Olivier Markowitch

Dimitrios Sisiaridis

22 Sep 2015

The Brufence project

Scalable Machine Learning for Automated Defence System

Automatic detection of threats and frauds in communication systems and payment transactions

Work Package 2: “Communication Systems Security - Detection of Threats and Attacks on Managed File Transfer and Collaboration Platforms”

Task 2.2: Analysis of existing techniques and products

Table of Contents

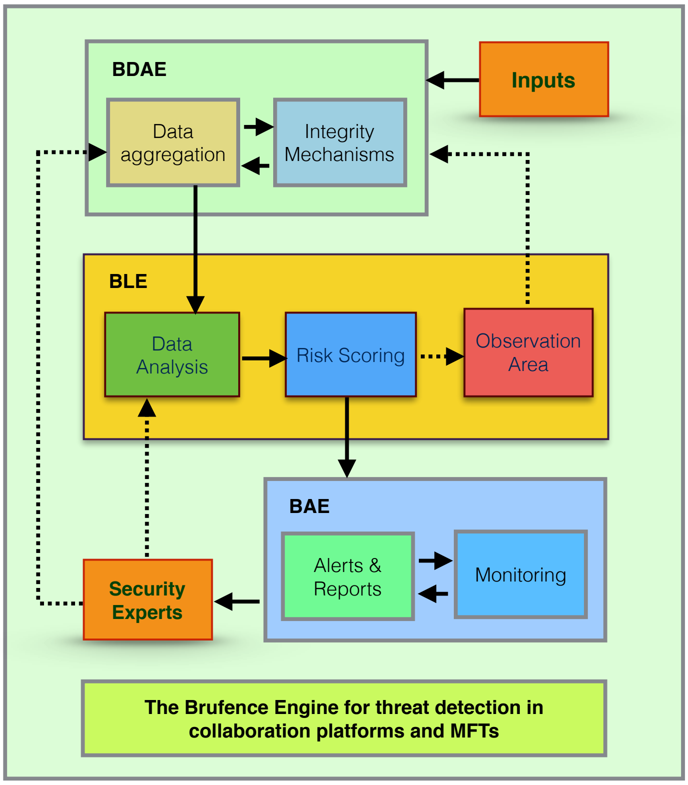
Introduction

Work Package 2 of the Brufence project aims to the design of a near real-time scalable modular framework for deploying Machine Learning algorithms for Big Data predictive analytics in the field of threat detection in collaboration platforms and MFTs. Different models will be compared in parallel in order to maximise their efficacy in predicting *abnormal behaviour*, evaluated for their performance in terms of scalability, accuracy, readability and QoS.

Automated traffic log analysis of temporal and spatial data, over a long period of time, will be engineered for advanced proactive threat protection. Real time log data acquired by network devices in a centralised log management system following e.g. an SDN approach, will be correlated with attack communication profiles, derived from a learning set of identified baseline behaviour profiles, representing a complete picture of how an adversary acts in a variety of environments aiming to achieve a rapid and accurate identification of threat patterns in order to detect or even to predict APTs, DDosS attacks and zero-day attacks.

Unstructured and structured data with respect to security-related events from users, services and the underlying network infrastructure, with a high level of large dimensionality and non-stationarity as well as temporal aspects, will be correlated and classified, in order to detect *abnormal* behaviours that deviate from *normal* behaviour. *Incident correlation* integrated with complex event processing techniques will be used to compare different events, often from multiple data sources in order to identify patterns and relationships enabling identification of events belonging to an attack or, indicative of broader malicious activity, allowing thus to a better understanding of the nature of an event, to reduce the workload needed to handle incidents, to automate the classification and forwarding of incidents only relevant to a particular consistency and thus to reduce security noise and false positives.

This processing and analysis of complex events will be materialised by deploying machine learning algorithms in terms of *outlier detection analysis* of threats and attacks in collaboration platforms and MFTs. The most significant challenge for an evaluation of threat and attack detection algorithms is the lack of any appropriate public relevant datasets, mainly due to restriction rules applied on sensitive data by organisations and enterprises regarding confidentiality as well as potential security breaches.



An overview on outlier detection analysis techniques of threats and attacks

Key Features

An *outlier* is a data point or a sequence of multiple data points recognised as deviations from the remaining *normal* data. Often it contains useful information about abnormal characteristics of the systems and entities. It could either be considered an *abnormality* or *noise*. In the case of a sequence of events, anomalies are also referred to as 'collective anomalies'.

- *Noise* is often modelled as a weak form of outliers. It does not always meet the strong criteria necessary for a data point to be considered interesting or anomalous enough such as:
  - the sparsity of the underlying region
  - nearest neighbour based distance
  - a fit to the underlying data distribution

An *outlier detection algorithm:*

- creates a model of the normal patterns in the data
- computes an *outlier score* of a given data point on the basis of the deviations from these patterns
  - a *threshold* on this score is used in order to declare data points as outliers

There are significant problems, derived from the choice of the threshold value such as the existence of:

- many false negatives i.e. missing true outlier points
- false positives, when a big number of data points is declared.

G*round-truth* can be used to evaluate the effectiveness of an algorithm in terms of its:

- *precision* i.e. the percentage of reported outliers, which truly turn out to be outliers
- *recall* i.e. the percentage of ground-truth outlier

Examples of outlier scoring mechanisms can be given either in:

- *probabilistic modelling*
  - such as the likelihood fit of a data point to the model is the outlier score
  - outliers are defined by the relative positions of the data values with respect to each other
- proximity-based modelling
  - based on the k-nearest neighbour distance
  - based on the distance to closest cluster centroids
  - based on the local density value
- linear modelling

  - the residual distance of a data point to a lower-dimensional representation of the data
- temporal modelling
  - a function is used to create the outlier score, using either:
    - the distance from previous data points
    - the deviation from a forecasted value).

An *outlier model* characterises the normal behaviour of the data and it is usually *dataset-specific*.

- Any deviations from this model determine outliers

More specifically, the *interpretability* of the model, known also as the *intentional knowledge*, explains why a data point is an outlier. Its choice is based on several factors of the dataset such as:

- data type
- data size
- availability of data size examples

There are several outlier models:

- In *probabilistic* and *statistical models*, data (of a scalar or mixed data type) are modelled with a probability distribution, usually using the *Expectation-Maximisation* (EM) algorithm, in terms of the membership probability of the data points to the different clusters and density-based fit to the modelled distribution
  - An *extreme value analysis* determines the outliers
  - In case of high number of parameters, outliers may overfit the underlying model of normal data (over-fitting)
    - The method can be applied in one-dimensional (univariate) data
    - Values which are either too large or too small are outliers and can be found at the statistical tails of the underlying distribution
    - it is usually used as the final step in outlier detection, when deviations are represented as univariate values and can be extended to multivariate data, using distance-based, or depth-based methods
- In *linear models*, data are modelled into lower dimensional embedded subspaces using *linear correlations* based on *regression analysis*
  - *A least squares fit* determines the optimal lower dimensional subspace
  - Distances of data points from this plane are determined first
  - Extreme values analysis then determines the outliers on the basis of the derived residuals while subspaces are provided by PCA
- *Spectral models* can be applied to graphs and networks for clustering graph datasets in order to identify anomalous changes in temporal sequences of graphs using matrix decomposition methods (such as PCA)
- In *proximity-based models*, *outliers* are modelled as points which are isolated from the remaining data, using either:

  - *cluster analysis*
    - data points are segmented by determining dense regions of the data set and then an outlier score is measured using e.g. the distance of the nearest centroid
      - hierarchical cluster analysis
      - k-means clustering
  - *density-based analysis*
    - space is segmented, by looking for sparse regions in the data that can be presented in terms of combinations of the original attributes, using either nearest neighbour analysis or by determining the distance of each data point to its k-th nearest neighbour
- *Information theoretic models* aim to a concise *representation* of the dataset using conventional techniques for coding representation such as frequent pattern mining, histograms or spectral methods
  - The key idea is to construct a *code book* that represents the data
  - Outliers are defined as points which their removal results either in the largest decrease in description length, or in the most accurate summary representation in the same description length[4][81][101] [46]
- Finally, when there is *high dimensionality,* data become sparse
  - All pairs of data points become almost equidistant from one another
  - Several dimensions may be very *noisy*
  - Outliers are discovered in a lower dimensional local subspace, using either:
    - clustering analysis
    - regression analysis

Ensembles methods are used to improve the quality of outlier detection algorithms. An ensemble X is a triple (x, A_x, P_x) where the outcome x is the value of a random variable, which takes on one of a set of possible values $Ax=\{a_1,a_2,\dots,a_i,\dots a_I\}$ having probabilities $P_x=\{p_1,p_2,\dots p_I\}$, with $P(x=a_i)=p_i$, $p_i>=0$ and $\Sigma_{a_i}P(x=a_i)=1$. A measure of the information content of the outcome $x=a_i$ in an ensemble X, is the Shannon information content $h(x=a_1)=\log_2 1/p_i$. The *entropy* of the ensemble $H(x)=\Sigma_i p_i \log_2 1/p_i$ is a measure of the ensemble's average information content.

- in *sequential ensembles* one or more outlier detection algorithms are applied sequentially to either all or portions of the data only for a small number of constant passes. In each iteration, a successively refined algorithm may be used on refined data
- *Independent ensembles* on the other hand can be used in high-dimensional datasets. Different instantiations of the algorithm or different portions of the data are used. In some cases, the same algorithm may be applied with a different initialisation, parameter set, or a random seed as it is in the case of a randomised algorithm.

In a collaboration platform or in managed file transfer, there are relationships among the data points of the dataset set of values, generated by continuous measurement over time (*temporal* data). *Time-series* data that form complex events, derived by the underlying network, middleware and AAA services.

- In this case, outlier detection is highly related to the problem of *change detection* where normal models of data values are highly governed by *adjacency. Discrete Sequences* can be either temporal (categorical or time series data) or not temporal (based on their relative placement with respect to one another). New data values are referred to as *novelties* [75][74][71].

- A change could happen either slowly over time (known as *concept drift*) or abruptly.
  - In the first case, it can only be detected by a detailed analysis over a long period of time while it does not necessarily corresponds to outliers
  - in the second on, a suspicion of a possible change to the underlying data generation mechanism, detected with real-time analysis, can be taken as an indication of an anomaly, in terms of an Advanced Persistent Threat.

In *spatial data*, non-spatial attributes are measured at spatial locations. Any unusual local changes in such values are reported as outliers.

- Spatiotemporal data are a generalisation of both spatial and temporal data. Their analysis can enlighten attacker's intentions in kill chains and thus, helping the detection and tracing back of zero-day attacks.

