UDC 004.67

Framework for detecting outlier and database intrusions

Mykhailo V. Kolomytsev[1], Svitlana O.Nosok[1]

[1]*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Abstract

This paper presents a methodology and framework for detecting anomalies in the actions of relational database users, with a focus on insider threats. The architecture of the framework is described, including the choice of parameters for logging user behavior and the justification of the anomaly detection algorithm. An overview of the existing anomaly-detection solutions is provided. The proposed methodology for the functioning of the framework is outlined with recommendations on the choice of algorithm parameters. The analysis of insider actions in databases provides an original approach to anomaly detection and contributes to the field of information security.

Keywords: Anomaly detection, insiders, machine learning algorithm

Introduction

Theft or modification of information in a database by insiders can lead to unusual activities in the user activity time series. For example, a sudden increase in activity by a particular user at a time when no activity would normally occur could indicate an unusual activity. Changes in the frequency and volume of access to specific databases, which may indicate information theft, can also be detected. Such changes appear as anomalous values, that is, data points that deviate significantly from the typical distribution of a dataset. Time-series analysis can help to identify such anomalies and prevent confidential information leaks and data breaches.

Finding anomalous values in a time series can be challenging because time-series data have unique characteristics that differ from those of stochastic datasets. Some of the time series features to consider when searching for anomalies are:

Time-series data can show trends; that represent long-term changes in the data. For example, an employee's skill improvement may increase the number of database queries per unit of time.

Many time-series data exhibit seasonality, which is a periodic pattern. A typical example is performing all types of routine operations, such as reporting.

Time-series data can exhibit autocorrelation, which means that the current value of the data correlates with its past values. Typically, the degree of autocorrelation is high and positive; the user performs similar activities daily as part of their functional responsibilities. This makes it difficult to distinguish between normal and true data fluctuations.

Time-series data may be nonstationary, meaning that the statistical properties of the data may change over time. This makes it difficult to use traditional statistical methods to detect anomalies.

To prevent hidden actions by insiders in the database, it is necessary to implement an access control and monitoring system that can detect unusual user activities. The authors proposed a framework for detecting the abnormal behavior of database users based on machine learning.

Theft or modification of information in a database by insiders can lead to unusual activities in the user activity time series. For example, a sudden increase in activity by a particular user at a time when no activity would normally occur could indicate an unusual activity. Changes in the frequency and volume of access to specific databases, which may indicate information theft, can also be detected. Such changes appear as anomalous values, that is, data points that deviate significantly from the typical distribution of a dataset. Time-series analysis can help to identify such anomalies and prevent confidential information leaks and data breaches.

Identifying anomalous values in a time series can be challenging, because time-series data

have unique characteristics that differ from those of stochastic datasets. Some of the time-series features to be considered when searching for anomalies are as follows:

- Time-series data show trends that represent long-term changes. For example, an employee's skill improvement may increase the number of database queries per unit of time.
- Many time-series data exhibit seasonality, which is a periodic pattern. A typical example is performing all types of routine operations such as reporting.
- Time-series data can exhibit autocorrelation, meaning that the current value of the data correlates with its past values. Typically, the degree of autocorrelation is high and positive; users perform similar activities daily as part of their functional responsibilities. This makes it difficult to distinguish between the normal and true data fluctuations.
- Time-series data may be non-stationary, meaning that the statistical properties of the data may change over time. This makes it difficult to use traditional statistical methods to detect the anomalies.

To prevent hidden actions by insiders in a database, it is necessary to implement an access control and monitoring system that can detect unusual user activities. The authors proposed a framework for detecting abnormal behavior of database users based on machine learning.

1. An overview of the anomaly detection techniques

Traditional anomaly-detection methods, such as clustering or classification algorithms, may be unsuitable for time-series data because of the dependence between observations over time. Therefore, specialized anomaly detection methods for time series have been developed, such as statistical methods based on time-series decomposition, autoregressive models, and machine learning approaches.

