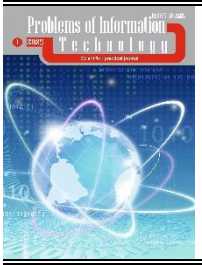

Available online at www.jpit.az

16 (1)
2025

Real-time endpoint anomaly detection using adaptive statistical methods for baseline deviations

Kamran Asgarov

Azerbaijan Technical University, Baku, Azerbaijan

kamran.asgarov.n@student.aztu.edu.az

https://orcid.org/0009-0000-3659-024X

ARTICLE INFO

Keywords:

Endpoint security
Cybersecurity
Anomaly detection
Baseline deviation
Statistical methods

ABSTRACT

Real-time anomaly detection is an important part of endpoint security, which offers a promising alternative to traditional security and monitoring methods. This paper introduces a framework based on adaptive statistical methods for real-time endpoint anomaly detection and investigates six different statistical methods and their effectiveness in detecting anomalies in three anomaly scenarios. The framework approach is based on collecting detailed telemetry metrics that include major endpoint metrics categories such as CPU usage, network activity, disk operations to establish a baseline of normal behavior. Deviations from this baseline are flagged as anomalies. Methods are tested using hyperparameter optimization and evaluated using performance metrics such as F1-score, accuracy, and precision. This study demonstrates the potential of statistical methods for scalable, interpretable, and efficient anomaly detection in endpoint security.

1. Introduction

Endpoint security and monitoring are critical components of cybersecurity. Despite advances in security technology, endpoint security is still one of the biggest challenges in this area. The main reason for this complexity is the constant emergence of new threats and the evolution of existing threats to endpoints (Li and Liu, 2021). Traditional security methods are often unable to adapt to such rapid changes in threats. For instance, traditional antivirus systems use signatures to recognize threats, which are easily bypassed by new threats simply by using obfuscation techniques, or monitoring techniques based on prewritten rules, which are also unstable to new threats because they require constant regulation of the rules themselves (Al-Asli and Ghaleb, 2019).

Anomaly detection offers a promising alternative to traditional endpoint security and monitoring methods (Chandola et al., 2009). The concept of anomaly detection originated in statistics and was later formalized in computer science and machine learning. Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, an anomaly $\mathbf{x}_i \in \mathcal{X}$ is a data point that significantly differs from the majority of the data according to a defined metric or model. In this paper, we consider endpoint telemetry as dataset $\mathcal{X}$ and each element of this dataset is the state of the endpoint at a particular point in time. Telemetry is collected from the endpoint every few seconds, which means that we only have a discrete time of endpoint telemetry points, so we can use natural numbers as index of each telemetry point $\mathbf{x}_i$, where i indicates a specific timestamp.

Endpoint telemetry is the main source of data from which the anomaly will be detected.

Received 23 September 2024, Received in revised form 25 November 2024, Accepted 9 December 2024
http://doi.org/10.25045/jpit.v16.i1.02

Therefore, it is very important to include in this set all key endpoint indicators that need to be continuously monitored. These indicators include six major categories of metrics (Table 1): CPU metrics $(x_1, \dots, x_5)$, GPU metrics (x_6, x_7, x_8), RAM metrics, (x_9, x_{10}) disk operation metrics (x_{11}, x_{12}), network metrics $(x_{13}, \dots, x_{18})$ and process activity metrics (x_{19}, x_{20}, x_{21}).

Table 1. Telemetry metrics

Variable	Description	Unit
x_1	CPU utilization	%
x_2	CPU temperature	°C
x_3	CPU Speed	GHz
x_4	CPU fan speed	RPM
x_5	Average CPU usage across all active processes	%
x_6	GPU utilization	%
x_7	GPU temperature	°C
x_8	GPU fan speed	RPM
x_9	RAM usage	%
x_{10}	Average RAM usage across all active processes	%
x_{11}	Number of disk read operations	-
x_{12}	Number of disk write operations	-
x_{13}	Volume of data sent over the network	KB
x_{14}	Volume of data received over the network	KB
x_{15}	Number of packets sent over the network	-
x_{16}	Number of packets received over the network	Mbps
x_{17}	Network interface speed	-
x_{18}	Total active network connections	-
x_{19}	Total number of running processes	-
x_{20}	Total number of active threads	-
x_{21}	Total number of open handles	-

Next, we build a framework based on the adaptive statistical methods for baseline deviations to detect anomalies in endpoint telemetry dataset. The framework consists of two stages, in first stage it collects endpoint telemetry data shown in table 1 and trains a statistical model on this data, and in the second stage it checks each new telemetry data point for an anomaly using trained model.

