sensors within window w, within a distance r. Once outlier sensors are discovered, Franke *et al*. [121], [122] propose models to compute polygonal outlier regions in the sensor network.

5 Outlier Detection for Spatio-Temporal Data

Some studies find outliers considering only their temporal neighbors, while other work concerns outliers with respect to their spatial neighbors only. Combining these two notions, a spatio-temporal outlier (ST-Outlier) is a spatio-temporal object whose behavioral/thematic (non-spatial and non-temporal) attributes are significantly different from those of the other objects in its spatial and temporal neighborhoods. Fig. 6 shows a typical ST-Outlier detection pipeline.

5.1 Techniques for ST-Outlier Detection

Given: A spatio-temporal dataset
Find: ST-Outliers, i.e., spatio-temporal objects whose behavioral (non-spatial and non-temporal) attributes are significantly different from those of the other objects in its spatial and temporal neighborhoods

Most of the ST-Outlier detection techniques first find spatial outliers and then verify their temporal neighborhood. The techniques differ in the way spatial outliers are computed. We discuss these techniques below.

Birant *et al*. [123] propose a density-based ST-Outlier detection mechanism. First, they run a modified DBSCAN [124] clustering algorithm on the data with two main modifications: (1) To support the temporal aspect, a tree is traversed to find both spatial and temporal neighbors of any object within a given radius. (2) To find outliers, when clusters have different densities, the algorithm assigns a density factor to each cluster and compares the average value of a cluster with the new coming value. After clustering, potential outliers are detected. This is followed by checking the spatial neighbors to verify whether these objects are actually spatial outliers. The next step checks the temporal neighbors of the spatial outliers identified in the previous step. If the characteristic value of a spatial outlier does not have significant differences from its temporal neighbors, this is not an ST-Outlier. Otherwise, it is confirmed as an ST-Outlier.

Cheng *et al*. [125], [126] propose a four step approach. (1) *Clustering:* Form some regions that have significant semantic meanings, i.e. the spatio-temporal objects of interest. (2) *Aggregation:* Spatial resolution (scale) of the data is reduced for clustering. It is also called filtering since the outliers (noise) will be filtered after the spatial scale change. (3) *Comparison:* Results obtained at the two spatial scales are compared to detect the potential spatial outliers. The objects that are found in Step 1 and are missed (or filtered) in Step 2 are potential ST-Outliers. The comparison can be realized either by exploratory visualization analysis or visual data mining. (4) *Verification:* Check the temporal neighbors of the suspected ST-Outliers detected in the previous step to detect ST-Outliers.

Adam *et al*. [127] studied distance based outlier detection for a spatio-temporal dataset. Voronoi diagrams are used to establish spatial clusters called micro-clusters. Based on the spatial and semantic similarity (computed using the corresponding time series) between two micro-clusters they can be merged to get macro-clusters. Spatial similarity for micro-cluster computation uses two measures: the Jaccard coefficient that represents similarity and the Silhouette coefficient that represents quality of clustering. Any datum that differs sufficiently from other points in the macro neighborhood is then marked as an ST-Outlier.

5.2 Tracking of ST-Outliers

Given: A spatio-temporal dataset
Find: ST-Outlier solids, i.e., a region across time

While previous techniques consider an ST-Outlier as a point in space and time, an ST-Outlier could be considered as a solid with its base in the XY space dimension and volume across the time dimension. Wu *et al*. [128] propose an ST-Outlier detection algorithm called Outstretch, which discovers the outlier movement patterns of the top-K spatial outliers over several time periods. The top-K spatial outliers are found using the Exact-Grid Top-K and Approx-Grid Top-K algorithms (popular spatial outlier detection algorithms) with a well-known spatial scan statistic known as Kulldorff's scan statistic. An ST-Outlier may exist over more than one time period. For example, if there is higher than average precipitation in Peru over the years 1998-2002, then the solid in three dimensional (X, Y and time) space is an outlier. Using the top-K spatial outliers for each time period, they find all the sequences of outliers over time and store into a tree. The Outstretch algorithm takes as input the top-K values for each year period under analysis, and a variable r, the region stretch, which is the number of grids to 'stretch' by, on each side of an outlier. Outstretch then examines the top-K values of the second to last available

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

year periods. For all the years, each of the outliers from the current year are examined to see if they are framed by any of the stretched regions from the previous year. If they are, the item is added to the end of the previous year's child list. As a result, all possible sequences over all years get stored into the outlier tree and can be retrieved for analysis.

Similarly, Lu *et al*. [129] propose a wavelet fuzzy classification approach to detect and track region outliers in meteorological data. First, wavelet transform is applied to meteorological data to bring up distinct patterns that might be hidden within the original data. Then, a powerful image processing technique, edge detection with competitive fuzzy classifier, is extended to identify the boundary of region outlier. After that, to determine the center of the region outlier, the fuzzy-weighted average of the longitudes and latitudes of the boundary locations is computed. By linking the centers of the outlier regions within consecutive frames, the movement of a region outlier can be captured and traced.

5.3 Trajectory Outliers

Given: A set of trajectories
Find: Anomalous trajectories

Most of the techniques are based on distance, direction and density of trajectories. However, classification and historical similarity based techniques have also been proposed. We discuss these below.

Distance Between Trajectories: Lee *et al*. [130] propose TRAjectory Outlier Detection (TRAOD) algorithm which consists of two phases: (1) Partitioning Phase and (2) Detection Phase. In the first phase, each trajectory is split into partitions using a 2-level partitioning. Distance between trajectory partitions is calculated as a weighted sum of three components: perpendicular distance, parallel distance and angle distance. In the second phase, a trajectory partition is considered outlying if it does not have sufficient number of similar neighbors (with appropriate considerations of local density). A trajectory is an outlier if the sum of the length of its outlying partitions is at least F times the sum of lengths of all of its partitions.

Direction and Density of Trajectories: Ge *et al*. [131] consider two types of outlying trajectories: outliers in terms of direction and outliers in terms of density. The continuous space is discretized into small grids and a probabilistic model is used to turn the direction information of trajectories in a grid into a vector with eight values to indicate the probabilities of moving towards eight directions within this grid. The direction trend is thus generated within a fixed area by summarizing the moving directions from large amounts of trajectories for a period of time. Then, once some objects move across this area along the completely different directions from the summarized directions, they can be labeled as outliers in a real-time fashion by measuring the similarity between the directions of the observed objects and the summarized directions. Also, with this discretized space, the density of trajectories can be computed in each grid conveniently. The trajectory density within each grid is estimated as the number of trajectories across this grid. The trajectory density distribution can be obtained with sufficient historical trajectory data. The outlying score of a new trajectory can then be measured based on the density of trajectories in the grids where this trajectory actually passes.

Historical Similarity: While the above techniques stress on the spatial continuity to define outliers, Li *et al*. [132] propose a method for detecting temporal outliers with an emphasis on historical similarity trends between data points. At each time step, each road segment checks its similarity with the other road segments, and the historical similarity values are recorded in a temporal neighborhood vector at each road segment. Outliers are calculated from drastic changes in these vectors. Given a road segment with a historically stable set of neighbors (in feature space, not physical space), an outlier is loosely defined as a drastic change in the membership of this set. A change is noted heavily for a pair of road segments if the two road segments are historically similar but dissimilar at this time instant, or if the two road segments are not historically similar but similar at this time instant. Based on this, each road segment is given an exponential reward or penalty each day. The outlier score of an road segment on a particular day is then equal to the sum of rewards and penalties. The power of this method compared with a method that measures only the singular road segment is that it is robust to population shifts.

