Topological Data Analysis for Enhancing Embedded Analytics for Enterprise Cyber Log Analysis and Forensics

Trevor J. Bihl
Air Force Research Laboratory, USA
Trevor.Bihl.2@us.af.mil

Robert J. Gutierrez
Air Force Research Laboratory, USA
Robert.Gutierrez@afit.edu

Kenneth W. Bauer
Air Force Institute of Technology, USA
Kenneth.Bauer@afit.edu

Bradley C. Boehmke
84.51°, a Kroger Company, USA
Bradley.boehmke@8451.com

Cade Saie
United States Army, USA
Cade.m.Saie.mil@mail.mil

Abstract

Forensic analysis of logs is one responsibility of an enterprise cyber defense team; inherently, this is a big data task with thousands of events possibly logged in minutes of activity. Logged events range from authorized users typing incorrect passwords to malignant threats. Log analysis is necessary to understand current threats, be proactive against emerging threats, and develop new firewall rules. This paper describes embedded analytics for log analysis, which incorporates five mechanisms: numerical, similarity, graph-based, graphical analysis, and interactive feedback. Topological Data Analysis (TDA) is introduced for log analysis with TDA providing novel graph-based similarity understanding of threats which additionally enables a feedback mechanism to further analyze log files. Using real-world firewall log data from an enterprise-level organization, our end-to-end evaluation shows the effective detection and interpretation of log anomalies via the proposed process, many of which would have otherwise been missed by traditional means.

1. Introduction

e-Government cyber systems are enterprise level systems which encompass a variety of devices and standards due to the myriad of e-Government services offered. Additionally, these system are expected to be both secure and reliable in operations, accounting for and responding to a variety of cyber events [1] [2]. Enterprise cyber systems are all encompassing, which monitor and control access for any device that may use the network, from computers to Internet-of-Things (IoT) devices. Additionally, due to their scale and the sensitivity of the data and operations they support, enterprise cyber security involves many layers and efforts, such as encompasses multiple, very large local networks, which have their own devices, standards, and administration. Enterprise cyber security often includes forensics analysis and security or data fusion centers to detect emerging threats [3] [4].

Firewalls and intrusion detection and prevention systems (IDPS) are one line of defense in securing networks by identifying and stopping suspicious network traffic. These systems operate by evaluating traffic against rules to: 1) evaluate event type (e.g., invalid user password), 2) consider predicates on event attributes, (such as "user failed 3 times"), and 3) trigger actions if both the incoming traffic matches a particular type and the predicates are satisfied [5]. Rules can refer to specific IP addresses to block or be sophisticated and evaluate multiple events over a time period [6]. When rules are triggered by a suspicious event, firewalls and IDPS devices save the event as an entry in a log file with details of what preprogrammed rules were violated and how the event was handled [7]. While containing a lot of useful information, log files do not contain the packet data itself that led to a particular event. Thus, analysis is on contextual data while analysis of the transmitted data itself, as in [8], is not generally possible.

Logs from enterprise-level organizations can include thousands of events per second and millions per day [9]. This is expected to grow with increasing use of IoT endpoints [9]. Log files are one digital forensics data source to understand threats and discover new and emerging threats; however, log files analysis is often neglected in cyber security [9]. Since enterprise level cyber systems must provide accountability and look for emerging threats, cyber forensics of firewall and IDPS logs is a useful component of any enterprise-level cyber security operations [10]. However, the unstructured-ness of cyber logs in enterprises results in significant manual analysis for cyber forensics [3] [10].

The goal of this application is to improve general firewall digital forensics at the enterprise level through advanced analytical methods. Current approaches limit discovery by providing only an instance of results based on a given query. To remedy this, the authors present the application of topological data analysis (TDA) to log files, through which a feedback approach can be used for further analysis with descriptive/predictive statistics, and visualizations. TDA is an advanced numerical,

