The 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
22-25 September, 2021, Cracow, Poland

Visualization of Data Cubes for Anomaly Detection in Network Traffic Data Streams

Volker Ahlers, Tim Laue, Nils Wellermann, Felix Heine
University of Applied Sciences and Arts Hannover, Germany
{volker.ahlers, tim.laue, nils.wellermann, felix.heine}@hs-hannover.de

Abstract — **For anomaly-based intrusion detection in computer networks, data cubes can be used for building a model of the normal behavior of each cell. During inference an anomaly score is calculated based on the deviation of cell metrics from the corresponding normality model. A visualization approach is shown that combines different types of diagrams and charts with linked user interaction for filtering of data.**

Keywords* — *network security, data cubes, multidimensional analysis, user interfaces, information visualization

I. Introduction

There are currently three different approaches to intrusion detection in computer networks. First, traditional intrusion detection systems (IDS) like Snort [1] or Suricata [2] are *signature-based* as they use a pre-defined set of attack signatures to detect attacks. They can find these attacks with high precision, however, they are limited to well-known attacks.

Second, machine learning based approaches (see e.g. [3]–[6]) use *classifiers* like neural networks or random forests. They are trained on data containing attacks and normal instances, usually on a flow basis. These classifiers often achieve good results on benchmark datasets. However, the model is trained only on those attack types that are present in the training data. Furthermore, it is questionable whether the learned models generalize to new networks. In [7], the authors show that this is not the case, i.e. the model is overfitted to the network used during benchmark construction. This makes it nearly impossible to use this approach in real-live scenarios, as one would need training data collected from the target network *including* a broad range of attack types to be learned.

The third approach is *anomaly detection*, or, more specifically, one-class classification. The idea is to train a model on normal-only data (specific to a network), and to use this model to find traffic patterns that deviate from the previous normal behavior. This approach has the benefit that it is in principle possible to detect new kinds of attacks, if they result in network usage that differs from the normal behavior. Furthermore, the approach can be applied in practice, assuming that we can collect attack-free training data that is representative for the normal behavior of the network.

Many types of attacks, however, cannot be detected by looking at each flow individually, as they manifest as *collective anomalies* [8]. The typical approach used in network datasets is to provide features that aggregate data from multiple flows, i.e. the number of connection to the same host within the last 20 seconds. However, these is typically only a small amount of hand-crafted features which is rather generic and might miss interesting aggregated statistics. For example, a high number of connections from server X as well as a high amount of traffic over port P might be unsuspicious on their own, while the combination (X, P) within a short time interval might indicate an attack.

Recent approaches [9], [10] define such aggregations in a combinatorially high-dimensional space more flexible and adapted to the target network. Multidimensional analysis is a versatile tool to reach this goal. Multidimensional data consist of several independent dimensional attributes (or dimensions) and one or several dependent metric attributes (or metrics), such as counts for given combinations of dimension values. Such data can be efficiently stored and processed in data warehouses as OLAP (online analytical processing) cubes, or data cubes for short [11]. Using the OLAP model, we can for example look at the traffic of a single host, the traffic in a subnet, the https traffic of a host, the https traffic in a whole subnet, etc. OLAP allows to filter and aggregate over all possible combinations of dimensions.

A data cube with n independent dimensions can be understood as an n-dimensional array. For $n \leq 3$ dimensions a literal visualization as an n-dimensional hypercube might in principle be possible (although this would not account for aggregations), whereas for $n > 4$ the human mind is not capable of understanding direct visualization, thus calling for alternative approaches. Different methods for visualizing data cubes have been proposed, e. g., using heat maps, scatter plots [12]–[14], or radial tree layouts [15].

Anomalies detected by our method could have different

This work was financially supported by the German Federal Ministry of Education and Research (BMBF), project *GLACIER* (grant no. 16KIS0950, https://www.glacier-project.de/index.php/home-eng.html).

DOI of the original publication: https://doi.org/10.1109/IDAACS53288.2021.9660978

causes: 1) an attack, 2) a change in the behavior of a device or a human, 3) a reconfiguration of the network, or 4) a too narrow model for normality. To distinguish between these cases, a human operator is needed. To be able to understand the anomaly scores related to high dimensional OLAP cubes, good visualization techniques are required.