1.1. Statistical methods

Statistical methods are typically used to detect anomalies in time series data. One of the most popular statistical methods is the Z-score method, which calculates the deviation of a data point from the mean and is expressed as standard deviation [1].

Another statistical method is the Grubbs test [2], which detects anomalies by identifying the extreme values in a dataset. The test assumes that the data have a normal distribution, and calculates the maximum deviation from the mean value for a data point.

The (interquartile) interquartile range (IQR) method is another popular method for detecting outliers. It is based on quartiles of the dataset, which divides the dataset into four equal parts. The IQR was calculated as the difference between the third (Q3) and first (Q1) quartiles, and values exceeding 1.5 IQR were considered outliers [3].

The Mahalanobis distance method is a statistical method that measures the distance between a point and the mean of a dataset by considering the covariance of the variables. This method is particularly useful for datasets with multiple variables. Points with a large Mahalanobis distance from the mean are considered outliers [4].

Autoregressive Integrated Moving Average (ARIMA) models are popular methods for modeling and forecasting time-series data [5]. These models can also be used to detect anomalies in data.

Seasonal decomposition of time series (STL) is a method for decomposing time-series data into trend, seasonal, and residual components. Anomalies were detected in the residual components [6].

Wavelet analysis is used to analyze time-series data at different scales. Anomalies can be detected by identifying wavelet coefficients that exceed a certain threshold [7]

1.2. Machine Learning

Within machine learning, there are two main approaches: supervised and unsupervised. The main difference is that one uses labeled data to predict outcomes, whereas the other does not.

In supervised learning, the algorithm "learns" from the training dataset, iteratively making predictions based on that data and adjusting to obtain the correct answer. Although supervised learning models tend to be more accurate than unsupervised learning models, they require prior human intervention to label the data correctly. Unsupervised learning models, on the other hand, work on their own by discovering the

internal structure of unlabeled data. It should be noted that human intervention is still required to validate the output variables. Unsupervised learning is similar to artificial intelligence in that it continues to learn new things with increasing experience.

Machine learning methods are becoming increasingly popular for detecting anomalies in time series data. One such method is the Support Vector Machine (SVM) algorithm [8], which uses a set of labeled data points to train a model that can detect anomalies in new data. The SVM algorithm determines the boundary between the normal and anomalous data points.

One of the simplest machine learning algorithms based on supervised learning is the K-Nearest Neighbor (KNN) algorithm. This algorithm is commonly used as a classification algorithm based on the assumption that similar points can be found near each other. It is widely used in real-world scenarios because it is non-parametric, that is, it does not make any underlying assumptions about the distribution.

Another machine-learning technique is the Isolation Forest algorithm [9], which randomly selects a feature and separation value for each node of the decision tree. Anomalies were defined as data points with a short path length to the leaf node, indicating that they were isolated from the rest of the data.

The Local Outlier Factor (LOF) method is a machine learning method that determines outliers based on their deviation from the surrounding points. It measures the local density of points around a given point and compares it with the local density of the surrounding points. Points whose density is significantly lower than that of their neighbors are considered outliers [10].

Deep Learning.

Deep learning methods, particularly recurrent neural networks (RNN), have shown promise in detecting anomalies in time-series data [11]. RNNs are designed to process sequential data and learn to detect patterns in time series data. Anomalies can be detected by comparing the predicted values obtained using the RNNs with the actual values.

Another deep learning method is the autoencoder [12], which is a neural network trained to reconstruct input data. Anomalies were defined as data points with high reconstruction errors, indicating that they were significantly different from the rest of the data.

1.3. Spectral methods

Spectral methods for detecting anomalies in a time series were used to analyze the time series in the frequency domain. Spectral methods can be used to detect anomalies in time series associated with cyclicality, such as seasonal fluctuations. Anomalies can be detected based on unusual values within certain frequency ranges of the spectrum, which may indicate the presence of unusual cyclic patterns.