2. Related work

Both in the academic literature and in the field of enterprise information systems, much attention has been paid to endpoint cybersecurity. There is a lot of work on malware and endpoint attacks, and most of it uses signature-based methods to detect malware. However, this approach requires constantly analyzing new malware and adding their signatures to an existing malware database. In this article, we will look at a completely different approach to endpoint protection, namely anomaly detection. Anomalies include not only malware, but also any system malfunctions, such as hardware failures. Anomalies can also be related to malfunctioning software that may not be malware. Chandola et al. (2009) provide one of the seminal surveys on anomaly detection techniques, outlining both statistical and machine learning approaches that have since influenced numerous applications in cybersecurity (Chandola et al., 2009).

Asgarov et al. (2024) investigated unsupervised machine learning methods for real-time anomaly detection in endpoints (Asgarov et al., 2024). Their study highlights the critical vulnerabilities of endpoint devices within modern digital ecosystems, where traditional security solutions often fail to identify subtle deviations from normal behaviour. By applying unsupervised methods such as Isolation Forest, One-Class Support Vector Machines, Gaussian Mixture Models, and Local Outlier Factor on endpoint telemetry data. They demonstrated that unsupervised methods can effectively flag anomalies indicative of security breaches or system malfunctions.

The anomaly recognition problem is also often posed in recognizing anomalies in network data. For example, the work of Ahmed et al. (2016) presents a detailed review of network anomaly detection methods and emphasizes the importance of adaptability in the face of ever-changing cyber threats (Ahmed et al., 2016). In their work, they consider an in-depth analysis of four main categories of anomaly detection methods, which include classification, statistics, information theory, and clustering.

In this paper, we investigate the effectiveness of adaptive statistical methods in the task of anomaly detection based on endpoint telemetry. The good thing about these statistical methods is their simplicity compared to machine learning methods, whose performance can often be

described as a "black box". Statistical methods have wide successful applications in anomaly recognition in financial and mathematical problems (Hodge & Austin, 2004). Statistical methods require only a set of data that have been collected at regular intervals, so in this paper we use endpoint telemetry and investigate the presence of anomalies in a set of endpoint telemetry in real time. Endpoint anomaly detection methods that only use endpoint telemetry metric data might not be able to handle all types of complex attacks, but they can help find a variety of anomalies that are hard to find with more traditional methods.

3. Problem statement

An anomaly detection system must be able to handle challenges such as, dynamic baselines that can change depending on endpoint behavior, performance so that the system does not consume a lot of processing power and most importantly anomaly detection of diverse scenarios.

To validate the framework proposed in this paper, was collected real telemetry from an endpoint for a little more than 2 hours at intervals of 5 seconds. Each telemetry data point has average values of all metrics of table 1 within 5 seconds. The total number of data points is 1492. Telemetry dataset is divided into two parts: training and test data in proportions of 75% and 25% (Fig. 1). Statistical models must first be trained on the training dataset, all of which data will be assumed normal by the definition and build its baselines, and then on the test data it must compare this new data with the data it was trained on and check each point for an anomaly.

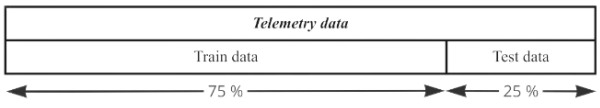

Fig. 1. Telemetry dataset distribution

For proper validation of statistical models, it is necessary that the test data contain both normal data points and anomalies. Therefore, the test data was divided into 4 parts as shown in Fig. 2.