In the following we present a visualization approach for data cubes used for anomaly detection in security-relevant network traffic data streams. Our approach provides different views on the data by combining different kinds of diagrams and charts, which are linked to each other to allow consistent filtering of data. The aim is to give an overview of high anomaly scores in order to facilitate the detection of both critical anomalies and false positives.

Visualization dashboards are a central component of SIEM (security information and event management) systems, using bar charts, heat maps, and other types of diagrams [16]. Multidimensional visualization methods like parallel coordinates have also been applied to network traffic analysis [17], [18]. The novelty of our approach consists of the application of different visualization methods to OLAP cubes, enabling 1) the efficient filtering and aggregation of combinations of attribute dimensions, and 2) the computation of anomaly scores from normality models specific to individual cells of the cube.

II. Data Cubes for Anomaly Detection

In this section, we will describe our anomaly detection approach more detailed. First, we formally define cubes and related terms. Then we describe our anomaly model, and finally the data used in the following sections.

A. Formal Definition

Assume we are given multidimensional data with independent *dimensional attributes* (dimensions) $A_1, \ldots, A_n$, a single *metric attribute* (metric) M, and a relation R that attributes a metric value m to all tuples $(a_1, \ldots, a_n)$ of values of the n dimensions. It is further assumed that the dimensional attributes are finite and of nominal, ordinal, or binned cardinal scale, i. e., have discrete values $A_i \in \{a_{i1}, a_{i2}, \ldots, a_{im_i}\}$. A *cell* is a tuple $c = (a_1, \ldots, a_n)$ with each a_i being either a specific dimension value a_{ij} or $*$, meaning “all”. The metric value $m(c)$ of each cell is the sum of the metrics of all tuples of the relation R that match the cell pattern. The *cube* is the set of all cells c_i for relation R with their metric values $m(c_i)$.

Any cell with specific values in all dimensions is a *base cell*, while all other cells (having at least one $*$) are *aggregate cells*. The single cell $(*, \ldots, *)$ is the *apex cell*. A *cuboid* consists of all cells with a common pattern, e. g., for $n = 3$ the cuboid $(A_1, *, A_3)$ consists of all cells with specific values for dimensions A_1, A_3 and $*$ for dimension A_2. A cuboid is thus a subset of the cube. Child and parent relations, which can be defined for cells and cuboids, are not relevant for this work.

B. Anomaly Score

We now assume that the dimensions of a cube represent features relevant for network traffic, such as IP addresses, ports, or network protocols, and the metric counts events (connections or packets) with specific combinations of dimension values within a given time slice.

For each cell of the training data cube that meets an iceberg condition, e. g., that contains a minimum number of event counts, a *normality model* is built. A simple normality model is, e. g., given by a normal distribution of event counts over time slices, i. e., by assuming that $e \sim N(\mu, \sigma^2)$, where e is the event count, $\sim$ reads “is distributed as”, and $N(\mu, \sigma^2)$ represents the normal distribution with mean μ and variance σ^2. More advanced models are of course possible.

During inference, the event counts within a given cell of the live data are compared to the normality model. Staying with the normal distribution model introduced above, the *anomaly score* is then computed as