In *network* and *graph data,* data values correspond to *nodes (*structural data) while relationships among the data values correspond to the *edges.* Outliers may be modelled in different ways, depending upon any *irregularity* of nodes, in terms of their relationships to other nodes, or of edges.

Supervised outlier detection algorithms are generally, designed for *anomaly detection*, rather than *noise removal*. They form a special case of the *classification problem*; labels are extremely unbalanced in terms of relative presence, known as the *rare class* detection problem. There are several variations such as the *Positive Un-labeled Classification* (PUC) problem [33], where a limited number of instances of the positive (outlier) class may be available while *normal* examples may contain an unknown proportion of outliers.

- *Semi-supervised novelties* can be either, only instances of a subset of the normal and anomalous classes may be available, where some of the anomalous classes may be missing from the training data [72][123], or labeled examples of all normal and some anomalous classes are available, where labels for the anomalous classes are not exhaustive.
- in the case of *active learning*, human feedback is utilised in order to identify relevant outlier examples [83].

State-of-the-art outlier detection techniques and their implication to threat detection in communication systems

PROBABILISTIC AND STATISTICAL MODELS FOR OUTLIER DETECTION

Probabilistic and *statistical models* for outlier detection are usually engineered for finding outliers at the outer boundaries of a data space. They present limited ability to distinguish between noise and abnormalities due to several simplifying assumptions, especially for real or large data sets.

- For example, statistical methods for extreme value analysis are used in univariate data distributions
- Extreme values in a probability distribution are referred as the *distribution tail*
  - very low probability values of a tail should be considered anomalous

Probabilistic tail inequalities can be used to bound the probability that a value in the tail of a probability distribution should be considered anomalous, such as the:

- *Markov inequality*
  - for distributions, which take on only non-negative values
- *Chebychev inequality*
  - a direct application of the Markov inequality to a non-negative derivative, square deviation-based, distribution of X
- inequalities in case of a sum of bounded random variables
  - *Chernoff bound inequality*
    - for *Bernoulli* aggregate date values
  - *Hoeffding inequality*
    - data values can be or not Bernoulli
      - In the latter, when sample size is large, then the *Central Limit Theorem* can be used to identify outliers.

In *statistical tail confidence tests*, it is assumed that data values follow a *normal distribution,* expressed as a function of *Z-number.*

- *Normal distribution tables* are used to map the different values of z_i to probabilities
- In case that deviations are aggregated together (e.g., in multidimensional data) as for example in DDoS attacks, of models produced using PCA, they are modelled to be statistically independent of one another by using the cumulative probability tables of the χ^2-distribution as the sum of squares of deviations.
- A *t-value test* can be applied in scenarios where little domain knowledge of the data may be available.
  - a *t-distribution is* a function of several independent identically distributed standard normal distributions based on the *degrees of freedom v;* if v> 1000, it converges to normal distribution

Extreme Value analysis in multivariate data can be used for finding outliers within the inner regions of the data space. There is a variety of methods to deploy:

- D*epth-based Methods*
  - *they* make use of convex hull analysis to find outliers by applying an iterative algorithm
  - complexity is increased exponentially with dimensionality
- *Deviation-based* methods are distribution independent
  - They measure the impact of outliers on the data variance
  - Outliers are defined as exception sets; their removal causes the maximum reduction in variance of the data
- A*ngle-based methods*

  - data points at the boundaries of the data are likely to enclose the entire data within a smaller angle
  - data points in the interior are likely to have data points around them at different angles
  - data points with a smaller angle spectrum are taken as outliers
- *Distance Distribution-based Methods* use a multivariate Gaussian distribution
  - The *Mahalanobis distance* between a data point X and the mean µ of the data is used, especially:
    - in cases with increasing dimensionality
    - in cases where the entire data set is distributed in one large cluster about the mean.

A generative statistical model is one that specifies a full probability density over all variables. Such a model can make use of latent variables to describe a probability distribution over observables [113].

- Probabilistic mixture models are typically generative models.
- They provide an estimation of the *generative* or *fit probability* for each data point, usually by using the *EM algorithm.*
- In this case, anomalies will have very low fit probabilities.
- Other latent variable models include mixture models, Hidden Markov models and factor analysis models.

LINEAR MODELS FOR OUTLIER DETECTION

In *linear regression*, data are embedded in a lower dimensional subspace, related to one another using a set of linear coefficients.

- The number of observed values are typically much larger than the dimensionality.
- *Attributes* are usually generated by the same underlying process.
  - they are highly correlated with one another and are used extensively to detect anomalies in time-series data.
- The *square error* of the deviations of data points is optimised from values predicted by the model.

Linear regression does not try to recognise cases where correlation behaviour in different localities may be different but instead, it tries to fit the data into a single global model.

- When a particular variable is considered special then an optimal *plane* is determined by minimising the *mean-square error* of the *residuals.*
- If all variables are treated in a similar way then an optimal regression plane is determined by minimising the *projection error* of the data to the plane.

Principal Component Analysis (PCA) is used to find *optimal representation hyperplanes* of any dimensionality, recursively. It is much more *stable* to the presence of a few outliers. When the scales of the different dimensions are very different, then, results may be very informative.

- Data need to be *normalised.*

- Each dimension is divided with its standard deviation, producing a correlation matrix instead of the covariance matrix (without normalisation)
  - in order to analyse e.g. DDoS attacks as well as long-term attacks.
- The *k-dimensional hyperplane* (for any value of k < d) minimises the squared projection error.
- The *d×d covariance matrix* over d-dimensional data is computed.
- Then, it is diagonalized as $\Sigma = P \cdot D \cdot P^T$.
  - D is a diagonal matrix whereas its entries provide the eigenvalues.
  - P is an orthonormal matrix. Its columns correspond to the *eigenvectors* of Σ.
  - The eigenvectors correspond to directions of correlations in the data
    - a small number of eigenvectors can capture most of the variance in the data.

A similar approach to PCA, called *Latent Semantic Indexing* has also been used in the context of text data for the reduction in the noise effects, such as:

- *synonymy*
  - the same concept can be represented with multiple words
- *polysemy*
  - a word may mean multiple things) [26] [82]

PROXIMITY-BASED OUTLIER DETECTION

Proximity-based algorithms aim to defect both noise and anomalies. Thus, a data point is defined as an outlier, if its *locality* (or *proximity*) is sparsely populated, as it is the case in most zero-day attacks. Several methods have been proposed to define the proximity of a data point:

- *cluster-based methods*
  - they can be used in *sparse* data domains, for noise detection, where most attributes take on zero values, based on criteria such as:
    - the *non-membership* in any cluster
    - *distance* from other clusters
      - defined in terms of the distance to cluster centroids
      - normalised using the Mahalanobis distance
    - the *size* of the closest cluster [2]
- *distance-based* methods
  - often used for anomaly detection [52]
  - The *distance* of a data point *to its k-nearest neighbour* (or any other variant):
    - is normalised using the Mahalanobis distance
    - is calculated to define outliers
  - variations:
    - cell-based

    - index-based
    - partition-based
    - reverse nearest neighbour approach
- *cell-based methods*
  - data space is divided into cells while their width is a function of a threshold D, and the data dimensionality
  - they provide effective pruning of the distance computations in low dimensionality
  - they make use of grid-based localisation in order to reduce the number of false positives
- *index-based methods*
  - data points are ranked in decreasing order of the k-nearest neighbour distance
  - the top n such data points are reported as outliers [53][89]
- *Partition-based speedup* methods
  - they are used to prune away those data points which could not possibly be outliers in a computationally efficient way
  - for each partition generated by a clustering algorithm, a lower and an upper bound are computed of the k-nearest neighbour distances of all included data points
- *reverse nearest neighbour approach*
  - the number of reverse k-nearest neighbours is used to define outliers based on a predefined threshold.
- *density-based* methods
  - proximity is defined using the number of other points within a specified local region
    - grid region
    - distance-based region
    - local density values may be converted into outlier scores
- *kernel-based methods* or statistical methods for *density estimation:*
  - the *Local Outlier Factor* (LOF) [14]
    - provides a quantification by adjusting the variations in the different densities
  - The *Local Correlation Integral* (LOCI)
    - defines the density M(X,ε) of a data point X in terms of the number of data points within a pre-specified radius ε around a point, a method known as the *counting neighbourhood* of the data point X.
    - A data point is an outlier if its *multi-granularity deviation factor (*MDEF*)* value is unusually large among any of the values computed at different granularity levels.
  - *Histograms* are particularly suitable for *density-based summarisation* of univariate data
    - Data are discretised into *bins*
    - *T*he *frequency* of each bin is estimated
    - data points which lie in bins with very low frequency are reported as outliers
    - they have wide applicability in intrusion detection where *normal data* are modelled with the use of *histogram-based profiles*

- *Kernel Density Estimation*
    - similar to histogram-based techniques, in terms of building *density profiles*
    - *Accuracy* degrades with increasing dimensionality
    - A continuous estimate of the density is generated at a given point as the sum of the smoothed values of kernel functions K_h' $(\cdot)$ associated with each point in the data set
    - Each kernel function is associated with a *kernel width* h, which determines the *level of smoothing* created by the function [100].

HIGH-DIMENSIONAL OUTLIER DETECTION ANALYSIS

According to the *curse of dimensionality,* in high-dimensional space, data become sparse (data sparsity); data points are situated in an almost equally sparse regions in full dimensionality [1]. In such scenarios as it is common in cloud installations of a collaboration platform or secure managed file transfer in enterprises, *strong* outliers (i.e. anomalies) may only be discovered in low dimensional subspaces of the data, as they are usually hidden by the *noise effects* of multiple dimensions, such as *masking* and *dilution.*

- *Rarity-based* methods can be used to discover the *subspaces* based on rarity of the underlying distribution
- In *unbiased* methods, *subspaces* are sampled in an unbiased way while *scores* are combined across different subspaces
- In *aggregation-based* methods, the relevance of the subspaces is determined using aggregate statistics such as *cluster*, *variance* or *non-uniformity* statistics of local or global subsets of the data
- *Projected outliers* are determined by finding localised regions of the data in low dimensional space, having abnormally low density
- *Abnormal lower dimensional projections,* a grid-based approach, can be used in order to determine projections of interest where each attribute of the data is divided into ɸ *ranges* on an equidepth basis, e.g. for analysing data from a MS Sharepoint farm
- *Evolutionary algorithms* [47] imitate the process of organic evolution to solve parameter optimisation problems
- *Distance-based subspace algorithms* can be used in lower dimensional subspaces where either outliers are determined by exploring relevant subspaces or relevant outlying subspaces for a given data point are determined, providing thus *intensional knowledge* and increased *interpretability*
    - In the case where outliers are defined from *multiple subspaces,* a given data point may show very different behaviour in terms of its *outlier degree* in different subspaces
    - *outlier scores* from different subspaces may all be very different
    - independent ensembles methods such *Random Subspace Sampling* or methods which select high contrast subspaces can be used [61]
        - if many dimensions are *noisy*, usually a small number of them is included in each subspace sample, and thus a larger number of subspace samples will be required in order to obtain more robust results; subspaces with an unexpected *non-uniformity* are more likely to contain outliers.
- *Generalized Subspaces* algorithms are used in finding outliers in *axis-parallel subspaces.*

- They can be effective for finding outliers in cases where the outliers naturally deviate in specific subspaces from the clusters while they are not very useful in finding clusters, where the points are aligned along lower-dimensional manifolds of the data
- they are generalisations of either *PCA-based linear models* that find the global regions of correlation in the data or *axis-parallel subspace models* that can find deviations when data are naturally aligned along low dimensional axis-parallel subspace clusters.