URI: https://hdl.handle.net/10125/63977
978-0-9981331-3-3

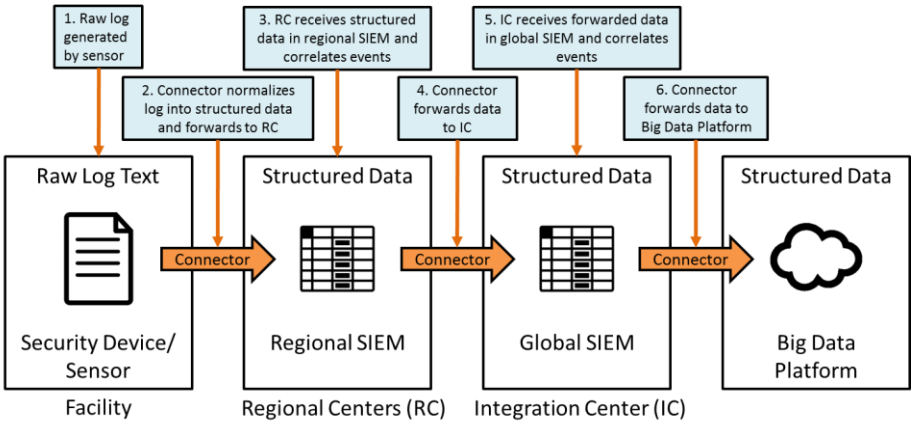


Figure 1. Generic representation of an enterprise-level data collection hierarchy, from [3]**.**

graph-based, and visual analytics approach that simultaneously considers similarity and shape of the data [11].

While TDA has been proposed for cyber network analysis [12] [13] [14], it has yet to be applied for firewall logs. The closest application of TDA to log analysis is that of Chow [15], who applied TDA to Windows event logs from the BRAWL (Blue versus Red Agent War-game Evaluation) project. However, the work in [15] considered synthetic cyber data, a constrained and experimental environment, and could not find direct value in applying TDA to cyber data. Additionally, Coudriau et al. [14] applied TDA to network address data, but the data was not high in variety as typically found in logs. This paper thus illustrates the first application of TDA to real-world collected firewall logs from a large enterprise system, as well as the incorporation of TDA into a big data analytics platform for batch analysis of cyber logs. Beyond these contributions, the results also address the limitations of using TDA mentioned in [15].

Analytics by themselves can be brittle; of interest in this paper is proposing and employing a combined experiential and analytical approach to log forensics through the data-driven, user-friendly embedded analytics platform. Herein, statistical and visual methods are combined into an embedded application. For this, the authors leverage their past work [3] in cyber analytics and employ a tabulated feature vector (TFV) approach to process log files and identify anomalies. In the TFV approach, log files are divided into temporal blocks and analysts are cued to blocks of interest. TDA extends this by looking at similarity across blocks, in addition to a further feedback loop process. The key contributions from this paper are:

- The first application of TDA to log analysis
- Development of a human-machine approach that is simultaneously temporal, similarity, and statistically-based that incorporates feedback
- Combination of statistical and visual methods for outlier/anomaly detection and interpretation
- An evaluation of the developed approach on real enterprise log files on data consistent with [3].

2. Background

Big data, such as network traffic, can be processed as batches or streams. Batches are collected and then processed whereas streams are analyzed in real-time or near real-time [16]. Firewalls and IDPS systems consider streaming data wherein rules are applied to network traffic [6]. In contrast, digital forensics considers aggregated data in a batch process to find threats and develop new firewall/IDPS rules [17].

A general enterprise level data collection approach is conceptualized in Figure 1. At enterprise systems, raw logs are normalized into a structured file at a given facility, then forwarded to a Regional Center (RC) which both collects data and defends against cyber threats [18]. At the RCs, a regional Security Information and Event Management (SIEM) device aggregates, correlates, monitors and generates alerts from the received data. Data is then sent and aggregated from multiple regional SIEMs and sent to a global SIEM. The global SIEM then connects to cyber analysis.

In Figure 1, the big data platform can be considered as a centralized database for managing and analyzing big data [3]. Here, both structured and unstructured data is collected with high volume, variety, and velocity, the 3 V's [19], from which data can be queried and analyzed. Of interest is looking for understanding and interpretation of events that would have triggered a log entry; such understanding can be gained by developing embedded analytics methods that analysts can run to analyze log files and provide interpretations.

2.1. Cyber Forensics Analytics

Fundamental to statistical data analysis, colloquially known as analytics, is the use of algorithms to find patterns [20]. Analytics methods range from supervised, with known classes/groups in the data (e.g. malignant and benign), to unsupervised, where classes/groups are unknown and need to be discovered (e.g. discovery of new groupings). Supervised analytics in cyber could include determining how similar observations are to known threats based on classifier algorithms. Unsupervised analytics, e.g. clustering, are especially relevant in cyber since the dynamic nature of cyber events means new threats are constantly emerging.

Problematically, normal behavior for cyber networks constantly changes and conservative approaches are often pursued which yield secure systems with high false positive detection rates [21]. Such systems could hamper authorized use of networks, furthering the importance of identifying actual threats. In analyzing log files, one is essentially interested in 1) detecting the anomalies within the anomalies, and 2) finding emerging threats via forensic analysis.