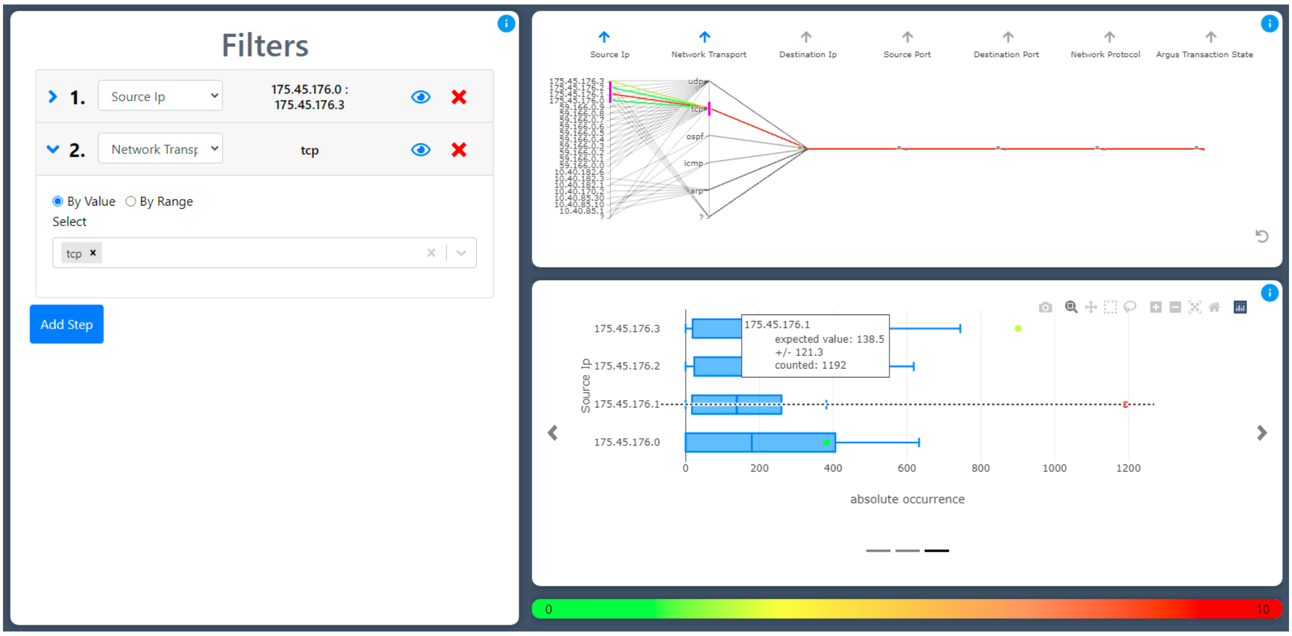
$$s = \min\left(\frac{|e - \mu|}{\sigma}, 10\right), \tag{1}$$

where c is the event count for a given cell, μ is the mean and σ the standard deviation of the normality model for that particular cell. s can be understood as the number of standard deviations the event count differs from the mean of the normality model; for practical purposes s is limited to $s \leq 10$. The *anomaly cube* consists of all cells with the anomaly score s as a new metric. *Anomaly cuboids* are defined accordingly. Further details and evaluations of our approach are explained in [10].

C. Data Preparation

As a demonstration we apply our method to the UNSW-NB15 data set, which consists of real network flow and packet data with normal traffic and labeled attacks [19]–[21]. The following seven dimensions have been selected from the data set (original name in parentheses): *source IP* (*srcip*), *destination IP* (*dstip*), *source port* (*sport*), *destination port* (*dsport*), *network protocol* (*service*), *network transport* (*proto*), *argus transaction state* (*state*).

The UNSW-NB15 data are first divided into 20 min time slices. The data of each time slice are then cubed with the above-named dimensions and the event count per time slice as metric attribute. An iceberg condition is applied, sorting out cells with too few counts. For training the normality models the time interval from 06:00 to 12:00 on 2015-02-18 has been used, corresponding to 18 cubes (6 × 3 time slices à 20 min). Events labeled as attacks have not been used for training. For inference, i. e., for creating the anomaly cube, the single time slice from 12:00 to 12:20 on the same day, 2015-02-18 has then been used. For each cell that has a normality model the anomaly score is computed as defined above.



Filters

1. Source Ip 175.45.176.0 : 175.45.176.3

2. Network Transport tcp

● By Value ○ By Range

Select

tcp ×

Add Step

Figure 1. User interface with four components: filters, parallel coordinates diagram for the current cuboid, choice of different charts (bar chart, heat map, modified box plot, the latter of which is shown in the figure), legend for the colormap. Filters are applied to the dimensions *source IP* and *network transport*, while all other dimensions are aggregated, as indicated by the asterisk $*$.