An example of a spectral method is the Spectral Anomaly Detection method [13], which is based on the detection of anomalous frequency components in the spectrum of a time series. This method is based on the analysis of the noise component of the spectrum, which can be used to determine the threshold above which the frequency components are considered to be anomalous.

Another example is the spectral-entropy-based measurement method [14] that uses the entropy of the spectrum to detect anomalies. This method is based on the fact that anomalous frequencies in the spectrum have lower entropy than normal frequencies, which allows the detection of unusual patterns in the spectrum of the time series.

2. Proposed methodology and framework

The proposed framework is based on a machine-learning method. As mentioned above, machine-learning methods can be divided into two approaches: supervised and unsupervised.

In situations where the output data are not known and only the input data are known, it is better to use unsupervised learning. Reasons why unsupervised machine learning techniques should be favored in anomaly detection systems:

- A small amount of anomalous data and a large amount of normal data.
- The existence of many different types of anomalies, makes it difficult to train an algorithm, especially if there is little anomalous data.

Also, in the case of insider attacks, databases usually do not have a training sample and if they do, the threat is usually eliminated.

In addition, the computational complexity of unsupervised learning is less than that of supervised learning. Unsupervised learning can be used to analyze real-time data.

The authors proposed a framework whose architecture is based on unsupervised learning (Figure 1):

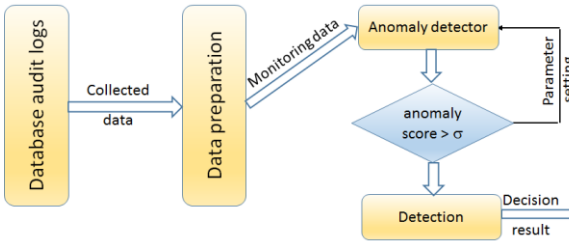

Figure 1. General framework of anomaly detection

The following tasks must be solved to fill in the framework components with functionality:

- analyze the characteristics of insider behavior in a database and, based on this analysis, form a set of attributes whose abnormal values are highly likely to be indicative of insider behavior;
- chose a suitable algorithm to detect anomalies;
- determine the parameters of the algorithm;
- develop a methodology for implementing the framework.

2.1 Analysis of the peculiarities of insider actions

Insiders can use various methods to hide their activities from a database. The characteristic features of insider activities [15,19] are:

- Modifying audit data: Insiders can modify, delete, or add data to the logs.
- Use of third-party tools and unusual technologies: Insiders can use tools such as remote access software (VPN) or programs to intercept traffic and encrypt to gain access to the database without leaving traces.
- Unusual time slots: Insiders can perform their actions during unusual time slots, such as at night or on weekends, to reduce the likelihood of detection.
- Unusual access patterns: An insider gets access to confidential information outside the scope of his or her normal duties or to information he or she does not need for work.
- Policy violation: Violation of a security policy, such as sharing passwords or accessing restricted resources.
- System anomalies: Insiders cause unusual system activity such as unexpected bursts of network traffic or system configuration changes.
- Suspicious email or file transfers: The insider sends large amounts of confidential information outside the organization or uses personal email or cloud storage services.

This set of attributes must be considered when selecting attributes to be recorded. It is also necessary to consider that the insider knows that his operations are being monitored; he can gradually change the nature of his actions so that the monitoring system does not detect significant deviations. In this regard, the anomaly detection task [18] has the following features:

- Definition of a normal area for each possible normal observation without a precise boundary between normal and abnormal observations.
- Anomalies can be the result of fraudulent actions, and these actions are most likely adapted to behave very much like normal actions.
- Normal behavioral characteristics may change, and detection patterns become increasingly outdated over time.
- Lack of available data with labeled anomalies for model evaluation.

2.2 Selecting the attributes to register

The question of the optimal number of attributes that should be retained in a dataset is always controversial because the selection of only some attributes usually leads to a loss of information from the original dataset. Several papers in the field of feature selection argue that a larger number of attributes usually leads to better approximations. However, this may be true for perfect, fully consistent, and noise-free data, when all attributes are independent.