Two general approaches exist for cyber log forensics analytics: experiential/qualitative and analytical/ quantitative. Experiential cyber log forensics uses cyber analysts who rely on manual sorting and experientially gained knowledge to find possible threats for further investigation [3]. Such approaches are heavily effective and make use of the innate ability of humans to process large amounts of complex data [20]; however, big data challenges make it impossible to analyze all data effectively and novice analysts might miss events that veteran analysts would not [22]. Additional issues include the asymmetrical nature of cyber-attacks where attackers can focus on one approach while defenders must protect all systems from many different types of attacks, vulnerabilities, and threats [23].

To solve this problem, various analytical methods have been proposed to analyze logs [24]. Log analytics methods include both high level approaches, such as anomaly detection [25] [26], text analytics [27], dynamic rule creation [28], and event correlation [24], to the application of specific algorithms such as support vector machines [29], random forests [30], principal component analysis (PCA) and factor analysis (FA) [3]. However, analytics by themselves can be brittle. Of interest in this paper is proposing and employing a combined experiential and analytical approach to log forensics through a data driven embedded application.

2.2. Big Cyber Log Data

Due to the large size of enterprise networks and quantity of users, the data is of significant volume and emerging at high velocity; i.e. a big data problem. With a wide variety of devices at use in enterprise networks, each observation to the log can be highly variable from the others with the some logged fields being sparse due to differing approaches used at Regional SIEMs.

Traditionally, analysts employed experiential approaches to digital forensics where large log files are manually sorted to find anomalies to further investigate. However, inspecting numerous potential incidents on a daily basis is not scalable. The work in [3] extended the manual process to develop an embedded analytics tool that examines an entire log file and provides tools to find novelty, anomalies, and contextual meaning in a log. Notably, the authors in [31] recently proposed a similar process to that of [3]. Herein, we extend upon [3] with more advanced analytics as well as a feedback approach.

Due to sparsity, this research focuses on a reduced set of collected log fields, listed in Table 3. These fields were selected based on i) commonality between data fields logged by Regional SIEMS and ii) demonstrating statistical approaches to log file analysis without incorporating text mining techniques. Redundant fields from Regional SIEMs, e.g. *Device_Vendor* and *Device_Product*, were also grouped avoid confusion.

Table 3. Dataset Variables

Field Name	Description
Device Vendor	Company who made the device
Device Product	Name of the security device
Source Address	IP address of the source
Destination Address	IP address of the destination
Transport Protocol	Transport protocol used
Bytes In	Number of bytes transferred in
Bytes Out	Number of bytes transferred out
Category Outcome	Action taken by the device
Country_Name	Country of the source IP address

2.3. Tabulated Feature Vectors (TFVs)

Tabulated feature vector (TFV) extraction gets around difficulties handling mixed and sparse data and facilitates the use of statistical methods for log data. The method proposed by Winding et al. [32], and further applied in [3] takes log files and aggregates observations. The resultant feature vector is a count of occurrences of unique values and columns [32].

Figure 2 presents a conceptualization of the TFV process. In Figure 2a a conceptual raw log file is shown which include both categorical/text (Field A) and numerical (Field B) data columns. The process to create a TFV involves i) condensing text fields into columns of unique counts, and ii) summations of the original numerical values, which results in the TFV seen in Figure 2b. Naturally, this method can be extended to different fields, categories, etc., with the result being a TFV of length dictated by the complexity of the log file.

The process in Figure 2 would result in one observation, or row, if one applied it to an entire log file.

Thus, the TFVs are most useful if they divide the log files into blocks and then compute feature vectors for each block. Notably, variables such as the number of observations per block become important and one would want to employ a trial and error approach to finding the right number of samples per block. Whole log files can be divided into blocks by specifying an amount of time each block should represent or the number of observations each block represents. Intuitively, this can be accomplished using embedded analytics themselves.

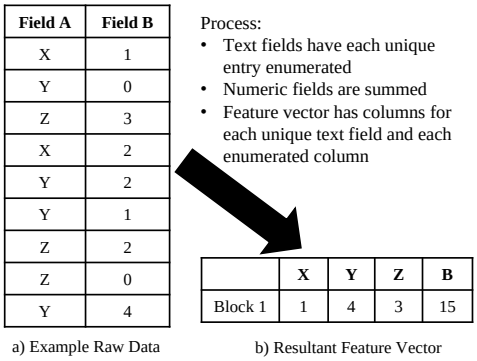


Figure 2. Generic feature vector creation process: (a) raw data; (b) resultant feature vector, from [3]

As a representation of the approach, the example log file under analysis was divided into 136 sequential time blocks, which enabled each block to contain 289 observations. TFVs were then computed for each block and the 9 data variables in Table 3 yielded 91 total features, including the top 10 source and destination IPs.