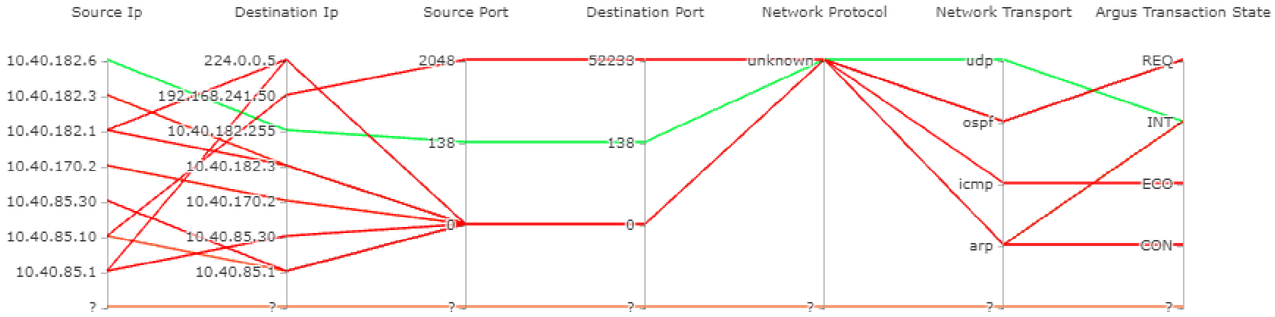


Figure 2. Parallel coordinates diagram for the base cuboid of the anomaly cube, as shown after reading the data.

III. Visualization of Data Cubes

For the visualization of the outlier cubes we developed a web application based on TypeScript, React, and Plotly.js. The user interface of the frontend consists of four components (cf. Fig. 1): a list of adjustable filters, a parallel coordinates diagram giving an overview of the current cuboid, a choice of different charts (bar chart, heat map, modified box plot, see below), and a legend for the common colormap which is used in all diagrams. The first three components are connected in a way that user interactions in one component influence the data displayed in the other components; for details see Sec. III-C below.

The backend consists of the main control component and a data service, which reads the data (i. e., the outlier cubes) and provides specific views on the data as requested by the frontend components. For demonstration UNSW-NB15 data are pre-processed as described above and then loaded from CSV files.

A. Parallel Coordinates

Initially the base cuboid $(A_1, A_2, A_3, \dots)$ is displayed by means of a parallel coordinates diagram (cf. Fig. 2). Parallel coordinates provide a method for visualizing multivariate or multidimensional data by providing one vertical axis for each dimensional attribute and connecting values across the axes according to each cell [22]. Due to the iceberg condition only a small number of cells with high packet counts is shown. The line connecting the question marks ? represents a *default cell*, which aggregates data that would otherwise be lost due to the iceberg condition (details will be published elsewhere).

The color of each line connecting the axes indicates the anomaly score (cf. Eq. 1) of the corresponding cell, according to the colormap shown by the legend at the bottom right of Fig. 1 (from green for $s = 0$ to red for

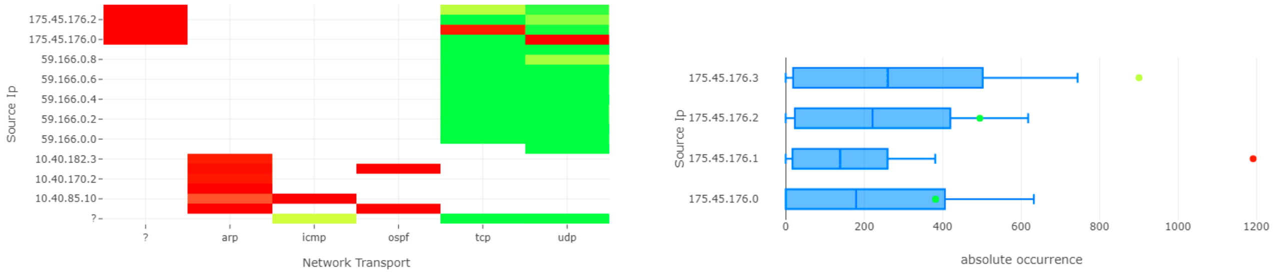


Figure 3. Heat map (left) and modified box plot (right) for the dimensions *source IP* and *network transport*, with the latter filtered to TCP for the modified box plot (cf. text for further explanations). For the heat map, only every second vertical axis label is printed due to lack of space. The box of the modified box plot displays mean and standard deviation of the normality model, the whiskers two standard deviations, and the colored dot the event count of the current cell.