SUPERVISED OUTLIER DETECTION

The goal of the supervised outlier detection is to provide application-specific knowledge into the outlier analysis process in order to obtain more meaningful anomalies with the use of learning methods [60][67]. In most real data domains some examples of normal or abnormal data may be available. They are referred to as *training data* and can be used to create a classification model which it distinguishes *normal* and *anomalous* instances. Problems that can be tackled with supervised analysis include:

- the *class imbalance problem*
  - distribution between the *normal* and *rare class* will be very skewed
  - optimisation of *classification accuracy* may not be meaningful as *false positives* are more acceptable than *false negative*
    - it leads thus to cost-sensitive variations of the classification problem, in which the objective function for classification is changed
  - usually only a small number of rare instances may be available
  - it may be expensive to acquire examples of the rare class
  - usually, training algorithms which show differentially overfitting behaviour
    - the algorithm may behave robustly for the normal class, but may overfit the rare class
- the *positive unlabelled class problem*
  - refers to contaminated normal class examples where the data may originally be present in *unlabelled* form [64]
  - manual labelling is performed for annotation purposes
  - only the positive class is labeled while the remaining normal data contain some abnormalities
  - *anomalies* may be treated as contaminants
- *semi-supervised* or *novel class detection*
  - cases with only *partial training information*
  - examples of one or more of the anomalous classes may not be available
    - in intrusion detection, there are may be examples of the normal class and some of the intrusion classes, as new kinds of intrusions arise with time
  - a particularly case is the *one class variation* where only examples of the normal class are available
  - having unlabelled data, a positive class may be available while examples of the negative class may be much harder to model, as for example, when there is a need to classify or collect all documents (e.g. web documents) which belong to a rare class

    - *contaminants* in the negative class can reduce the *effectiveness* of a classifier, although their use is preferable instead of rather completely discard them
    - collected training instances for the unlabelled class may not reflect the true distribution of documents and thus, the *classification accuracy* may actually be harmed by using the *negative* class [64][121].
  - Variations:
    - *one-class novelty detection*
      - only the normal class is specified
      - The goal is to determine classes which are as different as possible from the specified training class
        - for example, in temporal data, *novelties* are defined continuously based on the past behaviour of the data
    - *novel-class detection with rare-class detection*
      - labeled rare classes are present in the training data
      - novel classes may also need to be detected
      - different kinds of anomalies are distinguished from one another in terms of whether they are found in a supervised or unsupervised way
        - if the test point is a *natural fit* for a model of the training data, then, a variety of unsupervised models such as clustering can be use
          - if not, then the test point is flagged as an *outlier*, or a *novelty*
        - If the test point is a fit for the training data, a classifier model can be used to determine whether it belongs to one of the rare classes
          - alternatively, any cost-sensitive model can be used
    - *online novelty detection*
      - in *concept drifting data streams*
        - usually there is an implicit assumption of *temporal* data [74][6]
        - the goal is to understand natural complementary relationships between clusters (normal unsupervised models) and novelties (temporal abnormalities), in the underlying data
        - classes can be defined as *novelty* only in terms of what has already been seen in the past.
      - In case of having only unlabelled data:
        - unsupervised clustering methods can be used in order to identify significant novelties in the stream
        - novelties occur as emerging clusters in the data which eventually become a part of the normal clustering structure of the data.
- *fully-supervised or rare-class problem*
  - *evaluation* and *model construction* are closely related
  - the rare class distribution has to be defined
    - it relates to the objective function of a classification algorithm

  - the next step is to identify any algorithmic changes required in order to incorporate the modifications to the modelling assumptions.

Several algorithms can be deployed such as *class imbalance* and *adaptive resampling* algorithms. In the first case of class implalance, cost-sensitive learning methods such as *relabelling* and *weighting methods* can be used to learn a classifier that maximises the *weighted accuracy* over the different classes.

- The objective function is modified in order *to weight the errors* in classification differently for different classes; in this way, classes with greater *rarity* have higher costs
- *MetaCost, a* relabelling approach [30]*,* relabels some of the training instances in the data by using the costs while normal training instances are relabelled to that rare class.

Weighting methods modify the classification algorithm to implicitly treat each training instance with a weight which corresponds to its *misclassification cost.* There are several examples:

- *Bayes classifier* is equivalent to multiplying the Bayes probability in the unweighted case with the cost, and picking the largest one
  - a posterior probability then can be inferred in terms of the likelihood, prior probability and evidence as *posterior probability=(likelihood x prior probability)/evidence*
- in p*roximity-based classifiers,* the number of k-nearest neighbours for each class can be multiplied with the corresponding cost for that class while the majority class is picked after the weighting process
- in *rule-based classifiers,* a *rule* relates a condition in the data (e.g., with numeric attributes) to *a class label*
  - then, *frequent pattern* mining algorithms may be adapted to determine the relevant rules at a given level of:
    - *support*
      - the number of training instances which are relevant to that rule
    - *confidence*
      - the fractional probability that the training instance belongs to the class on the right hand side, if it satisfies the conditions on the left-hand side
- in *decision trees,* training data are recursively partitioned using various *entropy* measures while instances of different classes are successively separated out at lower levels of the tree
- in *SVM classifiers, hyperplanes* are learned by the classifiers which optimally separate the two classes in order to minimise the expected error.

By deploying *adaptive resampling,* data are resampled so as to magnify the relative proportion of the rare classes [19][56].

- The process can be performed either *with or without replacemen*t:
  - rare class can be oversampled
  - normal class is under-sampled
  - or even both

- sampling probabilities are typically chosen in proportion to their misclassification costs in order to enhance the proportion of the rare costs in the sample used for learning [20]
- Variations:
    - *synthetic oversampling*
    - *one-class learning with positive class*
        - normal class examples are removed from the data while the representative data contain only anomalies
    - *ensemble techniques*
        - data instances are repeatedly classified with different samples
        - *majority vote* is used for predictive purposes
    - *boosting methods*
        - they are used in classification to improve the *classification performance* on difficult instances of data
        - A typical example is the *Adaboost* algorithm [94}
            - each *training example* is associated with a *weight,* which is updated in each iteration depending upon the results of the classification in the last iteration
            - instances which are misclassified, are given higher weights in successive iterations
        - A similar approach is followed in the *Adacost* method [34].

While working with a *novel-class detection* scenario several methods can be used to cope with outlier detection such as *k-nearest neighbour models*, *SVM models* [95], *extreme value methods* [92] or *kernel-based PCA* [44].

There are cases where a human expert may get involved in the outlier detection process in order to improve the *effectiveness* of the underlying algorithms. Thus for example, when the original dataset is large and the vast majority of examples are normal, an unsupervised or supervised outlier detection algorithm pre-filters results to a user for getting feedback. An alternative is when user-provided examples are combined with results from an unsupervised algorithm, in order to learn which outliers determined by the unsupervised algorithm are relevant; combined results can then be used in order to train a traditional rare class detection model.

- The first method is known as *active learning* where unlabelled data are analysed in an iterative procedure. In each iteration, a number of interesting instances are identified, for which the addition of labels would be helpful for further classification.
    - A human expert provides labels for these examples which are then used to classify the dataset.
- In the second case, known as *outlier-by-example* method, an unsupervised approach is utilised to perform *feature transformations* on the examples in order to augment the user provided examples by comparing the deviations of the objects with those of the user-provided examples
    - additional labelling and augmentation is done with the use of automated techniques [83].

OUTLIER DETECTION IN CATEGORICAL, TEXT AND MIXED ATTRIBUTE DATA

All classes of algorithms can be adapted for categorical data with suitable modifications, although *proximity-* and *density-based techniques* are the most promising to be developed. Categorical data can be transformed to binary data by treating each value of the categorical attribute as a binary attribute. Existing algorithms for numerical data can also be applied to this case when the number of possible values of a categorical attribute is relative small; a separate binary field needs to be dedicated to each distinct value of the categorical attribute

- In categorical data, a d-dimensional dataset contains N records denoted by D.
- For a *purely categorical* dataset, it is assumed that the i-th attribute contains n_i possible distinct values, while for a *mixed-attribute* data set, the first d_c attributes are assumed to be categorical and the remaining are assumed to be numerical.

By choosing to extend probabilistic models, using an appropriate generative model for categorical values

- outliers can be declared as:
  - data points with low *assignment probability* to their best matching cluster
  - low *absolute fit* to the generative model
- *extreme value analysis* can be used on the fit probabilities to determine the relevant outliers.
- in the case of modelling *mixed data:*
  - they are normalised using the EM-algorithm modified, by defining the generative probabilities of each component as a composite function of the different kinds of attributes
    - standard deviations in the similarity values over the two domains with the use of sample pairs of records are determined
    - each component of the similarity value (numerical or categorical) is divided by its standard deviation [103].

Linear models can be extended to categorical datasets by using the conversion from the categorical data set to the binary scenario

- extreme values among the different point-specific deviations are reported as outliers

Proximity models also can be extended for analysing categorical data by studying *similarities* rather than *distances* using aggregate statistical properties of the data which typically correspond to the statistical frequency of attributes in terms of statistical neighbourhoods of data points [24], measuring the *Inverse Occurrence Frequency* or the *Goodall measure*.

- There are several methods to be deployed for this purpose such *density-based or clustering methods*
  - *density-based methods* are more natural to perform in the context of discrete data where frequency profiles can be constructed on different combinations of attribute values.
  - In the case of categorical data the additional step of *discretisation* does not need to be performed when dealing with mixed attribute data, as numerical attributes may be discretised and categorical attributes may be retained in their original form.