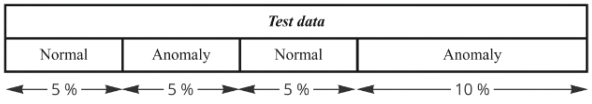

Fig. 2. Test data distribution

Since the test data initially contained only normal data, it was necessary to synthesize anomalies to evaluate the proposed framework's effectiveness. Anomalies were synthesized by simulating different real-life scenarios for comprehensive testing of the model. Following three anomaly scenarios were selected:

1) *Resource exhaustion*. This scenario simulates situations like Distributed Denial of Service (DDoS) attacks or resource-intensive applications running maliciously on endpoints (Hoque et al., 2015). This scenario was simulated by increasing the metrics of data volume and number of packets sent and received over the network, these metrics correspond to the coordinates $(x_{13}, x_{14}, x_{15}, x_{16})$ in Table 1. Also, along with these metrics, the CPU and RAM utilization metrics were increased, which correspond to (x_1, x_9). Thus in this scenario, 6 columns of test data were modified to synthesize anomalies.

2) *Thermal overload*. This scenario simulates overheating due to colling issues which may be caused by hardware failure or cyber-attacks such as "fan speed control attacks" that targets to physically damage hardware components or degrade performance by manipulating cooling systems (Ding et al., 2022). This scenario was simulated by increasing both CPU and GPU temperatures (x_2, x_7) as well as their fan speeds (x_4, x_8). Thus only 4 columns were modified in this scenario.

3) File system abuse. This scenario replicates behaviors typical of ransomware or crypto mining malware whose victims are becoming more and more numerous (Maurya et al., 2018). This scenario was simulated by generating spikes in disk read/write operations (x_{11}, x_{12}) and CPU utilization (x_1) spikes. Only 3 columns were modified to simulate this scenario which makes it, making its anomalies the most difficult to detect.

By simulating these three anomaly scenarios with test data we got three different datasets which can be used to test detection methods. This also means that we have a label for each data point as to whether it is an anomaly or not, which will allow us to use the evaluation methods such as confusion matrix and classification report that are usually used for supervised methods, even though framework's proposed methods are unsupervised (Vujovic, 2021).

4. Proposed approach

In the anomaly detection framework presented, each statistical method relies on the train data to establish a baseline that encapsulates normal

behavior based on endpoint telemetry in which it was trained. Each method extracts statistical characteristics, such as mean, standard deviation, or median, from the train data to define this baseline. For every data point x in the test data, its deviation from the baseline is calculated, and if this deviation exceeds a predefined threshold, the data point is flagged as an anomaly.

1) Z-Score. In the Z-score method (Venkataanusha et al., 2019), the mean μ and standard deviation σ of each data point are computed from the train data:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i, \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \mu)^2}$$

For each data point x in the test data, the Z-score is calculated as:

$$z = \frac{\mathbf{x} - \mu}{\sigma}$$

A data point is flagged as anomaly if $|z| > \tau$, where τ is threshold that determines the sensitivity of detection.

2) Modified Z-score (MAD). The Modified Z-score method leverages the median as a measure of central tendency and the Median Absolute Deviation (MAD) as a robust measure of dispersion. From the train data, the median of $\boldsymbol{x}$ and MAD are computed:

$$\text{MAD} = \text{median}(|\mathbf{x}_i - \text{median}(\mathbf{x})|)$$

The modified Z-score for each data point in the test data is then calculated as:

$$m = 0.6745 \cdot \frac{\mathbf{x} - \text{median}(\mathbf{x})}{\text{MAD}}$$

If $|m| > \tau$, the data point is identified as an anomaly.

3) Interquartile range (IQR). In the IQR method (Vinutha et al., 2018), the first ($Q1$) and third quartiles ($Q3$) are computed from the train data, and the interquartile range is defined as:

$$\text{IQR} = Q3 - Q1$$

Anomalies are detected by defining bounds:

$$\text{Lower Bound } = Q1 - \tau \cdot \text{IQR},$$
$$\text{Upper Bound } = Q3 + \tau \cdot \text{IQR}$$

A data point is flagged as an anomaly if it lies outside these bounds.

4) Exponential smoothing. Exponential Smoothing (Brown et al., 1961) gives greater weight to recent observations to model dynamic baselines. The exponentially weighted mean S_t is computed iteratively:

$$S_t = \alpha \cdot \mathbf{x}_t + (1 - \alpha) \cdot S_{t-1}$$

where α is the smoothing factor. For each test data point $\mathbf{x}_t$, the deviation from the smoothed mean is compared to a threshold:

$$|\mathbf{x}_t - S_t| > \tau \cdot \sigma_S$$

where σ_S is the standard deviation of the smoothed values. A data point is flagged as an anomaly if the deviation exceeds the threshold.