Trajectory Motifs: Li *et al*. [55] propose a motion-classifier for trajectory outlier detection, which consists of the following three steps. (a) Object movement features, called motifs, are extracted from the object paths. Each path consists of a sequence of motif expressions, associated with the values related to time and location. (b) To discover anomalies in object movements, motif based generalization is performed to cluster similar object movement fragments and generalize the movements based on the associated motifs. (c) With motif based generalization, objects are put into a multilevel feature space and are classified by a classifier that can handle high-dimensional feature spaces to discriminate anomalous trajectories from normal ones.

6 Outlier Detection for Temporal Networks

Given a stream of graphs, how does one identify an outlier graph? Given a temporal network with community distributions for each of its nodes, how does one track community evolution, and thereby define community outliers? In this section, we will study techniques that help us answer these questions.

6.1 Graph Similarity-Based Outlier Detection Algorithms

Given: A series of graph snapshots
Find: Outlier graph snapshots

Various graph distance metrics can be used to create time series of network changes by sequentially comparing graphs from adjacent periods. These time series are individually modeled as univariate autoregressive moving average (ARMA) processes and then outliers can be detected.

Let V be the set of nodes when both the graph snapshots G and H contain the same set of nodes, else V_G and V_H are used to denote the respective vertex sets of the two graphs.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

E_G and E_H are edges in graphs G and H. Let $w_G(u,v)$ and $w_H(u,v)$ denote the edge weights for edge (u,v) in the graphs G and H respectively. The distance/similarity measures to compute distances between two graphs G and H are mentioned below.

(1) Weight Distance [133]: It is a function of the difference between the edge weights of the graphs G and H normalized by the maximum of the edge weights in G or H.

(2) MCS Weight Distance [133]: It computes weight distance based on the edges in MCS where the maximum common subgraph (MCS) F of G and H is the common subgraph with the most vertices.

(3) MCS Edge Distance [133]: It is defined in terms of the number of edges in the MCS as follows. $d(G,H) = 1 - \frac{|mcs(E_G,E_H)|}{max(|E_G|,|E_H|)}$.

(4) MCS Vertex Distance [133]: It is defined in a similar way as in the MCS edge distance, but vertex sets of the two graphs rather than edge sets are used in the distance computation.

(5) Graph Edit Distance [133]–[135]: The simplest form of graph edit distance is given by $d(G,H) = |V_G| + |V_H| - 2|V_G \cap V_H| + |E_G| + |E_H| - 2|E_G \cap E_H|$. This captures the sequence of edit operations required to make graph G isomorphic to graph H. Graph edit distance can be expressed in other ways where the costs of inserting/deleting/substituting a node/edge are different [136]. Further, node edits and edge edits can be given different importance levels.

(6) Median Graph Edit Distance [133], [137]: Median graph can be computed for a fixed window of graphs using graph edit distance. Median graph edit distance is the edit distance of the graph from the median graph.

(7) Modality Distance [133]: The Modality distance between graphs G and H is the absolute value of the difference between the Perron vectors of these graphs. Perron vector is the eigen vector corresponding to the largest eigen value for the adjacency matrix of the corresponding graph.

(8) Diameter Distance [133], [138]: The Diameter distance between graphs G and H is the difference in the diameters for each graph.

(9) Entropy Distance [133], [138]: The Entropy distance between graphs G and H is defined using entropy-like measures associated with edge weights for the corresponding graphs.

(10) Spectral Distance [133], [138]: The Spectral distance between graphs G and H is defined in terms of the k largest eigenvalues of the Laplacian for the graphs G and H respectively.

(11) Umeyama graph distance [139]: The Umeyama graph distance is defined as the sum of the squares of the differences between edge weights for the corresponding edges of the graphs G and H respectively.

(12) Vector Similarity [134]: This is computed as the Euclidean distance between the principal eigenvectors of the graph adjacency matrices of the two graphs.

(13) Spearman's correlation coefficient [134]: This is calculated as the rank correlation between sorted lists of vertices of the two graphs. Vertices are sorted based on the PageRank or some other properties.

(14) Sequence similarity [134], [140]: It is the similarity of vertex sequences of the graphs that are obtained through a graph serialization algorithm. Such an algorithm creates a serial ordering of graph vertices which is maximally edge connected. That means that it maximizes the number of vertices that are consecutive in the ordering and are edge connected in the graph.

(15) Signature similarity [134], [140]: It is computed as the Hamming distance between the fingerprints of two graphs. To get such fingerprints, a similarity hash function is applied to the graphs being compared.

(16) Vertex/edge overlap (VEO) [140]: Two graphs are similar if they share many vertices and/or edges. The Jaccard index is one way to compute the overlap between two graphs. It is defined as the length of intersections of vertices/edges divided by that of the union.

(17) Vertex ranking (VR) [140]: Two graphs are similar if the rankings of their vertices are similar. The vertices can be ranked using their qualities, and the similarity of rankings can be computed using a rank correlation method such as Spearman's correlation coefficient.

For TCP/IP traffic data, it was observed that time series based on the MCS edge, MCS vertex, edit, median and entropy metrics had no false positives. However, the time series based on the spectral, modality and diameter distance metrics are considerably more computationally intensive and also did not lend themselves to accurate ARMA modelling [133]. Amongst vertex ranking, vector similarity, vertex/edge overlap, sequence similarity, and signature similarity, Papadimitriou *et al*. [134] observed that signature similarity and vector similarity were the best at detecting anomalies from web crawl graphs while not yielding false positives.

Another measure to define outliers on a pair of graph snapshots is the shortest path distance between any pair of nodes. Given two snapshots of a temporal graph, Gupta *et al*. [141] study the problem of finding node pair outliers which are node pairs with maximum change in shortest path distance across the two graph snapshots.

6.2 Online Graph Outlier Detection Algorithms

Given: A stream of graphs

Find: Outlier graph snapshots or outlier localized regions

The techniques in the previous subsection are usually applied over a fixed length time series of graphs. Next, we will discuss an eigenvector based approach and a structural outlier detection approach for graph streams.

Ide *et al*. [142] proposed an eigenvector-based method for anomaly detection over graph streams. The principal eigenvector of the graph weight matrix at time t is called activity vector $u(t)$. Activity vectors over time interval h are stored in the matrix $U(t) = [u(t), u(t-1), \ldots, u(t-h+1)]$. Then the "typical pattern" is defined to be the left singular vector of $U(t)$. For any new data point at time t, this $U(t-1)$ matrix is constructed, the typical pattern is extracted and then the typical pattern vector is compared against the activity vector at time t. The angle between the vectors gives similarity between the activity vector and the typical pattern of the data in the last h time snapshots. The authors provide an online algorithm to calculate the threshold for the

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

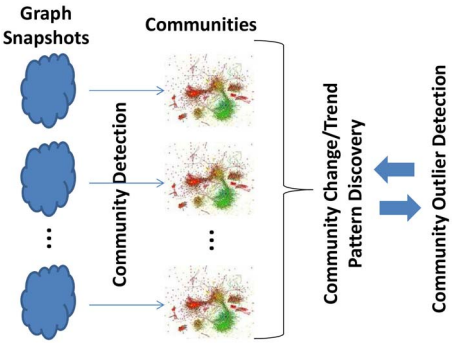

Fig. 7. Framework for community outlier detection.

angle as well. A similar method is also proposed by Akoglu *et al*. [143] to spot anomalous points in time at which many agents in an agent network change their behavior in a way such that it deviates from the norm.