Clustering methods make use of the EM algorithm

- the k-means clustering algorithm is itself an example of EM
- *Fit probabilities* can be used to effectively represent *outlier scores.*
- Such methods often discover noise instead of outliers
- other methods need to be defined in order to compute the distances between cluster representatives and individual data points.

All forms of categorical data and numerical data can be transformed to binary data by various forms of discretisation. A significant amount of categorical data is binary in nature. Transaction data in binary form are different from other forms of binary data which are obtained by artificial conversion as they are typically very high dimensional, sparse and often may contain highly correlated patterns.

- By using *subspace methods,* frequent patterns are less likely to occur in outlier transactions, as compared to normal transactions [1].
- The sum of the *support* of all frequent patterns occurring in a given transaction provides the outlier score of that transaction.
- The total sum is normalised by dividing with the number of frequent patterns. It can be omitted from the final score, since it is the same across all transactions [42].
- Frequent patterns with the largest support, which are also not included in the transaction T_i , are considered *contradictory patterns* to $T_{i.}$.

  A transaction which does not have many items in common with a very frequent item-set is likely to be one of the *explanatory patterns* for the T_i being an outlier [99][101].

Transaction data are often temporal in nature. Individual transactions are associated with a *timestamp. A* new transaction which does not reflect the aggregate behaviour of the transactions maintained can be flagged as a *novelty,* for example in the context of fast binary data streams by continuously maintaining clusters in the underlying temporal data stream.

- In such cases, data points which do not naturally fit into any of these clusters are regarded as novelties.
- For each incoming data point, its best *similarity* to the *centroids* of the current set of clusters is determined
  - if this similarity is statistically too low then, the data point is flagged as a novelty and it is placed in a cluster of its own.

Outlier detection in text data can be used for *noise removal,* for example, by using the *Latent Semantic Indexing* algorithm [82]. The goal is the improvement of the underlying data representation in order to reduce the impact of noise and outliers. For this purpose, the detection of interesting anomalies and determining unique segments of text, such as a *first story* in a text stream, can be achieved by using a variety of probabilistic and proximity-based methods. Documents with *low probability* of belonging to their closest cluster or *low similarity* with the current summary of the corpus can be declared as outliers [13].

TIME SERIES AND MULTIDIMENSIONAL STREAMING OUTLIER DETECTION

In scenarios as those with the analysis of operational logs from a collaboration platform or an MFT with time series and multidimensional streaming, outlier detection of unusual events needs to be performed in a time-critical manner, also referred as *streaming outlier detection.*

- Outliers are defined in terms of an *abrupt change detection* in the time-series [118], which corresponds to sudden changes in the trends in the underlying data stream, such as:
  - sudden changes in time-series values, with respect to immediate history
  - distinctive shapes of subsequences of the time series, with respect to long-term history [35] [51]

In multidimensional streaming, data contain individual multi-dimensional points, which are independent of one another. Anomalies could either correspond to:

- *time-instants* at which aggregate trends have changed
- *individual data point novelties*, which vary from these aggregate trends

In contextual anomalies, the values at specific timestamps are classified as outliers. In collective anomalies, entire time-series or large subsequences within a time-series are classified as outliers because of their unusual shapes:

- In case of labelled data (supervised anomaly detection), labels are associated with:
  - time-instants
  - time intervals
  - entire series
  - individual data points (in multidimensional data sets)

Regression-based forecasting models can be used for the detection of *deviation-based outliers* of specific time-instants (*contextual anomalies*). Different types of correlations include those *across time* or *across series*.

- In correlations across time, which are the same as the *temporal continuity* (referred also as *stream filtering),* significant deviations from the expected predictions are experienced as abrupt changes, quite different from the *smooth long-term concept drift* which is also often experienced in data streams.
- In correlations across series, one time series can frequently be used in order to predict another; deviations from such expected predictions can be reported as outliers.

Autoregressive models can be useful in the context of *univariate time-series,* where a value of X_t is defined in terms of the values in the *last window* of length p, known as an *AR(p) model.* The values of the regression coefficients $a_1 \ldots a_p$, c need to be learned from the training data (*previous history* of the time series).

- The model can be combined with a *Moving Average* model (MA Model).
  - The latter predicts subsequent deviations on the basis of the past history of deviations; the new model is called the *Auto-Regressive Moving Average* (ARMA) model.

- In case of *non-stationarity of the time series* (e.g in a *random walk time series*, where data drift away from the mean), the time series first are differencing
  - this new model is referred as the *Autoregressive Integrated Moving Average* Model (ARIMA).

In *multiple time-series regression models,* different time series may often contain the same information with large correlations. Prediction can be accomplished either by using:

- a direct generalisation of auto-regressive models such as *Muscles* or selective *Muscles*
- PCA *and hidden variable-based models*
  - based on treating a single stream as the dependent variable
  - hidden variables can be used to generate all variables in the stream simultaneously
  - PCA can be generalised to window-based analysis by using a window of length p in order to create the covariance matrix
    - most well-known variation is SPIRIT.

Supervised Outlier Detection in time-series can be used to distinguish between *noise* and *true anomalies.*

- *t*rue events of interest are available as the *ground truth timestamps,* referred also as *primary abnormal events*
- timestamps which correspond to the common spurious deviations are referred as *secondary abnormal events.*
- The goal is to create a *composite alarm level* from the error terms in the time series prediction by using a univariate time-series prediction model in order to determine the error terms at a given timestamp.
- The composite alarm level can either be reported as:
  - an outlier score
  - a threshold defined to provide discrete timestamps at which anomalous events are reported

An interesting case is when dealing with time-series of unusual shapes where the shape of the series is different from other large deviations.

- The goal is to determine *windows of data* (or *subsequences*) in which a given series behaves differently from a database of multiple sequences.
- Outliers correspond to *multiple consecutive timestamps,* either as:
  - a full-series anomaly
    - the shape of the entire series is treated as an anomaly
    - noise variations within the series usually mask the anomalous shape and thus, pruning techniques are used
  - a subsequence-based anomaly
    - anomalous shape is detected over small windows of the time-series.
  - There are a number of methods that can be used for this purpose, such as:

- transformation to other representations
    - a numeric multidimensional transformation, the series (or each subsequence of the series) is transformed into a multi-dimensional vector
    - by applying a discrete sequence transformation*,* the series can be transformed to *symbolic representations* by using discretisation methods such as the *Symbolic Aggregate Approximation* (SAX) [115][65]
        - piecewise aggregate approximations (PAA) are used in order to represent the time series either with a window-based averaging
            - the series is divided into windows of length whereas the average time-series value over each window is computed
    - by applying a value-based discretisation, an averaged time-series values, following a Gaussian distribution, are discretised into a smaller number of approximately equidepth intervals.
    - alternatives for *series compression are* the *Discrete Wavelet Transform* (DWT) and the *Fast Fourier Transform* [86]
        - wavelets are used to isolate characteristics of signals via a combined time–frequency representation
- distance-based
    - Proximity can be computed either with:
        - the Euclidean distance where the nearest neighbour distance
        - k-th nearest neighbour distance to a series can be used as the *anomaly score.*
        - *Dynamic Time Warping* and the *Hotsax* approach [50] are two examples of distance-based methods
            - In the latter, standard numeric Euclidean distances are used to define distance-based outliers while discrete approximations are used to perform *pruning.*
- methods for finding unusual shapes in multivariate series
    - in a multivariate time-series with *n* behavioural attributes which can be mapped to a (n + 1)-*dimensional trajectory,* the*TROAD* method [62] can be used in order to determine unusual shaped trajectories by exploring subsets of behavioural attributes in order to determine outliers.
- supervised methods for finding unusual time-series shapes
    - labels may either be associated with the entire time-series or portions of the time-series (i.e., subsequences)
    - The goal is to develop *subsequence profiles* for both the normal class and the anomalous class
    - Classification can be based on either a k-nearest neighbour approach [79][120] or by using Hidden Markov Models.

In cases with multidimensional data streams, as for example with backup backbone operational logs from a Sharepoint farm, outliers can be detected either by:

- the analysis of individual records
    - outliers are referred as novelties and can be detected either by using proximity-based algorithms such as the *STORM* [8] or the LOF algorithm [87], probabilistic algorithms such as

clustering or mixture models for mixed-data or algorithms for the high-dimensional scenario such as the *SPOT* algorithm [122] or general clustering algorithms.

  - The *Velocity Density Estimation* method [100] constructs a density-based velocity profile of the data where a data point X is an outlier only in the context of aggregate changes occurring in its locality, rather than its own properties as an outlier.

- changes in the aggregate trends of the multi-dimensional data
  - by following statistically significant changes in aggregate distributions, significant changes in time windows can be reported as the unusual changes in the data stream
- the detection of a rare and novel class
  - by following a combination of supervised and unsupervised methods used to detect outliers such as rare class, novel class, or infrequently recurring class outliers [123]

OUTLIER DETECTION IN DISCRETE SEQUENCES

In discrete sequences the values and the timestamps are categorical. Position or combination outliers can be detected by either deploying algorithms such as distance-based (using the k-nearest neighbour distance), frequency-based, or model-based (where a *probabilistic generative model* e.g. a Markov model or a Hidden Markov model, is constructed, which generates the subsequences or supervised learning, when there are training data of previous anomalies).

In cases dealing with the detection of *position outliers*, *rule-based* or *Markov models* can be used for predictive outlier detection [17]. With semi-supervised models, training data and test sequences are distinguished explicitly. In the general form of unsupervised learning, all positions which should be considered anomalies are determined [93]; an alternative would be the use of the short-memory property.

- The goal with rule-based models is to estimate the value of P from the training database of sequences D which can be expressed as a rule.
- Markov models are a special kind of *Finite State Automaton.* They represent the *sequence generation process* with the use of transitions in a *Markov chain* andy correspond to a set of states A, which represent the different kinds of memory about the system events.
- A Markov Model can be depicted as a set of *nodes* representing the *states* and a set of *edges* representing the *events*, which cause movement from one state to another. The *probability* of an edge provides the *conditional probability* of the corresponding event while the *order* of the model encodes the memory which the model retains for the generation process; *first-order models* correspond to the least amount of retained memory and *k-th order models* correspond to *rules*, whose antecedents are of length k. For a given test sequence, windows of size (k + 1) are extracted from the sequence. The first k symbols are used to determine the relevant state in the model while the probability that the (k + 1)-th event is the same as that which occurs in the test sequence. Variable-order and don’t care subsequences models are variations of order-k models. In the first case, a *state* in the model corresponds to different orders, depending upon its frequency in the data. *Higher order states* with very low frequency can be pruned from the model, and replaced with lower order generalizations using *Probabilistic Suffix Trees* (PST) [106][40]; a PST is a hierarchical data structure representing the different suffixes of a sequence where a suffix tree of depth at most k will store all the required conditional probability values for the k-th order Markov models, including the conditionals for all lower order Markov Models. In *don’t care subsequences models*, which are also referred as *Sparse Markov Transducers* (SMT), each *state* of the Markov Model represents a *subsequence* of length *k*. By allowing a *don’t care symbol* as a valid symbol in the subsequence, it significantly

generalises the state. In this way, the number of states are reduced while the number of training subsequences matching a state are increased. Markov chain Monte Carlo random sampling methods have been used widely for the analysis of datasets in high-dimensional problems.