$s = 10$). The high anomaly scores for almost all cells can be considered as false positives, since the combinations of dimension values in the base cuboid are very specific.

B. Bar Chart, Heat Map, and Modified Box Plot

In the following we illustrate a typical workflow from an security operator's view. First we define a *loose dimension* as a dimension that is not aggregated and has more than one possible value after filtering. An example can be seen in Fig. 1, where *source IP* is a loose dimension with more than one possible value, whereas *network transport* has only one possible value and all other dimensions are aggregated.

Step 1: After setting the first filter we have only one loose dimension, which is by default displayed as a *bar chart* in the charts component, with the height and color of each bar corresponding to the anomaly score (not shown here). This can give a first indication of single abnormal values of certain dimensions. We can further filter the dimension to a range of interesting values.

Step 2: By adding a second filter we have two loose dimensions, which can be displayed as a *heat map*, visualizing the anomaly score by color in dependence of two dimensions. An example is given by Fig. 3 (left) for the anomaly cuboid $(\textit{source IP}, \textit{network transport}, *, *, *, *, *)$, where abnormal cells are found, e. g., for the IP range 175.45.176.* in combination with TCP and UDP.

Step 3: Filtering *network transport* to the single value TCP and *source IP* to the interesting range 175.45.176.*, we have only one loose dimension left (*source IP*). Now a *modified box plot* can be displayed, which allows to compare the event counts of cells with the corresponding normality models. For the given example this is shown in Fig. 3 (right). The box plot is modified in the sense that the mean and standard deviation are shown instead of the median and quartiles; the whiskers show twice the standard deviation. The circle represents the event count of the current cell; it is colored according to the anomaly score.

Result: From the modified box plot it becomes obvious that the number of TCP connections from 175.45.176.1 is approximately eight standard deviations larger than the mean of the corresponding normality model. The operator can then look further at the specific host and network protocol to decide if this is critical.

C. Filters and User Interaction

The filters, parallel coordinates, and charts components are connected in order to allow coherent user interaction. Obviously changing the filters influences the selection of data being displayed. The other way round, dimensions can be filtered by interactively selecting single values or value ranges in either the parallel coordinates diagram, the bar chart, or the heat map. The heat map allows 1D and 2D selections, i, e., selections of values within one or both displayed dimensions.

As an example, Fig. 1 shows the user interface for the choice of filters described in Sec. III-B for the modified box plot. The filters are visible in the parallel coordinates diagram, with all other cells grayed out. Detailed information on displayed data is provided via mouseover boxes, e. g., for the normality model shown in the modified box plot. Furthermore, the order of dimensions in the parallel coordinates diagram can be changed by dragging an axis to a different position, and the sorting of the values of a dimension can be reversed by the blue arrow above each axis. Finally, filters can be deactivated or removed by the eye and x buttons, respectively.

IV. Conclusion

In conclusion we have presented a visualization approach for cubed data with the application of anomaly detection in network traffic data. Anomaly scores are computed based on normality models specific to individual cells of the cube. These anomaly scores are visualized by means of parallel coordinates, bar charts, heat maps, and modified box plots, with filters allowing the user to select specific cuboids. For demonstration we have applied our approach to the UNSW-NB15 data set.