3. Topological Data Analysis

Topological Data Analysis (TDA), conceptualized in Figure 3, is a new unsupervised data mining approach wherein clustering, dimensionality reduction, and structure are considered together. Through this combination of methodologies, TDA provides additional understanding of relationships within data, to include both shape and similarity [11].

In TDA, one computes a topological model of the data using various distance measures and lenses. Practically, TDA extends upon this concept by binning observations based on similarity and then clustered based on shape and relationship. A graph-based relationship is then computed to show relative similarity between groupings of observations.

As a process and data mining approach, TDA employs the concepts presented in the Mapper algorithm [11] and is a generalization of hierarchical clustering [33]. Considering the input data as a matrix, TDA performs 3 steps via the Mapper algorithm [11]:

- data observations are placed into overlapping bins based on a metric
- clustering to group each bin
- realizing a network with vertices (clusters) and edges (intersections between clusters).

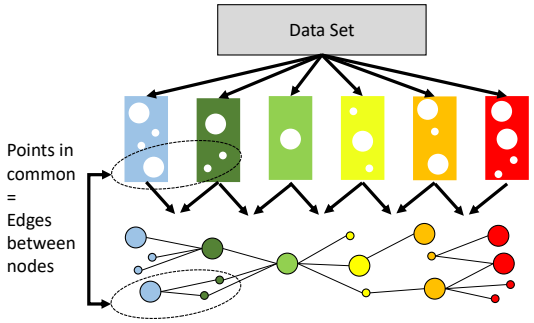


Figure 3. General Concept of TDA as implemented in the Mapper algorithm, adapted from [34]

As an algorithm, Mapper considers the inputs X as the data set, d as a metric on X, a scalar function $f:X \rightarrow \mathbb{R}$, and an overlapping interval covering (a_i, b_i) by $f(X)$ [11] [33]. The Mapper process considers a "preimage" $f^{-1}(a_i, b_i)$ and then uses the selected distance metric, d, to find pairwise distances between all observations [33], step 1 of Figure 3. Hierarchical clustering [35] is then performed to appropriately group points based on distance and other hyperparameter selections [33], step 2 of Figure 3. Next, step 3 of Figure 3, a graph-based topological model is constructed where vertices are clusters and clusters can be connected by an edge if these clusters share common observations.

The Mapper algorithm has multiple hyperparameters, which are consistent with other mapper implementations [33], these include:

- *metrics*, Table 1, the distance measure used for pairwise distance between all observations
- *lenses*, Table 2, clustering approach employed
- *resolution*, the number of intervals the data is portioned into
- *gain*, the number of edges in a network
- *equalization*, to ensure (or not) that all intervals contain the same number of points [33].

Practically, increasing resolution creates a TDA model with more nodes and increasing the gain increases the number of edges in a TDA model [33]. For this TDA application, the authors used the Ayasdi Workbench.

Table 1. General TDA Distance Metrics

Euclidean	Cosine	Angle
Variance or IQR Normalized Euclidean	Correlation	Norm Angle
Absolute, Norm, or Distance Correlation	Binary Jaccard	Haversine
Manhattan	Jaccard	Chebyshev

Table 2. General TDA Lenses

Gaussian Density	MDS
Isomap	Metric PCA
L1 Centrality	Neighborhood Graph
L-infinity centrality	Neighborhood
Local Linear Embeddings	SVD

4. Combined Statistical and Graphical Approach for Cyber Forensics

This work extends upon the work of [3] by incorporating TDA into the embedded analytics framework and also incorporating a feedback loop in the analytics process. For the embedded analytics, R was used, further illustrating the work of [36], which found that open source digital forensics can be effective. In operating embedded analytics apps, a cyber analyst can use the general process in Figure 4 whereby the preprocessing is automatically performed once a log file is selected, as discussed in [3]. From there, an analyst can use embedded applications to find blocks that have anomalous behavior. This section reviews and explains the process in Figure 4 and presents novel extensions of both applying TDA for log analysis, in addition to applying it in the analysis cycle of Figure 4 to enable graphical, similarity-based analytics of logs in a feedback approach in a larger analytics framework.