Combination outliers can also be used effectively in intrusion detection scenarios. The goal is to determine unusual combinations of symbols in a given sequence, with respect to other sequences.

- In the unsupervised version, all anomalous sequences are determined within a database of sequences.
- In a semi-supervised analysis, an anomaly score is determined for a test sequence, with respect to a training database of normal sequences; the training database is more robust in the semi-supervised case as it does not contain any anomalies.
  - In the case of having a short test sequence and long training sequence, those are compared to determine rarity of the test sequence.
  - A short test sequence and short training sequence can be analysed using multi-dimensional methods for anomaly detection such as the k-nearest neighbour. Measures of similarity between the sequences are defined known as point-based techniques.
  - In the case of having long test sequence and long training sequence, windows of the test sequence are extracted as small sub-sequences, known as *comparison units.*
  - The relative behaviour of a comparison unit is compared in terms of the training data and the test sequence; if it prevails in the training sequence, then it is considered as an anomaly.
  - Similarities are quantified either as a distance value, a frequency of presence, or a generative probability of a Hidden Markov model.
  - For a given test sequence, the final anomaly score can be derived as a combination score from all the subsequences extracted from it. For greater generality, it is assumed that the training data contains multiple sequences.
  - in case of *web log*s of an installation on cloud or on premises, an additional step is required as to extract portions of the undifferentiated sequence as test sequences, using a *multi-granularity* approach
    - first test sequences of different lengths are extracted, usually in geometrically increasing sizes
    - smaller subsequences are extracted from the test sequences, which correspond to the comparison units and finally, the relative difference between the derived test sequences and the training sequences is calculated in terms of the smaller window-based comparison units.
- While dealing with cases which involve the detection of anomalies in login attempts, provided by AAA services, the main interest lies in the presence or absence of domain knowledge about relevant comparison units for anomaly detection using *supervision learning* where the models are evaluated and combined in terms of domain-dependent comparison units
  - An example is the *primitive model for combination outlier detection*, which is constructed on the basis of relative comparisons between the training sequences, the test sequence and the comparison units, usually extracted from the test sequence.
  - Variations of the latter include cases where there is a need for a post-processing step where the overall anomaly score of the test sequence is computed by combining the results

from different comparison units, or handling of special cases when the test sequence and training sequences are short and of comparable size.

- The goal in this case is to compute similarity measures between pairs of short sequences utilising possibly *point-anomaly* detection, by either using semi-supervised or unsupervised learning.

In *distance-based models,* the *absolute distance* of the comparison unit is computed to equivalent windows of the training sequence.

- The distance of the k-th nearest neighbour window in the training sequence is used in order to determine the anomaly score whereas higher values indicate greater proximity [36].

In case of *proximity-functions* which are similarity functions rather than distance functions, lower values indicate greater proximity. Several methods have been proposed to this purpose:

- a *simple matching coefficient* can be used to determine the number of matching positions between two sequences of equal length, which is equivalent to the *Hamming distance* between a pair of sequences.
- the *normalised longest common subsequence* is the sequential analogue of the *cosine distance* between two ordered set
- the *edit distance* measures the distance between two sequences by the minimum number of *edits* required to transform one sequence to the other
- Other alternatives methods are the *compression-based dissimilarity*, *counting mismatches among look-ahead pairs* and the *length-sensitive recursive computation of subsequence similarity*
- In cases where there is a need for combining anomaly scores from comparison units there are several methods that can be used such as the number of anomalous units, the aggregate anomaly score, the selective aggregate anomaly score, clustered anomaly scores, the locality frame count (LFC) [112] or the Leaky bucket [39].

In *frequency-based models*, the goal is to measure the relative frequency of the comparison unit in the training sequences

- An anomaly score is based on testing the frequency of the comparison unit in the training and test patterns or the comparison unit is extracted directly from the test sequence, or results from multiple comparison units can also be combined into a single anomaly score.

Hidden Markov Models (HMM) are probabilistic models designed for sequential data with temporal correlations that generate sequences through a sequence of transitions between states in a *Markov chain,* which corresponds to an observed data sequence [37][15][112].

- The *model behaviour* is always known completely.
- The *states* of the system are hidden and not directly visible to the user.
- Only a sequence of discrete observations are visible to the user, generated by *symbol emissions* from the states after each transition.
- Each state is associated with a *set of emission probabilities* over the symbol Σ while a *visit* to the state j leads to an emission of one of the symbols $\sigma_i \in \Sigma$ with probability $\theta^j(\sigma_i)$.

- In the *black-box model,* the number of states represents the *level of complexity* and it needs to be decided a-priori, how many states should be used. Thus, a large number of states can encode greater *complexity* about variations in sequence patterns while a small number of states may *overfit* a training data set. The *n* states will have n^2 possible transitions between different states, including self-transitions, n · |Σ| symbol generation probabilities with a distribution associated with each state and comparison units; not all pairs of states may have transitions among them while transitions do not occur between all pairs of states.
- The goal with HMM is to learn the *initial state probabilities*, *transition probabilities* and the *symbol emission probabilities* from the training database $\{T_1 \ldots T_N\}$.
- Comparison units are used for computing the anomaly scores.
  - In the *training phase,* model parameters are estimated, such as the initial probabilities, transition probabilities, and symbol emission probabilities with an Expectation- Maximization algorithm e.g. the *Baum-Welsch algorithm* also known as the *Forward-backward algorithm.*
  - In the *evaluation phase,* the probability that it fits the HMM is determined for defining the anomaly scores while a *recursive forward algorithm* is used to compute it, known as the *Forward algorithm*
  - Finally, in the e*xplanation phase,* the most likely sequence of states which generated this test sequence is determined for interpretability; when the states correspond to an *intuitive understanding* of the underling system, then, the *Viterbi* algorithm can be used.

In cases of detecting either position outliers or combination outliers, supervised methods can be used such feature transformations (k-grams, pattern-based methods), or distance-based (edit distance, combination of k-grams) utilising HMM.

- Labels may be available in different ways and may be associated with each position in a sequence, something that can be found in natural language data [7][58]. *Rare labels* may be associated with small subsequences of a *single large sequence.* Labels are associated with the full sequences in a database of multiple sequences where most labels are *normal* while a small number of them may be *anomalous* as it is the most common scenario in intrusion detection.
- With k-grams, symbolic sequences of size k are extracted from the base sequences while *words* are formed from the *sequence vocabulary,* used as the *features* in SVM classifiers.
- Pattern-based methods are based on the mining of patterns of sufficient *support* and *confidence* in terms of the underlying sequence symbols.
- A HMM creates a *generative model* specific to each class whereas its *parameters* are learned separately for each class using only the sequences specific to a particular class; for a given test sequence, the evaluation algorithm is used to determine the *identity* of the best fit class.

SPATIAL OUTLIER DETECTION

Spatial outlier detection can be deployed for traffic log analysis. Spatial outliers are objects having behavioural attribute values distinct from those of surrounding spatial neighbours. The issue of spatial continuity is an analogous principle to the concept of *temporal continuity,* where abrupt changes in the behavioural attribute, which violate spatial continuity provide useful information about the underlying contextual anomalies.

Spatial data comprise of two types of attributes, the *behavioural* and the *contextual* attributes.

- The former are measured for each object and it is possible to have more than one behavioural attribute at a spatial location in a given application
- Contextual attributes define the *location of interest* at which the behavioural attribute is measured. Typically, this would contain two or three dimensions, when data are expressed in terms of *coordinates* or may be expressed at the *granularity* of a region of interest, such as a county, zip-code, or may correspond to individual pixels in an imaging application. Behavioural attribute values in spatial neighbourhoods are closely correlated with one another while their variances depend on spatial location.
    - In *spatiotemporal* data, as are those found in man cases in operational logs derived from a MFT system or a collaboration platform, contextual attributes may also contain a temporal component that can be used to determine important spatiotemporal anomalies (or events) based on the underlying dynamics. In the case of purely spatiotemporal data, *trajectories of objects* are measured over time, where either there is a lack of any behavioural attributes or, *temporal components* are treated as the contextual attribute and *spatial components* as the behavioural attributes.

Outlier detection analysis with spatial data can be deployed using either neighbour-based algorithms, auto-regressive models, visualisation with variogram clouds, algorithms for finding abnormal shapes, spatiotemporal outlier detection or supervised models

- In the case of using neighbourhood-based algorithms [54][68][57], *abrupt changes* in the spatial neighbourhood of a data point are used in order to diagnose outliers. Neighbourhoods are defined on the basis of multidimensional distances between data points while outliers are defined in a local basis.
- In *graph-based* neighbourhoods, these are defined by linkage relationships between spatial objects, while outlier detection can be made using either graph-based or *spatial proximity methods*, modelled with the use of links between nodes.
- In the case of handling of multiple behavioural attributes, deviations may be computed on each behavioural attribute using either the Mahalanobis distance-based method for extreme value analysis [112]*,* depth-based methods or angle-based methods; then these values are combined into a *single deviation value*, which provides the final outlier score*.*
- In *autoregressive models, s*patial data share a number of similarities with temporal data. They both measure a *behavioural attribute* (e.g. temperature) with respect to a *contextual attribute* (e.g. space or time). Extreme value analysis techniques can be used in order to determine any deviations which vary significantly from the norm; these values are assumed to be independent identically distributed random variables, which are drawn from a normal distribution. Regression models (ARMA, ARIMA, PCA) can be generalised to the spatial scenario by using the appropriate slice of values from the spatial data.
- Examples of *visualisation techniques* for spatial outliers are *pocket plots* and *variogram clouds* [41][114]. The latter are materialised by creating a *scatter plot* between the *spatial distances on the X-axis*, and the *behavioural square deviations on the Y-axis*, for every pair of points in the data set. The *spatial distance* is given by the *euclidian distance* between a pair of points. The *behavioural attribute deviation* is defined as the half the *square distance* between the behavioural attribute values. Smaller spatial distances will likely correspond to smaller behavioural attribute variances while large variations of the behavioural attribute for smaller spatial distances should be considered *deviants.*

OUTLIER DETECTION IN GRAPHS AND NETWORKS

In real domains, data may contain many small graphs, drawn over a small base domain of labeled nodes. Individual graph objects are defined as outliers based on the model of normal graph objects in the database.