While Ide *et al*. take a spectral approach, Aggarwal *et al*. [9] propose the problem of structural outlier detection in massive network streams. They use a structural connectivity model to define outliers in graph streams as those graph objects which contain such unusual bridging edges. To handle the sparsity problem of massive networks, the network is dynamically partitioned to construct statistically robust models of the connectivity behavior. These models are maintained using an innovative reservoir sampling approach for efficient structural compression of the underlying graph stream. Using these models, edge generation probability is defined and then graph object likelihood fit is defined as the geometric mean of the likelihood fits of its constituent edges. Those objects for which this fit is t standard deviations below the average of the likelihood probabilities of all objects received so far are reported as outliers. Further, in [162], the authors maintain the information about the changes in the different neighborhoods of the network using a localized principal component analysis algorithm. Localized regions of sudden change in the underlying network are then reported as outliers.

6.3 Community-Based Outlier Detection Algorithms

Given: A series of graph snapshots
Find: Outlier nodes with anomalous temporal community changes

Given two snapshots of a network, the communities evolve across the two snapshots. While most objects follow the popular community distribution change trends, some objects do not. Such rare objects are called Evolutionary Community Outliers (ECOutliers) [13]. Fig. 7 shows a typical framework to discover community outliers from network data. To begin with, communities can be discovered independently in both snapshots. To discover ECOutliers, it is important to first identify the usual transition trends of community evolution. A key problem in discovering the community transition trends is computing matching communities across both snapshots. ECOutlier detection can then be performed using these patterns. However, community matching itself suffers from ECOutliers and hence must be done in an outlier-aware way. The framework discussed in [13] integrates outlier detection and community matching together in an iterative manner. Such an integrated formulation performs much better than performing outlier detection after community matching separately.

Moreover, the authors investigated the task of outlier detection in a general setting of multiple network snapshots [12]. Such outliers are called Community Trend Outliers (CTOutliers). This study exploits the subtle temporal community evolution trends to perform outlier detection. While ECOutliers [13] capture the notion of anomalous community transitions, CTOutliers [12] capture anomalous community changes across multiple time snapshots, and in a more subtle way. Two main challenges were (1) computing patterns from a sequence of community membership probability distributions (soft sequences) associated with every object for a series of timestamps, and (2) computing the outlier score of individual soft sequences based on such patterns. The authors proposed an effective two-step procedure to detect community trend outliers. They first modeled the normal evolutionary behavior of communities across time using soft patterns discovered from the dataset. In the second step, they proposed effective measures to evaluate probability of an object deviating from the normal evolutionary patterns.

Besides these, a variety of metrics have been proposed in [144], that can be used for community based temporal outlier detection in graphs. These include consistency between two clusterings, snapshot clustering quality, social stability and sociability of objects. Though community based outlier detection has been studied for a static heterogeneous graph recently [145], there is no technique yet for temporal heterogeneous graphs.

7 Applications of Temporal Outlier Detection Techniques

In this section, we briefly discuss different environmental, industrial, surveillance, computer network, biological, astronomy, web, information network and economics applications, which are relevant to temporal outlier detection.

7.1 Environmental Sensor Data

Hill *et al*. [75], [106] use dynamic Bayesian networks to identify measurement errors in a wind speed data stream from WATERS Network Corpus Christi Bay testbed. Angiulli *et al*. [108] extract distance outliers from the rain, sea surface temperature, relative humidity, precipitation time series obtained from the Pacific Marine Environmental Laboratory of the U.S. National Oceanic and Atmospheric Administration (NOAA). Birant *et al*. [123] identify spatio-temporal outliers from the wave height values of four seas: the Black Sea, the Marmara Sea, the Aegean Sea, and the east of the Mediterranean Sea. The outliers consist of locations with significantly high wave height values on a particular date compared to its spatio-temporal neighbors. Cheng *et al*. [126] compute spatio-temporal outliers from the water height data obtained from Ameland, a barrier island in the north of the Netherlands. Sun *et al*. [146] explore the South China area dataset from 1992 to 2002 with five variables: cloud cover percentage, diurnal temperature range, precipitation, temperature, vapor and pressure. They answer outlier queries like "which locations always

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

have different temperature from their neighborhoods in the recent 10 years". They can find droughts and flood outliers like flood in Yangzi River Valley in 1998. Wu *et al*. [128] find extreme precipitation events from South American precipitation data set obtained from the NOAA (Hydrology) for Peru over 1998-2002. Outlier detection using rotated PCA has also been used to discover drought areas, intense fertility loss areas, hurricanes from Australian district rainfall [147], Sicily Island yearly Maximum Value Composit of SPOT/VEGETATION NDVI [148] and NOAA/NCEP (National Centers of Environmental Prediction) global reanalysis [129] data sets.

7.2 Industrial Sensor Data

The work in [74] predicts anomalies based on two different signals obtained from a flight data recorder (FDR). These correspond to the altitude of the aircraft and roll angle. These have been used to find anomalous altitude changes and roll angle using deviation from median-based prediction. Anomalies in jet engines have been discovered by analyzing the high/low pressure shaft harmonic frequencies in jet engine vibration data [25] using k-Means clustering. Similarly, anomalies such as tool breakage detection have been discovered using cutting force data [149] with a neural network. Bu *et al*. [93] discover discords from power consumption data for a Dutch research facility using Haar wavelets and augmented tries.

7.3 Surveillance and Trajectory Data

Li *et al*. [132] discover anomalies from average daily speed and average daily load dataset for taxicabs in San Francisco during July 2006. Ge *et al*. [131] find anomalies such as vehicle moving to the left side of the road (wrong way), vehicle taking a wrong short cut, and people jaywalking. Yankov *et al*. [97] also find discords from trajectory data. Trajectory data can also be used for monitoring the movement of patients and senior citizens. For example, it can be used to discover events such as taking a wrong bus, having a bad fall, encountering a sudden slow-down and getting lost [110]. They use dynamic cluster maintenance to discover such outlier activities.

Surveillance data can be useful in smart homes to discover anomaly situations. An example would be the situation where the resident has turned on the bathwater, but has not turned it off before going to bed [150]. Similarly, data from health and medical monitoring sensors attached to patients can be used to identify symptoms or diseases. Surveillance videos can be explored to discover outliers, such as the appearance of a new object, object zooming in a camera and novel video content [113] using the distance based local outliers for data streams. Pokrajac *et al*. [113] study trajectories from surveillance videos to identify anomalous behaviors such as people first walking right and then back left, or people walking very slowly.

7.4 Computer Networks Data

Techniques for outlier detection from temporal data have been widely used for intrusion detection [24], [61]–[63], [108], [151]. Lakhina *et al*. [152] use multivariate timeseries of origin-destination flows measuring the number of bytes, packets, and IP-level flows to discover anomalies such as high rate point-to-point byte transfer, denial of service (DOS), distributed denial of service (DDOS) attacks, flash crowd (large demand for a resource/service), host scanning for a vulnerable port or network scanning for a target port, WORM[5], OUTAGE[6], etc. They find outliers using the subspace method which is borrowed from the multivariate statistical process control literature. Domain-specific features, such as duration, protocol type, number of bytes transferred for TCP connection with Euclidean distance are used with a variant of single-linkage clustering in [28]. Ye [38] uses Markov models on the audit data of a Sun Solaris system from MIT Lincoln lab to detect intrusion scenarios, such as password guessing, root privileges access through symbolic links, remote unauthorized access attempt, etc. Sequences of Unix user commands are examined in [33] using SVMs to discriminate between normal sequences and intrusion sequences. A variant of the online sequential discounting algorithm has been used in [153] to characterize probabilistic correlation between flow-intensities measured at multiple points in a computer network using GMMs. An AutoRegressive model with eXogenous inputs (ARX) has been proposed in [154] to learn the relationship between flow intensities measured at multiple points in a computer network, and thereby detect correlation-based faults.