5) Moving average. The moving average method calculates a rolling mean and standard deviation over a fixed window size w. From the train data, the mean μ_t and standard deviation σ_t are computed as:

$$\mu_t = \frac{1}{w}\sum_{i=t-w+1}^{t} \mathbf{x}_i, \sigma_t = \sqrt{\frac{1}{w}\sum_{i=t-w+1}^{t} (\mathbf{x}_i - \mu_t)^2}$$

An anomaly is flagged if the deviation of a test data point $\mathbf{x}_t$ exceeds:

$$|\mathbf{x}_t - \mu_t| > \tau \cdot \sigma_t$$

6) Exponentially weighted moving average (EWMA). The EWMA method (Perry, 2011) computes a smoothed average $EWMA_t$ and standard deviation σ_{EWMA} using exponential weights:

$$\text{EWMA}_t = \alpha \cdot \mathbf{x}_t + (1 - \alpha) \cdot \text{EWMA}_{t-1}$$

Anomalies are identified by comparing test data points to dynamic bounds:

$$\text{Upper Bound } = \text{EWMA}_t + \tau \cdot \sigma_{\text{EWMA}},$$
$$\text{Lower Bound } = \text{EWMA}_t - \tau \cdot \sigma_{\text{EWMA}}$$

If data point lies outside these bounds, it is flagged as an anomaly.

5. Experimental results

All methods were trained on a same train data and tested using a hyperparameter optimization process to evaluate their anomaly detection performance on the test data. Each method was tested with different hyperparameters in all three anomaly scenarios while maximizing the methods' performance metrics such as F1-Score, accuracy and precision, which are defined as following:

$$\text{precision } = \frac{\text{TP}}{\text{TP} + \text{TN}}$$

$$\text{F1-Score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TP} + \text{FP} + \text{FN}}$$

where true positive (TP) stands for the number of data points that are correctly predicted as the positive class, false positive (FP) stands for the number of data points that are incorrectly predicted as positive class, false negative (FN) stands for the number of data points that are incorrectly predicted as negative class and true negative (TN) stands for the number of data points that are correctly predicted as negative class.

Each method has a common hyperparameter τ, which is a threshold that determines the sensitivity of the detection. EWMA and exponential smoothing methods have hyperparameter α, which is smoothing factor. And moving average method has a distinctive hyperparameter of window size w. By systematically tuning hyperparameters for each method, the evaluation ensured that comparisons were made under optimal conditions for each approach (Simon et al., 2023).

Experimental results showed that all statistical methods exhibit very close results in all three anomaly scenarios, as shown in tables 2, 3 and 4. The Z-score method achieved the highest overall performance, obtaining an F1-score of 0.93 and an accuracy of 0.91, indicating its reliability in detecting anomalies using a well-defined standard deviation approach. The IQR method came second in terms of performance results with an average F1-score of 0.87 and accuracy of 0.82 outperforming other methods and closely approaching the Z-score method. Other methods demonstrated average F1-Scores; however, their accuracy was very low, with precision also falling short, making them unreliable for endpoint anomaly detection.

Table 2. Classification metrics for resource exhaustion scenario

Variable	Best hyperparameters	F1-Score	Accuracy	Precision
Z-Score	$\tau = 4.08$	0.93	0.91	0.87
MAD	$\tau = 1.50$	0.75	0.60	0.60
IQR	$\tau = 2.33$	0.88	0.84	0.79
Exponential smoothing	$\alpha = 0.1$ $\tau = 4.43$	0.79	0.67	0.65
Moving average	$w = 10$ $\tau = 4.34$	0.76	0.62	0.61
EWMA	$\alpha = 0.1$ $\tau = 4.43$	0.79	0.67	0.65

Table 3. Classification metrics for thermal overload scenario

Variable	Best hyperparameters	F1-Score	Accuracy	Precision
Z-Score	$\tau = 4.03$	0.93	0.91	0.87
MAD	$\tau = 1.50$	0.75	0.60	0.60
IQR	$\tau = 1.61$	0.85	0.79	0.74
Exponential smoothing	$\alpha = 0.1$ $\tau = 4.43$	0.79	0.67	0.65
Moving average	$w = 10$ $\tau = 4.34$	0.76	0.62	0.61
EWMA	$\alpha = 0.1$ $\tau = 4.43$	0.79	0.67	0.65