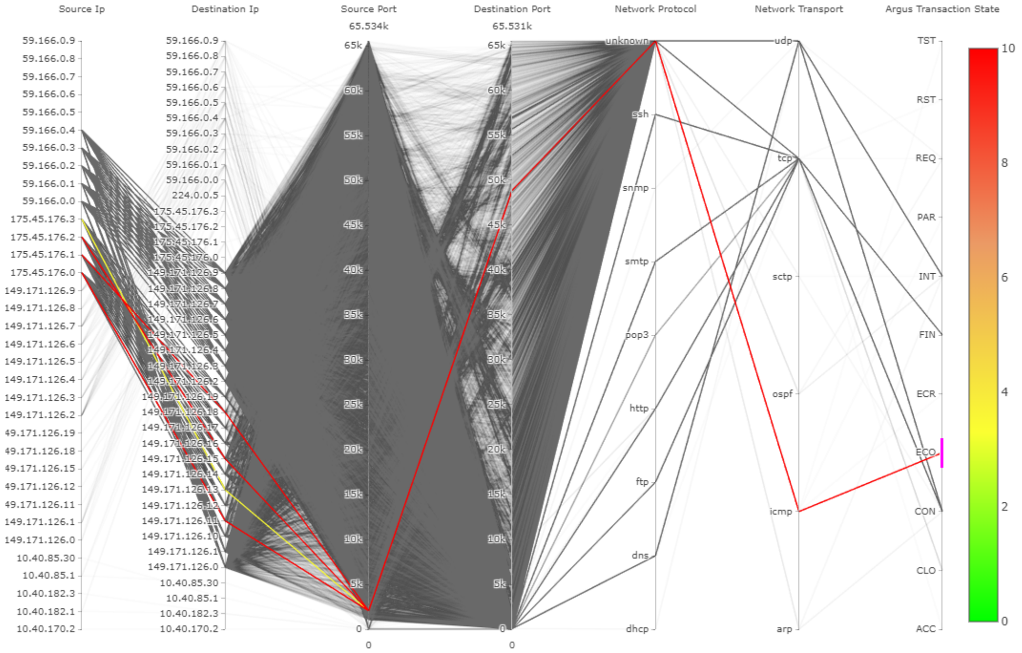


Figure 4. Parallel coordinates diagram for the full anomaly cube, with argus transaction state filtered to ECO.

One point of discussion was the amount of data that should be shown in the parallel coordinates diagram. An alternative approach would be to show the full cube, i. e., all possible cells regardless of the iceberg condition, and to highlight the selection defined by the filters (cf. Fig. 4), which would provide much more information on the data set in the parallel coordinates diagram. For large data sets, however, it is not feasible to keep all cells when building normality models.

Our current work is focused on the integration of the visualization approach into a SIEM (security information and event management) system which should then be applied to real data streams. When attached to a SIEM system, our visualization approach provides a visual representation of the event data within a certain time frame, emphasizing anomalies. Equipped with this view, SOC (security operations center) employees can quickly assess whether a part of the network needs closer inspection and subsequent countermeasures, or find an overview of affected systems after discovering a security breach through different means. Possible scenarios include port scans, which would be visible in a heat map of destination IPs and port numbers as one or more IP-columns showing mostly red cells, or exfiltration processes, which would be represented in the parallel coordinates diagram as a red line from an internal source IP to an external destination IP and possibly an application protocol like FTP used for data transfer.

Further possible research directions include accounting for dimension hierarchies (as in IP addresses) and conducting a usability study of our visualization approach.

Acknowledgment

Special thanks go to the students who contributed to the master's project *CubeVis* at University of Applied Sciences and Arts Hannover: J. Badenhop, M. Bublitz, R. Duprat, E. Kirch, M. Senge, D. Schlenk, M. Ottlik.

Fruitful discussions with C. Kleiner, P. Klostermeyer, K.-O. Detken, T. Klecker, D. Mahrenholz, T. Eisert, and further members of the GLACIER project are gratefully acknowledged.

References

[1] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proceedings of the 13th Conference on Systems Administration (LISA-99), Seattle, WA, USA, November 7-12, 1999*, D. W. Parter, Ed. USENIX, 1999, pp. 229–238. [Online]. Available: http://www.usenix.org/publications/library/proceedings/lisa99/roesch.html

[2] "Suricata," https://suricata.io/, accessed: 2021-07-08.