7.5 Biological Data

Keogh *et al*. [91] discover anomalous subsequences (discords) from electrocardiograms. Shape discords (unusual shapes) have been found to be useful in zoology, anthropology, and medicine [92]. Wei *et al*. [92] use shape discords to discover an odd butterfly, given a group of seemingly similar butterflies. Given a dataset of red blood cells, they use their techniques to discover teardrop shaped cells, or dacrocytes, which is indicative of several blood disorders. A fungus dataset has been studied to discover spores that have sprouted an "appendage" known as a germ tube. Endangered species equipped with sensors have been studied to discover abnormal migration behaviors of the species.

7.6 Astronomy Data

Keogh *et al*. [91] discover discords from Space telemetry data. These discords correspond to manually noted errors, such as "Poppet pulled significantly out of the solenoid before energizing". Yankov *et al*. [97] find discords from star light-curve data. Zhu *et al*. [101] detect high gamma ray bursts from Milagro Gamma Ray data stream.

7.7 Web Data

Given multiple crawls of the web graph, Papadimitriou *et al*. [134], [140] find a crawl graph with anomalies. These anomalies refer either to failures of the web hosts that do not allow the crawler to access their content or

5. A self propagating code that spreads across a network by exploiting security flaws.

6. Events causing decrease in traffic exchanged between a pair of nodes.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

hardware/software problems in the search engine infrastructure that can corrupt parts of the crawled data. They use graph similarity measures like vertex/edge overlap, vertex ranking, vertex/edge vector similarity, sequence similarity and signature similarity to derive respective time series and then detect outliers. Yankov *et al*. [97] study the MSN query logs to discover both anticipated bursts (e.g., "easter", "hanukkah") and unanticipated bursts (e.g., unexpected events like death of a celebrity) of web queries as discords. Lappas *et al*. [155] use the notion of spatiotemporal term burstiness to compute highly spatiotemporally impactful events as outliers. Mathioudakis *et al*. [156] also propose efficient dynamic programming algorithms for the similar task of identifying notable geographically focused events from user generated content.

7.8 Information Network Data

Aggarwal *et al*. [9] discover graphs representing interdisciplinary research papers as outliers from the DBLP dataset using structural outlier detection in massive network streams. They also discover movies with a cast from multiple countries as outliers from the IMDB dataset. Gupta *et al*. [13] discover those authors from DBLP coauthorship network as outliers which show a significant deviation in the changes of their research areas compared with other authors. This work explores the difference across two network snapshots using temporal community outlier detection methods. Similarly, they discover actors from IMDB as outliers which show a very unusual change in their movie genre distributions. Priebe *et al*. [157] study the communication graph of the Enron data with respect to the maximum degree and digraph size scan statistics. Excessive communication outliers which can suggest insider trading scenarios are discovered.

7.9 Economics Time Series Data

Various temporal economic datasets have been studied with respect to outlier detection. Gupta *et al*. [12] identify country outliers based on the unusual change in the constituents of the GDP (Consumption, Investment, Public Expenditure and Net Exports) across time, using temporal community outlier detection methods. They also find U.S. states as outliers from the Budget dataset with respect to anomalous change in the distribution of spending with respect to different components. Otey *et al*. [116] study the U.S. Census Bureau's Income data set to detect outliers, on the basis of unusual combinations of demographic attribute values. They also studied the U.S. Congressional Voting Data to find outliers. An example is a Republican congressman who voted significantly differently from his party on four bills. They use distance based outlier detection for distributed temporal data to obtain such outliers. Zhu *et al*. [101] perform outlier subsequence detection from the NYSE IBM stock time series data using Shifted Wavelet Trees.

8 Conclusion and Lessons Learned

In this work, we presented an organized overview of the various techniques proposed for outlier detection on temporal data. Modeling temporal data is a challenging task due to the dynamic nature and complex evolutionary patterns in the data. In the past, there are a wide variety of models developed to capture different facets in temporal data outlier detection. This survey organized the discussion along different data types, presented various outlier definitions, and briefly introduced the corresponding techniques. Finally, we also discussed various applications for which these techniques have been successfully used. This survey provides a number of insights and lessons as follows.

- The methods for different data types are not easy to generalize to one another, though some of them may have similarity in the framework at the broader level. For example, change detection in continuous time series and discrete time series both require forecasting methods. However, the specific kind of forecasting method is extremely different in the two scenarios (regression models in one vs Markov Models in the other).
- Most window based models are currently offline, whereas online methods do exist for point based models. Therefore, there is significant opportunity for research in the former.
- While the number of formulations of the temporal outlier detection problem are diverse, they are generally motivated by the most common applications which are encountered in the literature. Many recent applications, especially those corresponding to novel data types in the context of web-based applications, may result in combinations of facets, which have not been explored before.
- There are numerous formulations of temporal outlier detection, which have not been sufficiently explored. This is because of the many different combinations of facets, which can be used for defining temporal outlier detection problems. Complex data types such as social streams in which two different data types are present in combination (text and structure) have also not been studied.

This comprehensive survey provides a good insight into various techniques that have been applied on multiple forms of temporal data and can be used to identify problem settings that can be worked on in the future. Finally, for further reading, we direct the reader to a recent book on outlier analysis [1] and [158] for a tutorial version of this survey. Also, [159] present outlier detection techniques for time series data and [160], [161] discuss novelty detection techniques.

Acknowledgments

This work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement W911NF-11-2-0086 (Cyber-Security) and Cooperative Agreement W911NF-09-2-0053 (NS-CTA), in part by the U.S. Army Research Office under Cooperative Agreement W911NF-13-1-0193, and in part by U.S. National Science Foundation grant CNS-0931975, grant IIS-1017362, and grant IIS-1320617.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] C. C. Aggarwal, *Outlier Analysis*. New York, NY, USA: Springer, 2013.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[3] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.

[4] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," Centre Telemat. Inform. Technol. Univ. Twente, Enschede, The Netherlands, Tech. Rep. TR-CTIT-08-59, Oct. 2008.

[5] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *SIGMOD Rec.*, vol. 30, pp. 37–46, May 2001.

[6] C. C. Aggarwal and P. S. Yu, "Outlier detection with uncertain data," in *Proc. 2008 SIAM Int. Conf. SDM*, pp. 483–493.

[7] C. Aggarwal and K. Subbian, "Event detection in social streams," in *Proc. 12th SIAM Int. Conf. SDM*, 2012, pp. 624–635.

[8] C. C. Aggarwal, "On abnormality detection in spuriously populated data streams," in *Proc. 2005 SIAM Int. Conf. SDM*, pp. 80–91.

[9] C. C. Aggarwal, Y. Zhao, and P. S. Yu, "Outlier detection in graph streams," in *Proc. 27th ICDE*, Hannover, Germany, 2011, pp. 399–409.

[10] J. Gao *et al.*, "On community outliers and their efficient detection in information networks," in *Proc. 16th ACM Int. Conf. KDD*, 2010, pp. 813–822.

[11] A. Ghoting, M. E. Otey, and S. Parthasarathy, "LOADED: Link-based outlier and anomaly detection in evolving data sets," in *Proc. 4th IEEE ICDM*, 2004, pp. 387–390.