Table 4. Classification metrics for file system abuse scenario

Variable	Best hyperparameters	F1-Score	Accuracy	Precision
Z-Score	τ = 4.08	0.93	0.91	0.87
MAD	τ = 1.50	0.75	0.60	0.60
IQR	τ = 2.33	0.88	0.84	0.79
Exponential smoothing	α = 0.1 τ = 4.43	0.79	0.67	0.65
Moving average	w = 10 τ = 4.34	0.76	0.62	0.61
EWMA	α = 0.1 τ = 4.43	0.79	0.67	0.65

A graph was plotted to analyze the variation in F1-score as a function of the threshold (τ) across all methods for the "file system abuse" anomaly scenario. The threshold (τ) is a hyperparameter of all methods used, and all methods were tested with threshold values ranging from $\tau = 1.5$ to $\tau = 4.5$, as these values were found to provide the best balance between sensitivity and precision. Threshold values outside this range typically detect only the most obvious anomalies, leading to decreased performance.

For methods with additional hyperparameters, the best values were selected based on the results presented in table 4. As observed in figure 3 for threshold values less than $\tau \approx 2.7$, the IQR method performs better than Z-score and after this threshold Z-score outperforms all methods. It can also be seen that after a certain value of τ F1-score of the methods stops improving, for example for IQR method this value is $\tau = 2.33$.

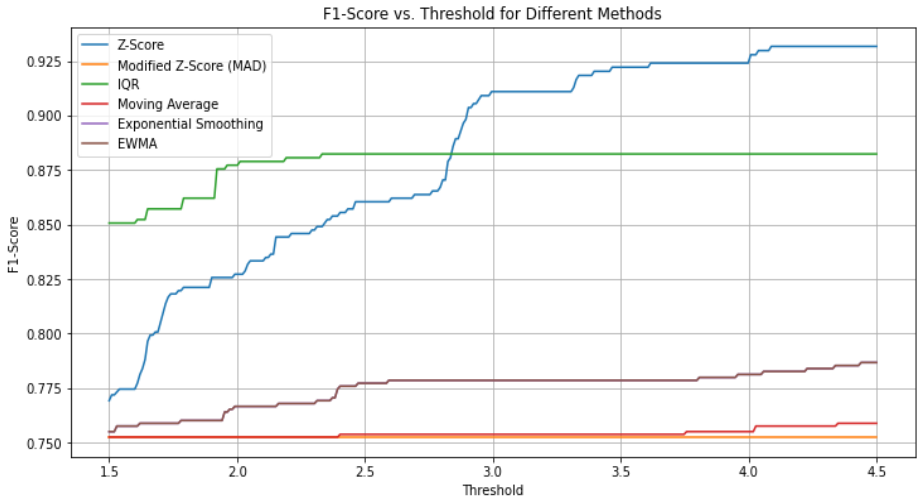

Fig. 3. Test data distribution

Figure 3. illustrates how each method's F1-score changes as we adjust the threshold (τ). At lower thresholds, methods like the Z-score are too sensitive, flagging minor fluctuations as anomalies, which results in many false positives. On the other hand, the IQR method, with its focus on the middle 50% of data, starts off with a higher F1-score because it naturally ignores extreme values, and it reaches a steady performance around $\tau \approx 2.33$. As threshold increases, the Z-score's performance improves, because it starts to distinguish actual anomalies from typical operational noise more effectively.

Curves corresponding to moving average, exponential smoothing, and EWMA methods show smoother changes, which reflects their purpose of smoothing short term deviations. These trends indicates a phenomena of endpoint

telemetry: common operational noise and true anomalies. At low thresholds, systems are overly sensitive, misinterpreting typical, transient deviations as anomalies, while at optimal thresholds the methods effectively balance false detections and true anomaly capture. Further increases in τ lead to performance degradation as true anomalies begin to be overlooked, which demonstrates how each statistical approach overcomes the inherent tension between responsiveness and robustness in dynamic environments.