For the selected features to be useful and effective, they must be highly relevant to the task at hand and not redundant [16]. In fact, a large number of published results have demonstrated that a smaller number of selected features can lead to a significant increase in the accuracy of anomaly detection [17]. In addition, a larger number of features stored in the dataset also increases the computational complexity [16].

The authors opted for the following parameters (Table 1,2).

Table 1
Characteristics specific to current session

Parameters	Reason 2
Operation time Login time Logout time	Operations after hours, too long a session
Address	Access is not from a workstation. Standard Ip = 0, otherwise 1

Table 2
Characteristics specific to the database

Parameters	Reason 2
Table Id	Addressing "unusual" tables
Type of access	Select, Insert, Update, Delete
Id of the error associated with the access denial	The number of access denials is counted. Attempts to access data that the user is not allowed to access. Attempts = 1, none = 0

2.3 Choice of anomaly detection method

As seen above, in the case of insider attacks, abnormal values may not differ significantly from normal values and may be difficult to detect using traditional statistical methods. In such cases, machine-learning-based algorithms can be effective.

One of the most popular machine-learning algorithms for anomaly detection is the Isolation Forest algorithm [9]. This algorithm is based on the idea of isolating anomalies rather than detecting normal data points. It works by recursively partitioning data into subsets and then isolating anomalies in sections with fewer data points.

Another popular algorithm for anomaly detection is the Local Outlier Factor (LOF) algorithm [10]. This algorithm measures the local density of data points and identifies outliers as points with significantly lower densities than their neighbors.

Both algorithms are effective in detecting anomalies in datasets where the anomalous values are not significantly different from the normal values. A study comparing the performance of several machine-learning-based anomaly detection algorithms, including Isolation Forest and LOF, found that these algorithms have a high detection rate and a low false positive rate in datasets with few anomalies [9].

The LOF algorithm considers local density and can detect anomalies of different shapes and sizes. However, it is not capable of processing large datasets and is unstable to noise. In addition, the computational complexity of the LOF is significantly higher than that of an Isolation Forest [9].

We selected the Isolation Forest algorithm for anomaly detection. The proposed algorithm has several advantages:

- no training is required;
- easily adapts to the online/incremental mode, which is suitable for detecting anomalies in near real-time;
- effective for detecting anomalies in large datasets;
- has low memory requirements;
- can work with any type of data;
- allows quick preparation of a model; the computational complexity is linearly dependent on time.

2.4 Choice of algorithm parameters

The parameters of the Isolation Forest algorithm are as follows: number of trees (n_estimators), maximum tree depth (max_depth), minimum leaf size (min_samples_leaf), and number of features for each tree (max_features). The algorithm parameters are chosen to optimize the quality of anomaly detection and to speed up the algorithm. The following considerations should be considered when choosing the parameters:

- Increasing the number of trees (n_estimators) may improve the quality of anomaly detection but may decrease the speed of the model.
- Increasing tree depth (max_depth) can improve the quality of anomaly detection but may increase the probability of overtraining the model and reduce model speed.
- Reducing the sample size for building each tree ((max_samples)) may speed up the model but degrade its quality.
- Reducing the feature sample size for each partition (max_features) may speed up the model but degrade its quality.
- The choice of the expected fraction of anomalous objects (contamination) depends on the specific task and can range from 0 to 0.5.

- Using a random initial value (random_state) for the random number generator can affect the training results when training the model. Therefore, a fixed value of the parameter should be used to ensure the repeatability of the experiment.

If the size of the training sample lies between 1000 and 10000 records, each record has nine attributes, and the authors recommend the following parameters of the Isolation Forest model:

- n_estimators: 200. This number of trees is sufficient to provide adequate partitioning density at each tree level for this sample size.
- max_samples: 256. This was the sample size used for building each tree. This value allows the algorithm to maintain adequate speed.
- max_features: 3. This represents the number of features for each partition in the tree. This value allowed us to maintain an adequate variety of features at each tree level.
- contamination: 0.02. This is the expected fraction of abnormal objects in the training sample.