[12] M. Gupta, J. Gao, Y. Sun, and J. Han, "Community trend outlier detection using soft temporal pattern mining," in *Proc. ECML PKDD*, Bristol, U.K., 2012, pp. 692–708.

[13] M. Gupta, J. Gao, Y. Sun, and J. Han, "Integrating community matching and outlier detection for mining evolutionary community outliers," in *Proc. 18th ACM Int. Conf. KDD*, Beijing, China, 2012, pp. 859–867.

[14] J. P. Burman and M. C. Otto, "Census bureau research project: Outliers in time series," Bureau of the Census, SRD Res. Rep. CENSUS/SRD/RR-88114, May 1988.

[15] A. J. Fox, "Outliers in time series," *J. Roy. Statist. Soc. B Methodol.*, vol. 34, no. 3, pp. 350–363, 1972.

[16] H. Cho, Y. jin Kim, H. J. Jung, S.-W. Lee, and J. W. Lee, "OutlierD: An R package for outlier detection using quantile regression on mass spectrometry data," *Bioinformatics*, vol. 24, no. 6, pp. 882–884, 2008.

[17] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York, NY, USA: Wiley, 1978.

[18] D. M. Hawkins, *Identification of Outliers*. London, U.K.: Chapman and Hall, 1980.

[19] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York, NY, USA: Wiley, 1987.

[20] T. Lane *et al.*, "Sequence matching and learning in anomaly detection for computer security," in *Proc. AAAI Workshop AI Approaches Fraud Detection Risk Manage.*, 1997, pp. 43–49.

[21] S. Budalakoti, A. Srivastava, R. Akella, and E. Turkov, "Anomaly detection in large sets of high-dimensional symbol sequences," NASA Ames Res. Center, Mountain View, CA, USA, Tech. Rep. NASA TM-2006-214553, 2006.

[22] S. Budalakoti, A. N. Srivastava, and M. E. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 39, no. 1, pp. 101–113, Jan. 2009.

[23] V. Chandola, V. Mithal, and V. Kumar, "A comparative evaluation of anomaly detection techniques for sequence data," in *Proc. 2008 8th IEEE ICDM*, Pisa, Italy, pp. 743–748.

[24] K. Sequeira and M. Zaki, "ADMIT: Anomaly-based data mining for intrusions," in *Proc. 8th ACM Int. Conf. KDD*, New York, NY, USA, 2002, pp. 386–395.

[25] A. Nairac *et al.*, "A system for the analysis of jet engine vibration data," *Integr. Comput. Aided Eng.*, vol. 6, no. 1, pp. 53–66, Jan. 1999.

[26] X. Pan, J. Tan, S. Kavulya, R. Gandhi, and P. Narasimhan, "Ganesha: BlackBox diagnosis of mapReduce systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 3, pp. 8–13, Jan. 2010.

[27] U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock, "Finding anomalous periodic time series," *J. Mach. Learn.*, vol. 74, no. 3, pp. 281–313, Mar. 2009.

[28] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proc. ACM CSS Workshop DMSA*, 2001, pp. 5–8.

[29] M. Gupta, A. B. Sharma, H. Chen, and G. Jiang, "Context-aware time series anomaly detection for complex systems," in *Proc. SDM Workshop*, 2013.

[30] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data," in *Proc. Appl. Data Mining Comput. Security*, 2002.

[31] P. Evangelista, P. Bonnisone, M. Embrechts, and B. Szymanski, "Fuzzy ROC curves for the 1 class SVM: Application to intrusion detection," in *Proc. 13th Eur. Symp. Artif. Neural Netw.*, 2005, pp. 345–350.

[32] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proc. IJCNN*, Jul. 2003, pp. 1741–1745.

[33] B. Szymanski and Y. Zhang, "Recursive data mining for masquerade detection and author identification," in *Proc. 5th Annu. IEEE SMC Inform. Assur. Workshop*, 2004, pp. 424–431.

[34] F. A. González and D. Dasgupta, "Anomaly detection using real-valued negative selection," *Genet. Program. Evolvable Mach.*, vol. 4, no. 4, pp. 383–403, Dec. 2003.

[35] C. Marceau, "Characterizing the behavior of a program using multiple-length n-grams," in *Proc. 2000 NSPW*, pp. 101–110.

[36] C. C. Michael and A. Ghosh, "Two state-based approaches to program-based anomaly detection," in *Proc. 16th ACSAC*, New Orleans, LA, USA, 2000, pp. 21–30.

[37] S. Salvador and P. Chan, "Learning states and rules for detecting anomalies in time series," *Appl. Intell.*, vol. 23, no. 3, pp. 241–255, Dec. 2005.

[38] N. Ye, "A Markov chain model of temporal behavior for anomaly detection," in *Proc. 2000 IEEE SMC Inform. Assur. Security Workshop*, vol. 166. pp. 171–174.

[39] J. Yang and W. Wang, "CLUSEQ: Efficient and effective sequence clustering," in *Proc. 19th ICDE*, 2003, pp. 101–112.

[40] P. Sun, S. Chawla, and B. Arunasalam, "Mining for outliers in sequential databases," in *Proc. 6th SIAM Int. Conf. SDM*, 2006, pp. 94–105.

[41] E. Eskin, W. Lee, and S. Stolfo, "Modeling system calls for intrusion detection with dynamic window sizes," in *Proc. DISCEX*, vol. 1. 2001, pp. 165–175.

[42] W. Lee, S. J. Stolfo, and P. K. Chan, "Learning patterns from unix process execution traces for intrusion detection," in *Proc. AAAI Workshop AI Approaches Fraud Detection Risk Manage.*, 1997, pp. 50–56.

[43] G. Florez-Larrahondo, S. M. Bridges, and R. Vaughn, "Efficient modeling of discrete events for anomaly detection using hidden Markov models," in *Proc. 8th Int. Conf. ISC*, 2005, pp. 506–514.

[44] B. Gao, H.-Y. Ma, and Y.-H. Yang, "HMMs (Hidden Markov Models) based on anomaly intrusion detection method," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2002, pp. 381–385.

[45] Y. Qiao, X. Xin, Y. Bin, and S. Ge, "Anomaly intrusion detection method based on HMM," *Electron. Lett.*, vol. 38, no. 13, pp. 663–664, Jun. 2002.

[46] X. Zhang, P. Fan, and Z. Zhu, "A new anomaly detection method based on hierarchical HMM," in *Proc. 4th Int. Conf. PDCAT*, Aug. 2003, pp. 249–252.

[47] X. Li and J. Han, "Mining approximate top-k subspace anomalies in multi-dimensional time-series data," in *Proc. 33rd Int. Conf. VLDB*, Vienna, Austria, 2007, pp. 447–458.

[48] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Proc. 7th Conf. USENIX SSYM*, Berkeley, CA, USA, 1998, pp. 6–20.

[49] J. B. D. Cabrera, L. Lewis, and R. K. Mehra, "Detection and classification of intrusions and faults using sequences of system calls," *SIGMOD Rec.*, vol. 30, no. 4, pp. 25–34, Dec. 2001.

[50] D. Dasgupta and F. Nino, "A comparison of negative and positive selection algorithms in novel pattern detection," in *Proc. IEEE Int. Conf. SMC*, Nashville, TN, USA, 2000, pp. 125–130.