- Outlier analysis detection can be applied using *proximity-based* models where each graph object is treated as an individual point.
- The goal is to determine the frequent subgraphs in the underlying data and then determine those graphs which do not contain these frequent pattern or make use of a k-nearest neighbour algorithm.

In cases dealing with data exchange in a collaboration platform or an MFTs, data may be represented as a single large graph.

- Nodes may correspond to distinct identifiers such as URLs, actors, or IP addresses.
- In temporal graphs, the structure of the network may change over time
  - outliers may correspond to significant changes in specific structural aspects of the network, such as communities, shortest paths, or other local structural properties.
- An *outlier node* could be a node with unusual high degree in connectivity structure, changing degree, changing community structure, changing distances to other nodes or relationships of node content to linkage structure.

There are different kind of outliers in a single large graph such as *node outliers*, *linkage outliers* and *subgraph outliers*.

- In the first case, the goal is to extract features from the neighbourhood of a node and then define the outliers in terms of these features [5].
- With linkage outliers the goal is to find *edges* which lie across dense partitions in the network by using either community linkage outlier detection, matrix factorisation or spectral methods [48];
  - if the nodes in the graph are clustered, then the edges across these node clusters are defined as outliers.
- The *Minimum Description Length* (MDL) principle or the *SUBDUE* system for subgraph pattern mining can be used to extract parts of the graph which exhibit unusual behaviour with respect to the normal patterns in the full graph in order to define subgraph outliers [81].
- Noisy outlier linkages can degrade the quality of a variety of network mining algorithms, known also as *network de-noising* [38][88].
- Truly anomalous linkages can provide valuable information for unusual connections among nodes. Such structural pattern changes refer to either linkage anomalies, which can be dealt with by using a cluster partitioning for computing the likelihood fits, pattern-based evolution, community evolution or distance evolution [10], whereas sudden and abrupt changes in pairwise distances between nodes are indicative of unusual events in a network.

Commercial and open source solutions based on machine learning for threat detection

In traffic log analysis with data representing a sequence of accesses in an *operational log*, the goal is to determine the unusual patterns of accesses in this log. Operational logs are typically pre-processed into a set of user-specific discrete sequences which correspond to the identifiers of e.g. the pages or services accessed by the users. Users can often be distinguished only at the level of their IP-addresses while user sessions can often be mined from the logs; the log needs to be decomposed into user sessions and then, it is further decomposed into test sequences, and comparison units [23].

- In order to detect *position outliers,* the goal is to determine any *unexpected accesses.* In the case of contextual anomalies corresponding to a *single unpredictable access,* this can be accomplished by using either Markov or rule-based models.
- When the aim is to detect combination outliers, the goal is to determine *unusual subsequences* in the test sequence, by using either unsupervised methods such as window-based nearest neighbour models or Hidden Markov Models, or supervised methods such as the extraction of relevant features (e.g. k-grams) [29][55].

In *host-based intrusion* of systems [28] of the underlying infrastructure of a collaboration system on cloud or on premises, operating system call traces are available in the form of symbolic sequences; any anomalous subsequences in these traces correspond to malicious computer programs. In this case, *data* are similar to *web logs* at a conceptual level.

- *Calls*, form the base alphabet Σ over which the mining is performed at the *user command level or* at the *operating system level.* Different kinds of programs execute different sequential combinations of calls. This *sequential ordering of the calls* provides critical information in order to distinguish between normal and malicious programs.
- In the *feature extraction* phase, logs are transformed into symbolic sequences. In the case where commands are coming from multiple sources, they are separated out into their different hostnames in order to examine the malicious behaviour of a particular host, usually by applying discrete sequence methods [25][112][36][39][60].

Having a stream of network packets or data records containing both continuous and categorical attributes, the goal is to determine network intrusions. *Temporal relationships* between data records are much weaker than in the case of host-based systems.

- Each individual record is multidimensional containing features extracted from the unit of network data (e.g., packet), or raw tcpdump data.
- Unsupervised learning algorithms will aim to detect aggregate change point in order to identify *network-wide traffic anomalies* corresponding to network intrusion and network attacks.
- Streaming and supervised novel-class detection methods will aim to identify *repeating attacks* in order to detect *novel classes* as new intrusions arise [59][69][70][105][107].

In the case of *event detection* in text files in order to detect e.g. data exfiltration of sensitive corporation data, having a document corpus D, the goal is to determine unusual documents, which differ significantly from the trend.

- In *first story* detection where a stream of documents is available, the goal is to determine unusual events corresponding to new topics in the stream of documents. User activities, such as *tweets*, may provide early knowledge of unusual events as *changes* in topical and linkage distributions; both supervised and unsupervised methods can be used, upon the availability of training data [3][49][85][111][119][123].

Having a stream of emails, the goal is determine the subset of emails which correspond to *spam*.

- Unsupervised methods can be used for unusual topic detection although results are often likely to be inaccurate.
- Working with supervised methods, specific features of the emails are learned which are then related to spam labels in the training data.
- The Bayes classifier or email spam filtering methods can be applied for text classification by taking into account specific email characteristics of a *junk email* such as the domain of the sender, any peculiar phrases or overemphasised punctuation or whether the recipient of the message was a particular user, or a mailing list.

In the underlying network of a collaboration platform or in Managed File Transfer, with content at the nodes, the goal of outlier detection analysis is to determine the noisy and spam links with the use of structure and content information as linkage outliers.

- in an *evolving network* with associated text content at the nodes, the goal is to determine the anomalous regions of activity or change in the network, a problem related to community evolution in the underlying network by using either purely structural approaches, community detection algorithms, spectral methods or eigen-space analysis, to discover any threat activities.
- Autoregressive models can be used for the removal of *noisy links*, while PCA can be used for *noise correction* or to improve the *representation quality* of data sets for mining and retrieval.

Complex event processing methods utilising machine learning algorithms have been used recently in security management for the analysis of security-related events.

- A Bayesian network and a Monte Carlo sampling heuristic algorithm have been used in [113].
- An approach to identify suspicious, unknown event patterns in the cloud, based on real-time Discriminant Analysis, is presented in [117].
- A method to deal with massive audio data streams, evaluated against a Gaussian mixture modelling, is given in [1].
- The use of temporal model in CEP is discussed in [116] while an approach for the detection of event processing patterns in event databases is presented in [45].
- In [63], an efficient machine learning approach is introduced in order to evaluate the security level of a masked implementation of AES in the case of side-channel attacks.
- The authors in [12] propose the use of Dynamic Data-Driven Application Systems (DDDAS), based on Bayes rule, for trust in cyber-security. The *Address Space Layout Randomisation* (ASLR) technique can be used to protect against buffer overflow attacks. In [66], there are several unsupervised multidimensional outlier detection methods that can be generalised for detecting network-based intrusions.
- An anomaly detection method based on *time-series analysis* for network security is presented in [98] using an inter-transactional association rule mining method engineering *Layer Divided Modelling* (LDM) for temporal pattern analysis; performance evaluation is based on *10-fold cross-validation.* The proposed network-IDS uses a supervised classification based on a *Col-*

lateral Representative Subspace Projection Modelling (C-RSPM) utilising data mining and detection of sequential intrusion patterns, based on the rule set trained/derived from the attack history.

- An unsupervised approach to identify threats using graph-based anomaly detection, utilising the *SUBDUE* compression, is presented in [32].
- A weighted approach based on a threshold can be found in [9] for the detection of insider threats in databases in the form of SQL injections.
- A proactive monitoring approach for assessing reliability of SOA-based systems (middleware-based), using AI-reasoning on dynamically collected failure data of each service and its components along with results from random testing is presented in [18]
- component system reliability is estimated using *Markov models*, *Bayesian Reasoning and component dependency graphs* (directed graphs).
- An approach for risk assessment with real-time constraint based on attack graphs using a feed-forward backward propagating neural network is presented in [76].
- CNN has been found that withstand the *brute-force* attack using *information entropy* for prediction and statistical analysis for evaluation, based on histogram analysis and correlation of adjacent pixels. An image encryption approach based on the use of *Chaotic Cellular Neural Network* is proposed in [84].
- A data mining technique using the *Local Outlier Factor* (LOF), *Shared Nearest Neighbour clustering* (SNN) and *k-Nearest Neighbour classification* (kNN) is proposed in [95], based on *secure multi-party computation* techniques to compute the nearest neighbours of points in horizontally distributed datasets.
- A non-intrusive approach to enhance legacy-embedded control systems with cyber protection features in [90], using a *Finite State Machine* evaluator and the *Mealy machine model* for modelling system behaviours.
- A context-model to support the formal definition and acquisition of context descriptions based on primary context types such as the identity, location, activity and time, is presented in [105] based on the spatiotemporal model by using methods such as *(k,e)-anonymity, t-closeness, variance control* and *(e,m)-anonymity,* for location privacy.

There are several state-of-the-art current European projects which use machine learning for security.

- The MASIIF's predictive security analyser can process a behaviour analysis to to detect misuse patterns [73].
- The *NECOMA* project [80] s an anomaly detection approach in backbone networks using Big Data analytics.
- The VIS-SENSE project [110] deals with the identification and prediction of complex patterns of abnormal behaviour in network security domain.

Outlier detection analysis has been proposed for the detection and the protection against attacks on machine learning algorithms.

- The RONI defence strategy [9], uses a *SpamBayes spam filter* that produces a classifier, which based on a set of spam and non-spam messages classifies new messages, either as ham (non-spam), spam or unsure. The authors argue that defences against causative targeted should be based on regularisation and randomisation techniques, while causative indiscriminate attacks can be mitigated using regularisation tools

- exploratory targeted attacks can be handled by deploying information hiding and randomisation techniques while in case of exploratory indiscriminate attacks countermeasures should be based on information hiding. They present a *Hypersphere Outlier Detection* method, based on a *Simple Outlier Detection Model*, that examines a causative attack to manipulate a naive learning algorithm.
- The use of spatiotemporal outlier analysis is proposed in [91] an effective defence is to detect attacks and their sources by performing a global analysis of SPAM and malicious traffic in order to generate classifiers to identify spam, resilient to reverse engineering.
- A survey on unsupervised outlier detection in high-dimensional numerical data in Euclidean space is presented in [124]
- Finally, a novel statistical technique to automatically detect long-term anomalies in cloud time-series data, including application and system metrics which build on Extreme Studentized Deviate (ESD) test is explained in [108].