However, it should be noted that the optimal parameters may vary depending on the specific task and the data. Therefore, it is recommended to conduct experiments with different parameter values to determine the optimal parameters for a particular task and dataset.

2.5 Anomaly detection methodology

Based on the results, the tasks solved by the framework were as follows:

Data preparation:

- Loading data from the database into memory.
- Data preprocessing: normalization, conversion of categorical features to numeric features, etc.; the algorithm expects the data type to be a real number.
- Partitioning the data into training and testing samples.

Model Training:

- Set the model parameters (number of trees, tree depth, etc.).
- Forming a set of trees.

Application of the model to verifiable data:

- Processing data on the generated tree set.
- Determining the threshold for classifying objects as abnormal. Calculates the verage path length c(X_i) for instances X_i in iTree (equation 1). Next, the anomaly estimate s(X_i) of each component x in the instance X_i is obtained by calculating equation (2).

$$c(X_i) = 2H(X_i - 1) - \frac{2(X_i - 1)}{X_i} \quad (1)$$

$$H(X_i - 1) \approx \ln(X_i - 1) + e$$

$$s(X_i) = 2^{-\frac{E(l(x))}{c(X_i)}} \quad (2)$$

$$E(l(x)) = \frac{1}{t}\sum_{n-1}^{t} \ln(x)$$

Where

- X_i - the size of the i-th sample;
- H(i) is the harmonic number and it can be estimated by ln(i) + e;
- e = 0.5772156649 (Euler's constant);
- E(l(x)) - expected path length.

The anomaly score ranges from 0 to 1, and a data instance is normal if the score is below 0.5 [9]. Data instances with scores close to one can be defined as outliers. To select the data with maximum scores, they can be sorted.

Interpretation of results:

Analysis of detected anomalies, identification of reasons for abnormal behavior, etc.

A logical flowchart of the proposed methodology is presented in Figure 2.

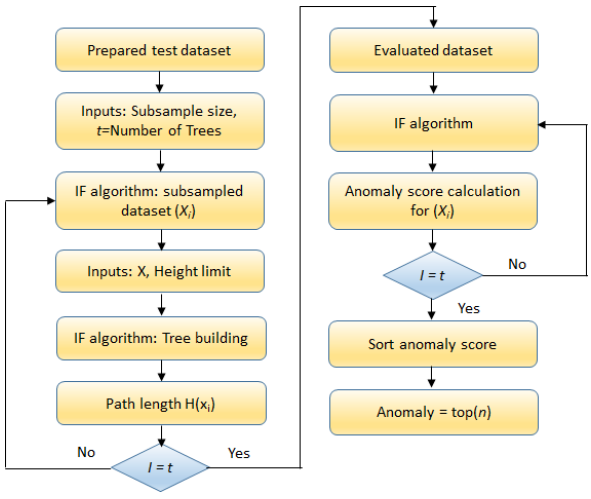

Figure 2 A logic flow chart of proposed framework

To implement the framework, one must use a programming language that supports databases and libraries to implement the IS algorithm, such as Scikit-learn for Python or RLOF for R. Additional customization may also be required to account for the specifics of a particular case. For example, the MS SQL Server has a number of functions that allow the user to view the content of a transaction log. The fn_dblog function allows the user to view the content of a transaction log for a particular database. This function returns a set of lines, each representing a transaction log entry. Similar approaches are implemented in other DBMS. There are also third-party tools such as ApexSQL, Red Gate and Idera.

3. Experimental analysis

The purpose of the experiments described in this section is to determine the effectiveness of the proposed framework in detecting the abnormal behavior of database users.

The datasets were generated as follows. The parameters specified in Table 1 were obtained from the CMU Software Engineering Institute [20]. The parameters in Table 2 were generated such that the values of the recorded parameters were normally distributed, and the coefficient of variation was approximately 30%.