[51] D. Endler, "Intrusion detection applying machine learning to solaris audit data," in *Proc. 14th ACSAC*, Phoenix, AZ, USA, 1998, pp. 268–279.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

[52] A. K. Gosh, J. Wanken, and F. Charron, "Detecting anomalous and unknown intrusions against programs," in *Proc. 14th ACSAC*, Phoenix, AZ, USA, 1998, pp. 259–267.
[53] A. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection," in *Proc. 1st USENIX Workshop Intrusion Detection Network Monitoring*, Berkeley, CA, USA, 1999, pp. 51–62.
[54] A. K. Ghosh and A. Schwartzbard, "A study in using neural networks for anomaly and misuse detection," in *Proc. 8th Conf. USENIX SSYM*, Berkeley, CA, USA, 1999, pp. 12–23.
[55] X. Li, J. Han, and S. Kim, "Motion-alert: Automatic anomaly detection in massive moving objects," in *Proc. 4th IEEE Int. Conf. ISI*, San Diego, CA, USA, 2006, pp. 166–177.
[56] D.-K. Kang, D. Fuller, and V. Honavar, "Learning classifiers for misuse detection using a bag of system calls representation," in *Proc. 3rd IEEE Int. Conf. ISI*, Atlanta, GA, USA, 2005, pp. 511–516.
[57] S. Tian, S. Mu, and C. Yin, "Sequence-similarity kernels for SVMs to detect anomalies in system calls," *Neurocomputing*, vol. 70, no. 4–6, pp. 859–866, Jan. 2007.
[58] M. Wang, C. Zhang, and J. Yu, "Native API based windows anomaly intrusion detection method using SVM," in *Proc. IEEE Int. Conf. SUTC*, vol 1. 2006, pp. 514–519.
[59] X. Li, J. Han, S. Kim, and H. Gonzalez, "ROAM: Rule- and motif-based anomaly detection in massive moving object data sets," in *Proc. 7th SIAM Int. Conf. SDM*, 2007, pp. 273–284.
[60] A. Ghosh *et al.*, "Using program behavior profiles for intrusion detection," in *Proc. SANS Intrusion Detection Workshop*, 1999.
[61] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *J. Comput. Security*, vol. 6, no. 3, pp. 151–180, Aug. 1998.
[62] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proc. 20th NISSC*, 1997, pp. 366–380.
[63] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," in *Proc. 5th ACM Conf. CCS*, 1998, pp. 150–158.
[64] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, USA, 1996, pp. 120–128.
[65] D. Dasgupta and N. Majumdar, "Anomaly detection in multidimensional data using negative selection algorithm," in *Proc. Congr. CEC*, Honolulu, HI, USA, 2002, pp. 1039–1044.
[66] P. D'haeseleer, S. Forrest, and P. Helman, "An immunological approach to change detection: Algorithms, analysis and implications," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, USA, May 1996, pp. 110–119.
[67] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, USA, 1994, pp. 202–212.
[68] E. Keogh, S. Lonardi, and B. Y.-C. Chiu, "Finding surprising patterns in a time series database in linear time and space," in *Proc. 8th ACM Int. Conf. KDD*, Edmonton, AB, Canada, 2002, pp. 550–556.
[69] M. Atallah, R. Gwadera, and W. Szpankowski, "Detection of significant sets of episodes in event sequences," in *Proc. 4th IEEE ICDM*, 2004, pp. 3–10.
[70] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Research Issues DMKD*, San Diego, CA, USA, 2003, pp. 2–11.
[71] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: A novel symbolic representation of time series," *Data Mining Knowl. Discov.*, vol. 15, no. 2, pp. 107–144, Oct. 2007.
[72] R. Gwadera, M. J. Atallah, and W. Szpankowski, "Markov models for identification of significant episodes," in *Proc. 5th SIAM Int. Conf. SDM*, 2005, pp. 404–414.
[73] R. Gwadera, M. J. Atallah, and W. Szpankowski, "Reliable detection of episodes in event sequences," *Knowl. Inform. Syst.*, vol. 7, no. 4, pp. 415–437, May 2005.
[74] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: An application to sensor data," *Knowl. Inform. Syst.*, vol. 11, no. 2, pp. 137–154, Feb. 2007.
[75] D. J. Hill and B. S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach," *Environ. Modell. Softw.*, vol. 25, no. 9, pp. 1014–1022, Sep. 2010.
[76] J. Ma and S. Perkins, "Online novelty detection on temporal sequences," in *Proc. 9th ACM Int. Conf KDD*, New York, NY, USA, 2003, pp. 613–618.
[77] N. D. Le, R. D. Martin, and A. E. Raftery, "Modeling flat stretches, bursts, and outliers in time series using mixture transition distribution models," *J. Amer. Statist. Assoc.*, vol. 91, no. 436, pp. 1504–1515, 1996.
[78] R. S. Tsay, D. Pena, and A. E. Pankratz, "Outliers in multivariate time series," *Biometrika*, vol. 87, no. 4, pp. 789–804, 2000.
[79] A. Luceno, "Detecting possibly non-consecutive outliers in industrial time series," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 60, no. 2, pp. 295–310, 1998.
[80] A. Justel, D. Pena, and R. S. Tsay, "Detection of outlier patches in autoregressive time series," *Statist. Sinica*, vol. 11, no. 3, pp. 651–674, 2001.
[81] A. M. Bianco, M. García Ben, E. J. Martínez, and V. J. Yohai, "Outlier detection in regression models with ARIMA errors using robust estimates," *J. Forecasting*, vol. 20, no. 8, pp. 565–579, Dec. 2001.
[82] I. Chang, G. C. Tiao, and C. Chen, "Estimation of time series parameters in the presence of outliers," *Technometrics*, vol. 30, no. 2, pp. 193–204, 1988.
[83] C. Chen and L.-M. Liu, "Joint estimation of model parameters and outlier effects in time series," *J. Amer. Statist. Assoc.*, vol. 88, no. 421, pp. 284–297, 1993.
[84] R. S. Tsay, "Time series model specification in the presence of outliers," *J. Amer. Statist. Assoc.*, vol. 81, no. 393, pp. 132–141, 1986.
[85] P. Galeano, D. Pena, and R. S. Tsay, "Outlier detection in multivariate time series by projection pursuit," *J. Amer. Statist. Assoc.*, vol. 101, no. 474, pp. 654–669, 2006.
[86] A. W. Williams, S. M. Pertet, and P. Narasimhan, "Tiresias: Black-box failure prediction in distributed systems," in *Proc. 21st IPDPS*, Long Beach, CA, USA, 2007, pp. 1–8.
[87] G. Silvestri, F. Verona, M. Innocenti, and M. Napolitano, "Fault detection using neural networks," in *Proc. IEEE Int. Conf. Neural Netw.*, Orlando, FL, USA, Jun. 1994, pp. 3796–3799.
[88] H. V. Jagadish, N. Koudas, and S. Muthukrishnan, "Mining deviants in a time series database," in *Proc. 25th Int. Conf. VLDB*, Edinburgh, U.K., 1999, pp. 102–113.
[89] S. Muthukrishnan, R. Shah, and J. Vitter, "Mining deviants in time series data streams," in *Proc. 16th Int. Conf. SSDBM*, Jun. 2004, pp. 41–50.
[90] E. Keogh, J. Lin, S.-H. Lee, and H. Van Herle, "Finding the most unusual time series subsequence: Algorithms and applications," *Knowl. Inform. Syst.*, vol. 11, no. 1, pp. 1–27, Dec. 2006.
[91] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently finding the most unusual time series subsequence," in *Proc. 5th IEEE ICDM*, 2005, pp. 226–233.
[92] L. Wei, E. Keogh, and X. Xi, "SAXually explicit images: Finding unusual shapes," in *Proc. 6th ICDM*, Hong Kong, 2006, pp. 711–720.
[93] Y. Bu *et al.*, "WAT: Finding top-k discords in time series database," in *Proc. 7th SIAM Int. Conf. SDM*, 2007, pp. 449–454.
[94] A. W.-C. Fu, O. T.-W. Leung, E. Keogh, and J. Lin, "Finding time series discords based on haar transform," in *Proc. 2nd Int. Conf. ADMA*, Xi'an, China, 2006, pp. 31–41.
[95] J. Lin, E. Keogh, A. Fu, and H. Van Herle, "Approximations to magic: Finding unusual medical time series," in *Proc. 18th IEEE Symp. CBMS*, 2005, pp. 329–334.
[96] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *Proc. 10th ACM Int. Conf. KDD*, Seattle, WA, USA, 2004, pp. 206–215.
[97] D. Yankov, E. Keogh, and U. Rebbapragada, "Disk aware discord discovery: Finding unusual time series in terabyte sized datasets," *Knowl. Inform. Syst.*, vol. 17, no. 2, pp. 241–262, Nov. 2008.
[98] X.-Y. Chen and Y.-Y. Zhan, "Multi-scale anomaly detection algorithm based on infrequent pattern of time series," *J. Comput. Appl. Math.*, vol. 214, no. 1, pp. 227–237, Apr. 2008.
[99] C. Shahabi, X. Tian, and W. Zhao, "TSA-tree: A wavelet-based approach to improve the efficiency of multi-level surprise and trend queries on time-series data," in *Proc. 12th Int. Conf. SSDBM*, Berlin, Germany, 2000, pp. 55–68.
[100] L. Wei *et al.*, "Assumption-free anomaly detection in time series," in *Proc. 17th Int. Conf. SSDBM*, Berkeley, CA, USA, 2005, pp. 237–240.
[101] Y. Zhu and D. Shasha, "Efficient elastic burst detection in data streams," in *Proc. 9th ACM Int. Conf KDD*, New York, NY, USA, 2003, pp. 336–345.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