6. Conclusion and future work

This study presented a framework based on adaptive statistical methods for real-time endpoint anomaly detection. The effectiveness of the framework was tested by 6 different statistical methods. All these methods were tested on the same dataset, which consisted of real endpoint telemetry, which consisted of 21 different metrics. On the test dataset the anomaly process was simulated using three different anomaly scenarios.

Methods showed similar results in all three anomaly scenarios. Among six methods tested, the Z-score method with F1-score of 0.93 and accuracy of 0.91 gave the best results. The second best result was in IQR method, which also performed well in anomaly detection. Other methods showed very low accuracy results, which indicates unreliability of using these methods in endpoint anomaly detection.

Future work consider telemetry data feature selection and investigating performance of statistical methods with different set of features. A hybrid of adaptive statistical methods with machine learning methods can also be explored, as each can detect anomalies based on set features that fits best for each method, and then make a final decision about telemetry data point based on the results of both types of methods rather than only single method.

References

Ahmed, M., Mahmood, A. N. & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31. https://doi.org/10.1016/j.jnca.2015.11.016

Al-Asli, M. & Ghaleb, T. A. (2019) Review of Signature-based techniques in antivirus products. International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, (pp. 1-6.) https://doi.org/10.1109/ICCISci.2019.8716381

Asgarov, K. N., Imamverdiyev, Y. N. & Abutalibov, M. M. (2024). Unsupervised machine learning for real-time anomaly detection in endpoints. Journal of Modern Technology and Engineering, 9(3), 141-155. https://doi.org/10.62476/jmte93141

Brown, R. G., Meyer, R. F. & D`Esopo, D. A. (1961). The fundamental theorem of exponential smoothing. Operations Research, 9(5), 673-687.

Chandola, V., Banerjee, A. & Kumar, V. (2009) Anomaly detection: a survey. ACM computing surveys, 43 (3), 1-58. http://doi.org/10.1145/1541880.1541882

Ding, S., Gu, W., Lu, S., Yu, R. & Sheng, L. (2022), Cyber-attack against heating system in integrated energy systems: model and propagation mechanism. Applied Energy, April 2022, 311. https://doi.org/10.1016/j.apenergy.2022.118650

Hodge, V. & Austin, J. (2004). A survey of outlier detection methodologies. Artificial Intelligence Review, 22, 85-126. https://doi.org/10.1023/B:AIRE.0000045502.10941.a9

Hoque, N., Bhattacharyya, D. K. & Kalita, J. K. (2015). Botnet in DDoS attacks: trends and challenges. IEEE Common Survey Tutorials, 17, 2242-2270. https://doi.org/10.1109/COMST.2015.2457491

Li, Y. & Liu, Q. A. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176-8186. https://doi.org/10.1016/j.egyr.2021.08.126

Maurya, A. K., Neeraj, K., Alka, A. & Raees, A. K. (2018). Ransomware: evolution, target and safety measures. International Journal of Computer Sciences and Engineering, 6(1), 80-85. https://doi.org/10.37934/araset.39.2.110131

Perry, M. B. (2011). The exponentially weighted moving average. Wiley Encyclopedia of Operations Research and Management Science. https://doi.org/10.1002/9780470400531.eorms0314

Simon, S., Kolyada. N., Akiki, C., Potthast, M., Stein, B. & Siegmund, N. (2023). Exploring hyperparameter usage and tuning in machine learning research. 2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN), Melbourne, Australia, (pp. 68-79). http://doi.org/10.1109/CAIN58948.2023.00016

Venkataanusha, P., Anuradga, C., Murty, P. & Chebrolu, S. K. (2019). Detecting outliers in high dimensional data sets using Z-Score methodology. International journal of innovative technology and exploring engineering (IJITEE), 9, 48-53. http://doi.org/10.35940/ijitee.A3910.119119

Vinutha, H., Poornima, B. & Sagar, B. (2018). Detection of outliers using interquartile range technique from intrusion dataset. In Information and Decision Sciences, Proceedings of the 6th International Conference on Ficta, Springer: Berlin/Heidelberg, Germany, (pp. 511-518). http://doi.org/10.3390/electronics13234735

Vujovic, Z. D. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6). http://doi.org/10.14569/IJACSA.2021.0120670