Conclusions and further work

An analysis of the state-of-the-art methodologies on the literature as well as best practices in commercial products and open source solutions in the field of auto-protected systems in terms of threat and attack detection by deploying machine learning algorithms has been presented in this report.

Among other techniques, outlier detection analysis is very promising on detecting threats and attacks in near-real time scenarios, such as APTs, DDoS and zero-day attacks in collaboration platforms and MFTs by identifying behaviours as outliers that deviate from normal behaviour.

In the next stage, security assessment of the basic building blocks will take place. Static and/or dynamic tools will be used for project procurement for vulnerability scanning using either commercial or preferably open source solutions such as the OpenVas framework.

Real operational logs taken from collaboration platforms and MFTs installed on premises or in the cloud such as MS Sharepoint, OwnCloud or other solutions will be used for Big Data analytics, utilising outlier detection analysis algorithms. Spark's MLib functionality with an API implementation in Scala, along with the adaptation of algorithms provided by the ELKI platform written in Java as well as the redefinement of other ML algorithms, especially in the case of unsupervised learning, as well as the potential use of Apache Flume or Kafka for data aggregation will be in line with research undertaken on fraud detection in the Brufence project.

References

1. AGGARWAL, Charu C. On classification and segmentation of massive audio data streams. Knowledge and information systems, 2009, 20.2: 137-156
2. AGGARWAL, Charu C.; REDDY, Chandan K. (ed.). Data clustering: algorithms and applications. CRC Press, 2013
3. AGGARWAL, Charu C.; SUBBIAN, Karthik. Event Detection in Social Streams. In: SDM. 2012. p. 624-635
4. AKOGLU, Leman, et al. Fast and reliable anomaly detection in categorical data. In: Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012. p. 415-424
5. AKOGLU, Leman; MCGLOHON, Mary; FALOUTSOS, Christos. Oddball: Spotting anomalies in weighted graphs. In: Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, 2010. p. 410-421
6. AL-KHATEEB, Tahseen, et al. Stream classification with recurring and novel class detection using class-based ensemble. In: Data Mining (ICDM), 2012 IEEE 12th International Conference on. IEEE, 2012. p. 31-40
7. ALTUN, Yasemin, et al. Hidden markov support vector machines. In: ICML. 2003. p. 3-10.
8. ANGIULLI, Fabrizio; FASSETTI, Fabio. Detecting distance-based outliers in streams of data. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, 2007. p. 811-820.
9. BARRENO, Marco, et al. Can machine learning be secure?. In: Proceedings of the 2006 ACM Symposium on Information, computer and communications security. ACM, 2006. p. 16-25
10. BERLINGERIO, Michele, et al. Mining graph evolution rules. In: Machine learning and knowledge discovery in databases. Springer Berlin Heidelberg, 2009. p. 115-130.
11. BEYER, Kevin, et al. When is “nearest neighbor” meaningful?. In: Database Theory—ICDT’99. Springer Berlin Heidelberg, 1999. p. 217-235.
12. BLASCH, Erik; AL-NASHIF, Youssif; HARIRI, Salim. Static Versus Dynamic Data Information Fusion Analysis Using DDDAS for Cyber Security Trust. Procedia Computer Science, 2014, 29: 1299-1313
13. BLEI, David M.; LAFFERTY, John D. Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning. ACM, 2006. p. 113-120
14. BREUNIG, Markus M., et al. LOF: identifying density-based local outliers. In: ACM sigmod record. ACM, 2000. p. 93-104
15. CADEZ, Igor, et al. Visualization of navigation patterns on a web site using model-based clustering. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2000. p. 280-284
16. CAMPBELL, C.; BENNETT, K. P. A linear-programming approach to novel class detection. In: NIPS Conference. 2000
17. CHAKRABARTI, Soumen; SARAWAGI, Sunita; DOM, Byron. Mining surprising patterns using temporal description length. In: VLDB. 1998. p. 606-617
18. CHALLAGULLA, Venkata UB; BASTANI, Farokh B.; YEN, I.-Ling. High-confidence compositional reliability assessment of SOA-based systems using machine learning techniques. In: Machine Learning in Cyber Trust. Springer US, 2009. p. 279-322
19. CHAN, Philip K.; STOLFO, Salvatore J. Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In: KDD. 1998. p. 164-168
20. CHAWLA, Nitesh V.; JAPKOWICZ, Nathalie; KOTCZ, Aleksander. Editorial: special issue on learning from imbalanced data sets. ACM Sigkdd Explorations Newsletter, 2004, 6.1: 1-6
21. CHEN, Dechang, et al. On detecting spatial outliers. Geoinformatica, 2008, 12.4: 455-475. [112]
22. CHENG, Haibin, et al. Detection and Characterization of Anomalies in Multivariate Time Series. In: SDM. 2009. p. 413-424

23. COOLEY, Robert; MOBASHER, Bamshad; SRIVASTAVA, Jaideep. Data preparation for mining world wide web browsing patterns. Knowledge and information systems, 1999, 1.1: 5-32

24. DAS, Gautam; MANNILA, Heikki. Context-based similarity measures for categorical databases. In: Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg, 2000. p. 201-210

25. DASGUPTA, Dipankar; NINO, Fernando. A comparison of negative and positive selection algorithms in novel pattern detection. In: Systems, Man, and Cybernetics, 2000 IEEE International Conference on. IEEE, 2000. p. 125-130

26. DEERWESTER, Scott C.. , et al. Indexing by latent semantic analysis. JAsIs, 1990, 41.6: 391-407

27. DEMPSTER, Arthur P.; LAIRD, Nan M.; RUBIN, Donald B. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), 1977, 1-38

28. DENNING, Dorothy E. An intrusion-detection model. Software Engineering, IEEE Transactions on, 1987, 2: 222-232

29. DESHPANDE, Mukund; KARYPIS, George. Selective Markov models for predicting Web page accesses. ACM Transactions on Internet Technology (TOIT), 2004, 4.2: 163-184

30. DOMINGOS, Pedro. Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999. p. 155-164

31. DRUMMOND, C.; HOLTE, R. Class imbalance, and cost sensitivity: Why under-sampling beats oversampling. In: Workshop on International Conference on Machine Learning (ICML'2003). 2003

32. EBERLE, William; Holder, Lawrence; Cook, Diane. Identifying threats using graph-based anomaly detection. In: Machine Learning in Cyber Trust. Springer US, 2009. p. 73-108

33. ELKAN, Charles; NOTO, Keith. Learning classifiers from only positive and unlabeled data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008. p. 213-220

34. FAN, Wei, et al. AdaCost: misclassification cost-sensitive boosting. In: ICML. 1999. p. 97-105

35. FAWCETT, Tom; PROVOST, Foster. Activity monitoring: Noticing interesting changes in behavior. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999. p. 53-62

36. FORREST, Stephanie, et al. A sense of self for unix processes. In: Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on. IEEE, 1996. p. 120-128

37. GAO, Bo; MA, Hui Ye; YANG, Yu Hang. Hmms (hidden markov models) based on anomaly intrusion detection method. In: Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on. IEEE, 2002. p. 381-385

38. GAO, Huiji, et al. Network denoising in social media. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ACM, 2013. p. 564-571

39. GHOSH, Anup K.; SCHWARTZBARD, Aaron; SCHATZ, Michael. Learning Program Behavior Profiles for Intrusion Detection. In: Workshop on Intrusion Detection and Network Monitoring. 1999

40. GUSFIELD, Dan. Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge university press, 1997

41. HASLETT, John, et al. Dynamic graphics for exploring spatial data with application to locating global and local anomalies. The American Statistician, 1991, 45.3: 234-242

42. HE, Zengyou, et al. Fp-outlier: Frequent pattern based outlier detection. Computer Science and Information Systems, 2005, 2.1: 103-118

43. HIDO, Shohei, et al. Statistical outlier detection using direct density ratio estimation. Knowledge and information systems, 2011, 26.2: 309-336

44. HOFFMANN, Heiko. Kernel PCA for novelty detection. Pattern Recognition, 2007, 40.3: 863-874

45. J. Mihaeli and O. Etzion. Detecting event processing patterns in event databases. 2007. URL: http://www.dcs.bbk.ac.uk/ptw/vldbw07/edaps/mihaeli.pdf

46. JAGADISH, H. V.; KOUDAS, Nick; MUTHUKRISHNAN, S. Mining Deviants in a Time Series Database. In: VLDB. 1999. p. 7-10

47. JOHN HENRY HOLLAND. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992

48. KARGER, David R. Random sampling in cut, flow, and network design problems. Mathematics of Operations Research, 1999, 24.2: 383-413

49. KASIVISWANATHAN, Shiva Prasad, et al. Emerging topic detection using dictionary learning. In: Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, 2011. p. 745-754

50. KEOGH, Eamonn; LIN, Jessica; FU, Ada. Hot sax: Efficiently finding the most unusual time series subsequence. In: Data mining, fifth IEEE international conference on. IEEE, 2005. p. 8 pp

51. KIFER, Daniel; BEN-DAVID, Shai; GEHRKE, Johannes. Detecting change in data streams. In: Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004. p. 180-191

52. KNORR, Edwin M.; NG, Raymond T. Finding intensional knowledge of distance-based outliers. In: VLDB. 1999. p. 211-222

53. KNOX, Edwin M.; NG, Raymond T. Algorithms for mining distancebased outliers in large datasets. In: Proceedings of the International Conference on Very Large Data Bases. 1998. p. 392-403

54. KOU, Yufeng; LU, Chang-Tien; CHEN, Dechang. Spatial Weighted Outlier Detection. In: SDM. 2006. p. 614-618

55. KRUEGEL, Christopher; VIGNA, Giovanni. Anomaly detection of web-based attacks. In: Proceedings of the 10th ACM conference on Computer and communications security. ACM, 2003. p. 251-261

56. KUBAT, Miroslav, et al. Addressing the curse of imbalanced training sets: one-sided selection. In: ICML. 1997. p. 179-186

57. KUT, Alp; BIRANT, Derya. Spatio-temporal outlier detection in large databases. CIT. Journal of computing and information technology, 2006, 14.4: 291-297

58. LAFFERTY, John; MCCALLUM, Andrew; PEREIRA, Fernando CN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001

59. LAKHINA, Anukool; CROVELLA, Mark; DIOT, Christophe. Mining anomalies using traffic feature distributions. In: ACM SIGCOMM Computer Communication Review. ACM, 2005. p. 217-228

60. LANE, Terran; BRODLEY, Carla E. Temporal sequence learning and data reduction for anomaly detection. ACM Transactions on Information and System Security (TISSEC), 1999, 2.3: 295-331