[102] K. Yamanishi and J.-I. Takeuchi, "A unifying framework for detecting outliers and change points from non-stationary time series data," in *Proc. 8th ACM Int. Conf. KDD*, Edmonton, AB, Canada, 2002, pp. 676–681.

[103] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, "Online unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining Knowl. Discov.*, vol. 8, no. 3, pp. 275–300, May 2004.

[104] A. V. Harold and S. Javitz, "The NIDES statistical component description and justification," SRI Int., Menlo Park, CA, USA, Tech. Rep. A010, Mar. 1994.

[105] C. C. Aggarwal and P. S. Yu, "On clustering massive text and categorical data streams," *Knowl. Inform. Syst.*, vol. 24, no. 2, pp. 171–196, 2010.

[106] D. J. Hill, B. S. Minsker, and E. Amir, "Real-time Bayesian anomaly detection for environmental sensor data," in *Proc. 32nd Conf. IAHR*, 2007.

[107] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proc. 24th Int. Conf. VLDB*, New York, NY, USA, 1998, pp. 392–403.

[108] F. Angiulli and F. Fassetti, "Detecting distance-based outliers in streams of data," in *Proc. 16th ACM CIKM*, Lisboa, Portugal, 2007, pp. 811–820.

[109] D. Yang, E. A. Rundensteiner, and M. O. Ward, "Neighbor-based pattern detection for windows over streaming data," in *Proc. 12th Int. Conf. EDBT*, 2009, pp. 529–540.

[110] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, "Efficient anomaly monitoring over moving object trajectory streams," in *Proc. 15th ACM Int. Conf. KDD*, Paris, France, 2009, pp. 159–168.

[111] H. Cao, Y. Zhou, L. Shou, and G. Chen, "Attribute outlier detection over data streams," in *Proc. 15th Int. Conf. DASFAA Part II*, Tsukuba, Japan, 2010, pp. 216–230.

[112] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. SIGMOD*, New York, NY, USA, 2000, pp. 93–104.

[113] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *Proc. IEEE Symp. CIDM*, Honolulu, HI, USA, Apr. 2007, pp. 504–515.

[114] J. Zhang, Q. Gao, and H. Wang, "SPOT: A system for detecting projected outliers from high-dimensional data streams," in *Proc. 24th ICDE*, Cancun, Mexico, 2008, pp. 1628–1631.

[115] J. Branch, B. Szymanski, C. Giannella, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," in *Proc. 26th IEEE ICDCS*, 2006, pp. 51–81.

[116] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *Data Mining Knowl. Discov.*, vol. 12, nos. 2–3, pp. 203–228, May 2006.

[117] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proc. 32nd Int. Conf. VLDB*, Seoul, Korea, 2006, pp. 187–198.

[118] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Distributed deviation detection in sensor networks," *SIGMOD Rec.*, vol. 32, no. 4, pp. 77–82, Dec. 2003.

[119] N. A. C. Cressie, *Statistics for Spatial Data*. New York, NY, USA: Wiley, 1993.

[120] M. Jun, H. Jeong, and C. Kuo, "Distributed spatio-temporal outlier detection in sensor networks," in *Proc. SPIE*, 2005, pp. 760–763.

[121] C. Franke and M. Gertz, "Detection and exploration of outlier regions in sensor data streams," in *Proc. IEEE ICDMW*, Pisa, Italy, 2008, pp. 375–384.

[122] C. Franke and M. Gertz, "ORDEN: Outlier region detection and exploration in sensor networks," in *Proc. ACM SIGMOD*, Providence, RI, USA, 2009, pp. 1075–1078.

[123] D. Birant and A. Kut, "Spatio-temporal outlier detection in large databases," *J. Comput. Inform. Technol.*, vol. 14, no. 4, pp. 291–297, 2006.

[124] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd ACM Int. Conf. KDD*, 1996, pp. 226–231.

[125] T. Cheng and Z. Li, "A hybrid approach to detect spatial-temporal outliers," in *Proc. 12th Int. Conf. Geoinformatics*, 2004, pp. 173–178.

[126] T. Cheng and Z. Li, "A multiscale approach for spatio-temporal outlier detection," *Trans. GIS*, vol. 10, no. 2, pp. 253–263, 2006.

[127] N. R. Adam, V. P. Janeja, and V. Atluri, "Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets," in *Proc. ACM SAC*, Nicosia, Cyprus, 2004, pp. 576–583.

[128] E. Wu, W. Liu, and S. Chawla, "Spatio-temporal outlier detection in precipitation data," in *Proc. 2nd Int. Conf. Sensor KDD*, Las Vegas, NV, USA, 2010, pp. 115–133.

[129] C.-T. Lu and L. R. Liang, "Wavelet fuzzy classification for detecting and tracking region outliers in meteorological data," in *Proc. 12th Annu. ACM Int. Workshop GIS*, 2004, pp. 258–265.

[130] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proc. IEEE 24th ICDE*, Washington, DC, USA, 2008, pp. 140–149.

[131] Y. Ge *et al.*, "Top-eye: Top-k evolving trajectory outlier detection," in *Proc. 19th ACM Int. CIKM*, Toronto, ON, Canada, 2010, pp. 1733–1736.

[132] X. Li, Z. Li, J. Han, and J.-G. Lee, "Temporal outlier detection in vehicle traffic data," in *Proc. 2009 IEEE ICDE*, Shanghai, China, pp. 1319–1322.