Single abnormal values were distributed uniformly throughout the sample, and their positions of abnormal values were determined using the SQL function rand(). The number of anomalous values was 1.5% and 2% for the first and second samples, respectively N. The research was carried out for anomalous values of 2σ for the first sample and 3σ for the second sample.

Table 3.
Characteristics of the datasets used

Dataset	Points	Outliers	Outliers σ
Dataset 1	10000	150 (1,5%)	2σ
Dataset 2	10000	200 (2%)	3σ

The parameters of the Isolation Forest algorithm are presented in Table 4.

Table 4
Parameters of the Isolation Forest algorithm

n_estimators	max_samples	max_features	contamination
200	256	3	0,02

The metrics are shown in Figure 3. These variables are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

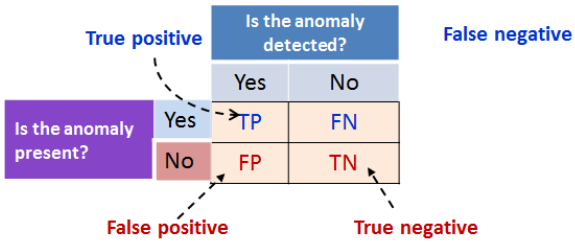

Figure 3: Typical anomaly detection estimates

The Isolation Forest algorithm is unsupervised, which means that it does not require labeled data to train or evaluate its performance. Consequently, there are no true-positive, true-negative, false-positive, or false-negative values that can be used to evaluate the algorithm's performance in the traditional sense.

The experimental estimates of precision, recall, and F1-score can be obtained for the Isolation Forest algorithm by comparing its output to a ground-truth set of labeled outliers. Precision measures the proportion of true positives (i.e., correctly identified outliers) among all data points classified as outliers, whereas recall measures the proportion of true positives identified by the algorithm among all the actual outliers in the dataset. Evidently, the higher the Precision and Recall are, the better. However, in real life, it is impossible to simultaneously reach the maximum of both indicators. Thus, we require a metric that combines the information on the accuracy and completeness of the proposed method. F-score is a metric. The F1-score is the harmonic mean of precision and recall and provides a single summary measure of the performance of the algorithm.

These estimates are calculated using (3)

$$Recall = \frac{TP}{TP + FN}$$
$$Precision = \frac{TP}{TP + FP} \quad (3)$$
$$F\text{ - }score = \frac{2 * Recall * Precision}{Recall + Precision}$$

The experimental results are presented in Table 5.

Table 5.
Experimental results

Dataset	TP	TN	FP	FN	recall	precision	F-score
Dataset 1	124	9747	103	26	0.83	0.57	0.66
Dataset 2	187	9786	14	13	0.94	0.93	0.93

The experimental results confirmed the validity of the methodology proposed by the authors and the framework in general.

Conclusions

In conclusion, the method and framework presented in this article provide a novel approach to detecting anomalies in the actions of users of relational databases, with a particular focus on insider threats. By analyzing the peculiarities of insider's actions in databases and selecting appropriate parameters for recording user behavior, the proposed framework is well-equipped to detect anomalous behavior. The chosen anomaly detection algorithm is justified and recommendations are provided on the choice of algorithm parameters. Overall, this research contributes to the field of information security by offering a new methodology for detecting insider threats and preventing data breaches. Although the proposed framework shows good results in benchmark datasets, its behavior in practical cases requires further investigation. This research can be continued by applying the proposed algorithm in practical evaluations in real databases.

References

[1] J. Behboodian, A. Asgharzadeh. On the distribution of Z-scores. IJSTS, V.32, p. 71-78, 2008. https://ijsts.shirazu.ac.ir/article_2245.html

[2] M. Aslam. Introducing Grubbs's test for Detecting Outliers under Neutrosophic Statistics- an Application to Medical Data. JKSUS V.32, 2020, p. 2696-2700. https://www.sciencedirect.com/science/article/pii/S1018364720301877

[3] T. A. Chawsheen, I. S.i Latif. Detection and treatment of outliers in data sets. IJSS V.9 2006 p. 58-74. https://stats.mosuljournals.com/article_32772_48e925736d326881577b1e6adad8d90c.pdf