61. LAZAREVIC, Aleksandar; KUMAR, Vipin. Feature bagging for outlier detection. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. ACM, 2005. p. 157-166

62. LEE, Jae-Gil; HAN, Jiawei; LI, Xiaolei. Trajectory outlier detection: A partition-and-detect framework. In: Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. IEEE, 2008. p. 140-149

63. LERMAN, Liran, et al. A machine learning approach against a masked AES. In: Smart Card Research and Advanced Applications. Springer International Publishing, 2014. p. 61-75

64. LI, Xiao-Li; LIU, Bing; NG, See-Kiong. Negative training data can be harmful to text classification. In: Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2010. p. 218-228

65. LIN, Jessica, et al. A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM, 2003. p. 2-11

66. Lior, Rocach, Ben Gurion University of the Negev, When cyber-security meets ML , a presentation at Penn State University, http://fr.slideshare.net/liorrokach/cyber-securityshort

67. LIU, Xuan; ZHANG, Pengzhu; ZENG, Dajun. Sequence matching for suspicious activity detection in anti-money laundering. In: Intelligence and Security Informatics. Springer Berlin Heidelberg, 2008. p. 50-61

68. LU, Chang-Tien; CHEN, Dechang; KOU, Yufeng. Algorithms for spatial outlier detection. In: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. IEEE, 2003. p. 597-600

69. LUNG-YUT-FONG, Alexandre; LÉVY-LEDUC, Céline; CAPPÉ, Olivier. Distributed detection/localization of change-points in high-dimensional network traffic data. Statistics and Computing, 2012, 22.2: 485-496

70. MAHONEY, Matthew V.; CHAN, Philip K. Learning rules for anomaly detection of hostile network traffic. 2003

71. MARKOU, Markos; SINGH, Sameer. Novelty detection: a review—part 1: statistical approaches. Signal processing, 2003, 83.12: 2481-2497

72. MARKOU, Markos; SINGH, Sameer. Novelty detection: a review—part 2:: neural network based approaches. Signal processing, 2003, 83.12: 2499-2521

73. MASIIF: Management of Security Information and events in Service Infrastructures, http://www.massif-project.eu

74. MASUD, Mohammad M., et al. Classification and adaptive novel class detection of feature-evolving data streams. Knowledge and Data Engineering, IEEE Transactions on, 2013, 25.7: 1484-1497

75. MASUD, Mohammad M., et al. Detecting recurring and novel classes in concept-drifting data streams. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE, 2011. p. 1176-1181

76. Mihai-Gabriel, Ionita, and Patriciu Victor-Valeriu. "Achieving DDoS resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory." Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on. IEEE, 2014

77. MILLER, Benjamin, et al. Eigenspace analysis for threat detection in social networks. In: Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on. IEEE, 2011. p. 1-7.

78. MONGIOVÌ, Misael, et al. SigSpot: mining significant anomalous regions from time-evolving networks. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, 2012. p. 865-865

79. MUEEN, Abdullah; KEOGH, Eamonn; YOUNG, Neal. Logical-shapelets: an expressive primitive for time series classification. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011. p. 1154-1162

80. NECOMA: Nippon-European Cyberdefence-Oriented Multilayer threat Analysis, http://www.necoma-project.eu

81. NOBLE, Caleb C.; COOK, Diane J. Graph-based anomaly detection. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003. p. 631-636

82. PAPADIMITRIOU, Christos H., et al. Latent semantic indexing: A probabilistic analysis. In: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. ACM, 1998. p. 159-168

83. PELLEG, Dan; MOORE, Andrew W. Active learning for anomaly and rare-category detection. In: Advances in Neural Information Processing Systems. 2004. p. 1073-1080

84. PENG, Jun; ZHANG, Du. Image encryption and chaotic cellular neural network. In: Machine Learning in Cyber Trust. Springer US, 2009. p. 183-213

85. PETROVIĆ, Saša; OSBORNE, Miles; LAVRENKO, Victor. Streaming first story detection with application to twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010. p. 181-189

86. PINSKY, Mark A. Introduction to Fourier analysis and wavelets. Pacific Grove: Brooks/Cole, 2002

87. POKRAJAC, Dragoljub; LAZAREVIC, Aleksandar; LATECKI, Longin Jan. Incremental local outlier detection for data streams. In: Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on. IEEE, 2007. p. 504-515

88. QI, Guo-Jun; AGGARWAL, Charu C.; HUANG, Thomas S. On clustering heterogeneous social media objects with outlier links. In: Proceedings of the fifth ACM international conference on Web search and data mining. ACM, 2012. p. 553-562

89. RAMASWAMY, Sridhar; RASTOGI, Rajeev; SHIM, Kyuseok. Efficient algorithms for mining outliers from large data sets. In: ACM SIGMOD Record. ACM, 2000. p. 427-438

90. REN, Shangping, et al. A non-intrusive approach to enhance legacy embedded control systems with cyber protection features. In: Machine Learning in Cyber Trust. Springer US, 2009. p. 155-181
91. Richard, Lippman, MIT Lincoln Laboratory, a Jones seminar presentation at Dartmouth College, http://engineering.dartmouth.edu/events/using-machine-learning-to-improve-security-in-adversarial-environments/, 14/01/2011
92. ROBERTS, Stephen J. Novelty detection using extreme value statistics. IEE Proceedings-Vision, Image and Signal Processing, 1999, 146.3: 124-129
93. RON, Dana; SINGER, Yoram; TISHBY, Naftali. The power of amnesia: Learning probabilistic automata with variable memory length. Machine learning, 1996, 25.2-3: 117-149
94. SCHAPIRE, Robert E.; SINGER, Yoram. Improved boosting algorithms using confidence-rated predictions. Machine learning, 1999, 37.3: 297-336
95. SCHÖLKOPF, Bernhard, et al. Support Vector Method for Novelty Detection. In: NIPS. 1999. p. 582-588
96. SHANECK, Mark; KIM, Yongdae; KUMAR, Vipin. Privacy preserving nearest neighbor search. In: Machine Learning in Cyber Trust. Springer US, 2009. p. 247-276
97. SHEKHAR, Shashi; LU, Chang-Tien; ZHANG, Pusheng. A unified approach to detecting spatial outliers. GeoInformatica, 2003, 7.2: 139-166
98. SHYU, Mei-Ling; HUANG, Zifang; LUO, Hongli. Efficient mining and detection of sequential intrusion patterns for network intrusion detection systems. In: Machine Learning in Cyber Trust. Springer US, 2009. p. 133-154
99. SIEBES, Arno; VREEKEN, Jilles; VAN LEEUWEN, Matthijs. Item Sets that Compress. In: SDM. 2006. p. 393-404
100. SILVERMAN, Bernard W. Density estimation for statistics and data analysis. CRC press, 1986
101. SMETS, Koen; VREEKEN, Jilles. The Odd One Out: Identifying and Characterising Anomalies. In: SDM. 2011. p. 109-148
102. SMYTH, Padhraic, et al. Clustering sequences with hidden Markov models. Advances in neural information processing systems, 1997, 648-654
103. SONG, Xiuyao, et al. Conditional anomaly detection. Knowledge and Data Engineering, IEEE Transactions on, 2007, 19.5: 631-645
104. SRIVASTAVA, Ashok N. Enabling the discovery of recurring anomalies in aerospace problem reports using high-dimensional clustering techniques. In: Aerospace Conference, 2006 IEEE. IEEE, 2006. p. 17 pp
105. Stephen J. H. Yang; Jia Zhang; Angus F. M. Huang. Model, properties and applications of context-aware web-services. In: Machine Learning in Cyber Trust. Springer US, 2009, p. 323-358
106. SUN, Pei; CHAWLA, Sanjay; ARUNASALAM, Bavani. Mining for Outliers in Sequential Databases. In: SDM. 2006. p. 94-105
107. THOTTAN, Marina; JI, Chuanyi. Anomaly detection in IP networks. Signal Processing, IEEE Transactions on, 2003, 51.8: 2191-2204
108. VALLIS, Owen; HOCHENBAUM, Jordan; KEJARIWAL, Arun. A novel technique for long-term anomaly detection in the cloud. In: Proceedings of the 6th USENIX conference on Hot Topics in Cloud Computing. USENIX Association, 2014. p. 15-15
109. VASCO, https://www.vasco.com
110. VIS-SENSE: Visual Analytic Representations of Large Datasets for Enhancing Network Security, http://www.vis-sense.eu
111. WANG, Xuanhui, et al. Mining correlated bursty topic patterns from coordinated text streams. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007. p. 784-793
112. WARRENDER, Christina; FORREST, Stephanie; PEARLMUTTER, Barak. Detecting intrusions using system calls: Alternative data models. In: Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on. IEEE, 1999. p. 133-145

113. WASSERKRUG, Segev, et al. Complex event processing over uncertain data. In: Proceedings of the second international conference on Distributed event-based systems. ACM, 2008. p. 253-264

114. WEBSTER, R.; OLIVER, M. A. Software for spatial data analysis in 2D. European Journal of Soil Science, 1997, 48.1: 173-175

115. WEI, Li; KEOGH, Eamonn; XI, Xiaopeng. SAXually explicit images: finding unusual shapes. In: Data Mining, 2006. ICDM'06. Sixth International Conference on. IEEE, 2006. p. 711-720

116. WHITE, Walker, et al. What is next in event processing?. In: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2007. p. 263-272

117. WIDDER, Alexander, et al. Identification of suspicious, unknown event patterns in an event cloud. In: Proceedings of the 2007 inaugural international conference on Distributed event-based systems. ACM, 2007. p. 164-170

118. YAMINSHI, K.; TAKEUCHI, J. A Unified Framework for Detecting Outliers and Change Points from Time Series Data. In: ACM KDD Conference. 2002

119. YANG, Yiming, et al. Topic-conditioned novelty detection. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002. p. 688-693

120. YE, Lexiang; KEOGH, Eamonn. Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009. p. 947-956

121. YU, Hwanjo; HAN, Jiawei; CHANG, Kevin Chen-Chuan. PEBL: Web page classification without negative examples. Knowledge and Data Engineering, IEEE Transactions on, 2004, 16.1: 70-81

122. ZHANG, Ji; GAO, Qigang; WANG, Hai. Spot: A system for detecting projected outliers from high-dimensional data streams. In: Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. IEEE, 2008. p. 1628-1631

123. ZHANG, Jian; GHAHRAMANI, Zoubin; YANG, Yiming. A probabilistic model for online document clustering with application to novelty detection. In: Advances in Neural Information Processing Systems. 2004. p. 1617-1624

124. ZIMEK, Arthur; SCHUBERT, Erich; KRIEGEL, Hans-Peter. A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal, 2012, 5.5: 363-387