[133] B. Pincombe, "Anomaly detection in time series of graphs using ARMA processes," *ASOR Bull.*, vol. 24, no. 4, pp. 2–10, 2005.

[134] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina, "Web graph similarity for anomaly detection," in *Proc. 17th Int. Conf. WWW*, 2008, pp. 1167–1168.

[135] P. Shoubridge, M. Kraetzl, and D. Ray, "Detection of abnormal change in dynamic networks," in *Proc. Int. Conf. IDC*, Adelaide, SA, Australia, 1999, pp. 557–562.

[136] K. M. Kapsabelis, P. J. Dickinson, and K. Dogancay, "Investigation of graph edit distance cost functions for detection of network anomalies," in *Proc. 13th Biennial CTAC*, vol. 48. Oct. 2007, pp. C436–C449.

[137] P. Dickinson, H. Bunke, A. Dadej, and M. Kraetzl, "Median graphs and anomalous change detection in communication networks," in *Proc. Int. Conf. Inform., Decis. Control*, Adelaide, SA, Australia, Feb. 2002, pp. 59–64.

[138] M. Gaston, M. Kraetzl, and W. Wallis, "Graph diameter as a pseudo-metric for change detection in dynamic networks," *Austr. J. Combinatorics*, vol. 35, pp. 299–312, Jun. 2006.

[139] P. Dickinson and M. Kraetzl, "Novel approaches in modelling dynamics of networked surveillance environment," in *Proc. 6th Int. Conf. Inform. Fusion*, vol. 1. Cairns, QLD, Australia, 2003, pp. 302–309.

[140] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina, "Web graph similarity for anomaly detection," *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 19–30, 2010.

[141] M. Gupta, C. C. Aggarwal, and J. Han, "Finding top-k shortest path distance changes in an evolutionary network," in *Proc. 12th Int. Conf. Adv. SSTD*, Minneapolis, MN, USA, 2011, pp. 130–148.

[142] T. Idé and H. Kashima, "Eigenspace-based anomaly detection in computer systems," in *Proc. 10th ACM Int. Conf. KDD*, Seattle, WA, USA, 2004, pp. 440–449.

[143] L. Akoglu and C. Faloutsos, "Event detection in time series of mobile communication graphs," in *Proc. Army Science Conf.*, 2010.

[144] M. Gupta, C. C. Aggarwal, J. Han, and Y. Sun, "Evolutionary clustering and analysis of bibliographic networks," in *Proc. 2011 Int. Conf. ASONAM*, pp. 63–70.

[145] M. Gupta, J. Gao, and J. Han, "Community distribution outlier detection in heterogeneous information networks," in *Proc. 2013 ECML PKDD*, Prague, Czech Republic, pp. 557–573.

[146] Y. Sun *et al.*, "Detecting spatio-temporal outliers in climate dataset: A method study," in *Proc. 2005 IEEE IGARSS*, pp. 760–763.

[147] W. Drosdowsky, "An analysis of Australian seasonal rainfall anomalies: 1950–1987," *Int. J. Climatol.*, vol. 13, no. 1, pp. 1–30, 1993.

[148] R. Lasaponara, "On the use of principal component analysis (PCA) for evaluating interannual vegetation anomalies from SPOT/VEGETATION NDVI temporal series," *Ecol. Modell.*, vol. 194, no. 4, pp. 429–434, 2006.

[149] D. Dasgupta and S. Forrest, "Novelty detection in time series data using ideas from immunology," in *Proc. 5th Int. Conf. Intelligent Systems*, Reno, NV, USA, 1996.

[150] V. Jakkula and D. J. Cook, "Anomaly detection using temporal data mining in a smart home environment," *Methods Inform. Med.*, vol. 47, no. 1, pp. 70–75, 2008.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.

[151] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Proc. 1999 IEEE Symp. Security Privacy*, Oakland, CA, USA, pp. 133–145.
[152] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *Proc. 4th ACM SIGCOMM IMC*, Taormina, Italy, 2004, pp. 201–206.
[153] Z. Guo, G. Jiang, H. Chen, and K. Yoshihira, "Tracking probabilistic correlation of monitoring data for fault detection in complex systems," in *Proc. ICDSN*, Philadelphia, PA, USA, 2006, pp. 259–268.
[154] G. Jiang, H. Chen, and K. Yoshihira, "Modeling and tracking of transaction flow dynamics for fault detection in complex systems," *IEEE Trans. Depend. Secur. Comput.*, vol. 3, no. 4, pp. 312–326, Oct. 2006.
[155] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras, "On the spatiotemporal burstiness of terms," *Proc. VLDB*, vol. 5, no. 9, pp. 836–847, 2012.
[156] M. Mathioudakis, N. Bansal, and N. Koudas, "Identifying, attributing and describing spatial bursts," *Proc. VLDB*, vol. 3, no. 1, pp. 1091–1102, 2010.
[157] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on Enron graphs," *Comput. Math. Org. Theory*, vol. 11, no. 3, pp. 229–247, Oct. 2005.
[158] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data," in *Proc. 13th SIAM Int. Conf. SDM*, 2013.
[159] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 823–839, May 2012.
[160] M. Markou and S. Singh, "Novelty detection: A Review—Part 1: Statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.
[161] M. Markou and S. Singh, "Novelty detection: A review—Part 2: Neural network based approaches," *Signal Process.*, vol. 83, no. 12, pp. 2499–2521, 2003.
[162] W. Yu, C. C. Aggarwal, S. Ma, and H. Wang, "On anomalous hotspot discovery in graph streams," in *Proc. 2013 IEEE ICDM*, Dallas, TX, USA.

Manish Gupta is an Applied Researcher at Microsoft Bing, Gachibowli, Hyderabad, India, and also a visiting faculty member at IIT-Hyderabad, Hyderabad, India. He received the master's degree in computer science from IIT Bombay, Mumbai, India, in 2007, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA, in 2013. He was with Yahoo!, Bangalore, India, from 2007 to 2009. His research interests are in the areas of data mining, information retrieval, and web mining.

Jing Gao received the Ph.D. degree from the University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA, in 2011. She is an Assistant Professor with the Computer Science and Engineering Department, University at Buffalo, State University of New York, Buffalo, NY, USA. She was the recipient of an IBM Ph.D. Fellowship. She is broadly interested in data and information analysis with a focus on information integration, ensemble methods, transfer learning, anomaly detection, and mining data streams. She is a member of the IEEE.

Charu C. Aggarwal is a Research Scientist with the IBM T. J. Watson Research Center in Yorktown Heights, NY, USA. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1996. He has worked in the the field of performance analysis, databases, and data mining. He has published over 200 papers in refereed conferences and journals, and has applied for or been granted over 80 patents. He has received the IBM Corporate Award in 2003, the IBM Outstanding Innovation Award in 2008, the IBM Research Division Award in 2008, and the Master Inventor at IBM thrice. He is a fellow of the ACM and the IEEE.

Jiawei Han is an Abel Bliss Professor with the Department of Computer Science at the University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA. His research interest include data mining, information network analysis, and database systems, with over 500 publications. He received the ACM SIGKDD Innovation Award in 2004, the IEEE Computer Society Technical Achievement Award in 2005, and the IEEE W. Wallace McDowell Award in 2009. His book *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been used worldwide as a textbook. He is a fellow of the ACM and the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 18,2021 at 04:04:26 UTC from IEEE Xplore. Restrictions apply.