[3] O. Faker and E. Dogdu, "Intrusion Detection Using Big Data and Deep Learning Techniques," in *Proceedings of the 2019 ACM Southeast Conference*. ACM, 2019, pp. 86–93. [Online]. Available: https://doi.org/10.1145/3299815.3314439

[4] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52 843–52 856, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2869577

[5] O. Savenko, A. Sachenko, S. Lysenko, G. Markowsky, and N. Vasylkiv, "Botnet detection approach based on the distributed systems," *International Journal of Computing*, vol. 19, no. 2, pp. 190–198, 2020. [Online]. Available: https://doi.org/10.47839/ijc.19.2.1761

[6] V. Hamolia, V. Melnyk, P. Zhezhnych, and A. Shilinh, "Intrusion detection in computer networks using latent space representation and machine learning," *International Journal of Computing*, vol. 19, no. 3, pp. 442–448, 2020. [Online]. Available: https://doi.org/10.47839/ijc.19.3.1893

[7] S. Al-Riyami, F. Coenen, and A. Lisitsa, "A Re-evaluation of Intrusion Detection Accuracy: Alternative Evaluation Strategy," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018. [Online]. Available: https://doi.org/10.1145/3243734.3278490

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[9] D. Bruns-Smith, M. M. Baskaran, J. Ezick, T. Henretty, and R. Lethin, "Cyber security through multidimensional data decompositions," in *2016 Cybersecurity Symposium (CYBERSEC)*, 2016, pp. 59–67. [Online]. Available: https://doi.org/10.1109/CYBERSEC.2016.017

[10] F. Heine, "Outlier detection in data streams using olap cubes," in *New Trends in Databases and Information Systems. ADBIS 2017 Short Papers and Workshops*, M. Kirikova *et al.*, Eds. Cham, Switzerland: Springer International Publishing, 2017, pp. 29–36. [Online]. Available: https://doi.org/10.1007/978-3-319-67162-8_4

[11] S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi, "On the computation of multidimensional aggregates," in *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB '96)*. San Francisco, CA, USA: Morgan Kaufmann, 1996, pp. 506–521.

[12] C. Stolte, D. Tang, and P. Hanrahan, "Multiscale visualization using data cubes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, pp. 176– 187, 2003. [Online]. Available: https://doi.org/10.1109/TVCG.2003.1196005

[13] A. S. Maniatis, P. Vassiliadis, S. Skiadopoulos, and Y. Vassiliou, "Advanced visualization for OLAP," in *Proceedings of the 6th ACM International Workshop on Data Warehousing and OLAP (DOLAP '03)*. New York, NY, USA: ACM, 2003, pp. 9–16. [Online]. Available: https://doi.org/10.1145/956060.956063

[14] C. Ordonez, Z. Chen, and J. García-García, "Interactive exploration and visualization of OLAP cubes," in *Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP (DOLAP '11)*. New York, NY, USA: ACM, 2011, pp. 83–88. [Online]. Available: https://doi.org/10.1145/2064676.2064691

[15] X. Wang and B. Yi, "The application of data cubes in business data visualization," *Computing in Science & Engineering*, vol. 14, no. 6, pp. 44–50, 2012. [Online]. Available: https://doi.org/10.1109/MCSE.2012.17

[16] "SIEM for the modern SOC (Elastic Security)," https://www.elastic.co/siem/, accessed: 2021-07-27.

[17] H. Choi, H. Lee, and H. Kim, "Fast detection and visualization of network attacks on parallel coordinates," *Computers & Security*, vol. 28, no. 5, pp. 276–288, 2009. [Online]. Available: https://doi.org/10.1016/j.cose.2008.12.003

[18] T. Galkin and M. Grigorieva, "Parallel coordinates visualization in the elk stack," *Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). Part 2*, p. paper10, 2020. [Online]. Available: https://doi.org/10.51130/graphicon-2020-2-3-10

[19] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6. [Online]. Available: https://doi.org/10.1109/MilCIS.2015.7348942

[20] N. Moustafa, "The UNSW-NB15 dataset," 2015. [Online]. Available: https://researchdata.edu.au/unsw-nb15-dataset/1425943

[21] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016. [Online]. Available: https://doi.org/10.1080/19393555.2015.1125974

[22] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *Proceedings of the First IEEE Conference on Visualization (Visualization '90)*, 1990, pp. 361–378. [Online]. Available: https://doi.org/10.1109/VISUAL.1990.146402