[4] M. Kim. Multivariate outliers and decompositions of mahalanobis distance. Communications in Statistics - Theory and Methods, Volume 29, 2000, p 1511-1526. https://www.tandfonline.com/doi/abs/10.1080/03610920008832559

[5] Bianco, Ana Maria, et al, 2001. "Outlier Detection in Regression Models with ARIMA Errors Using Robust Estimates, Journal of Forecasting, v. 20(8), pages 565-579. https://ideas.repec.org/a/jof/jforec/v20y2001i8p565-79.html

[6] E. B. Dagum. Time series modeling and decomposition. Statistica, Vol. 70 No. 4 (2010) https://rivista-statistica.unibo.it/article/view/3597

[7] Scargle J.D. Wavelet and Other Multi-resolution Methods for Time Series Analysis. Statistical Challenges in Modern Astronomy II /Ed. G.J.Babu and E.D.Feigelson. P. 333-347. N.Y.: Springer-Verlag. https://link.springer.com/chapter/10.1007/978-1-4612-1968-2_19

[8] M Debruyne. An outlier map for support vector machine classification. The Annals of Applied Statistics, Vol. 3, No. 4, 2009, pp. 1566-1580 https://arxiv.org/pdf/1009.5818.pdf

[9] Liu, F. T., Ting, K. M., & Zhou, Z. H. Isolation forest. Eighth IEEE International Conference on Data Mining, 2008 pp. 413-422.

https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf

[10] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. LOF: identifying density-based local outliers. ACM SIGMOD Record, Vol. 29, No. 2, 2000. https://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf

[11] Yue Wang, M. Perry, D. Whitlock, J. W. Sutherland. Detecting anomalies in time series data from a manufacturing system using recurrent neural networks. Journal of Manufacturing Systems, V. 62, January 2022, Pages 823-834. https://www.sciencedirect.com/science/article/abs/pii/S0278612520302211

[12] H. Torabi, S. Mirtaheri, S. Greco. Practical autoencoder based anomaly detection by using vector reconstruction error. 2023, Cybersecurity 6, 1 (2023). https://doi.org/10.1186/s42400-022-00134-9 https://cybersecurity.springeropen.com/articles/10.1186/s42400-022-00134-9

[13] C. Peng, Weilin Hu, L. Wang. Spectrum Anomaly Detection Based on Spatio-Temporal Network Prediction. Electronics 2022, 11(11). https://www.mdpi.com/2079-9292/11/11/1770

[14] Ning Jia. Anomaly Detection in Univariate Stochastic Time Series with Spectral Entropy. 2022. https://towardsdatascience.com/anomaly-detection-in-univariate-stochastic-time-series-with-spectral-entropy-834b63ec9343

[15] D.Vaidya Insider Threat (https://www.wallstreetmojo.com/insider-threat/)

[16] Jensen R, Shen Q. Are more features better? A response to attributes reduction using fuzzy rough sets. IEEE Trans Fuzzy Syst. 2009, 17(6), p. 1456–1458. https://www.researchgate.net/publication/224560258_Are_More_Features_Better_A_Response_to_Attributes_Reduction_Using_Fuzzy_Rough_Sets. DOI**:** 10.1109/TFUZZ.2009.2026639

[17] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003, V.3 h. 1157–1182. http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[18] Chandola, V., Banerjee, A., Kumar, V. Anomaly detection: A survey. ACM Computing Surveys (CSUR), 2009, V. 41(3),p. 1-58. DOI: 10.1145/1541880.1541882

[19] M. Al-Mhiqani, R. Ahmad A Review of Insider Threat Detection: Classification, Machine Learning Techniques, Datasets, Open Challenges, and Recommendations. Appl. Sci. 2020, 10(15); DOI:10.3390/app10155208. https://www.mdpi.com/2076-3417/10/15/5208

[20] CMU. Software Engineering Institute. Insider Threat Test Dataset. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099