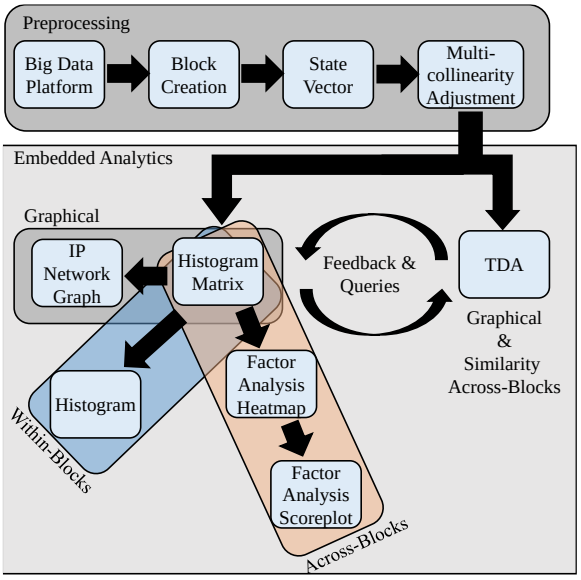


Figure 4. Multivariate and Graphical Approach to Firewall Log Anomaly Detection, extended from [3]**, to include the feedback and TDA for log analysis**

4.1. Preprocessing

Once log data is retrieved from the Big Data Platform, pre-processing, including TFV creation, is of interest prior to employing the analysis process. As described in [3], preprocessing involves:

1. Dividing the log file into time-regional blocks
2. Creating tabulated state vectors
3. Performing data quality checks and adjustments for multi-collinearity

Figure 5, from [3], shows an example of a block created which consists of the categorical variables being labeled as factors and the numerical variables being labeled as numeric. In Figure 4, the counts are the number of levels associated with a categorical variable then denote the number of unique entries.

```
Block 14                                    289 Obs. Of 8 variables
    transportProtocol:  Factor w/ 4 levels "ICMP", "No Protocol", …:  2 2 2 2
    bytesIn:        int     0       0       0       0       0       0 …
    bytesOut        :       int     0       0       0       0       0       0 …
    categoryOutcome         :       Factor w/ 4 levels "/Attempt", "/Failure", … : 4 4 4
    Country_Name            :       Factor w/ 31 levels "Country 1", "Country 2", … : 31
    Device_Name :           Factor w/ 19 levels "Device 1" , "Device 10", …       : 11  11
    sourceAddress_mask  :           Factor w/ 3376 levels "Source IP 1", … :       2  2 2
    destinationAddress_mask     :           Factor w/ 4786 levels "Destination IP 1",…
```

Figure 5. Sample time block

One example can be seen in the 6th line of Figure 5, *Device_Name*, which has 19 unique categories, or levels, in the example log file. This indicates that logged entries from 19 different devices were found in this log file. These 19 devices will then become 19 different columns in the tabulated state vectors, per Section 2.3. Logged IP addresses are considered to only the top 10 observed IP addresses (both source and destination) and the associated counts of these observed IP addresses are recorded. These vectors are then aggregated into a single matrix, the state vector matrix.

Before analytics are applied, the columns are checked for multicollinearity which could result in unreliable mathematical results. In this step, columns with multicollinearity issues, e.g. matrix singularity, rank deficiency, and strong correlation values, are identified and removed [3].

4.2. Embedded Analytics

Once the data has been preprocessed, an analyst can consider the embedded analytics suite and look for anomalies, threats, and emerging behavior. For this step, there are two general paths:

1) the Histogram Matrix (HMAT) that served as the foundation of [3]
2) Topological Data Analysis (TDA) and its feedback ability with HMAT, as proposed in this paper.

With these methods, HMAT is used to enable graphical, within-blocks, and across-blocks analysis of blocks, whereas TDA is used to look across blocks for similarity and enable feedback and queries of blocks for HMAT analysis. By incorporating TDA, embedded analytics can be expanded to examine across the state matrix.

4.2.1. Histogram Matrix (HMAT). HMATs, a general extension of heatmaps, are used consistent with [3] where each cell's value is a colorized circle with the color representing one value and the size of the circle representing another value. HMAT for cyber log files visualize anomalies by coloring cells based on their squared Mahalanobis distance, a general outlier detection method, and provide an understanding of the relative magnitude of the Mahalanobis distances based on a scaling factor, the Breakdown Distance (BD) [3].

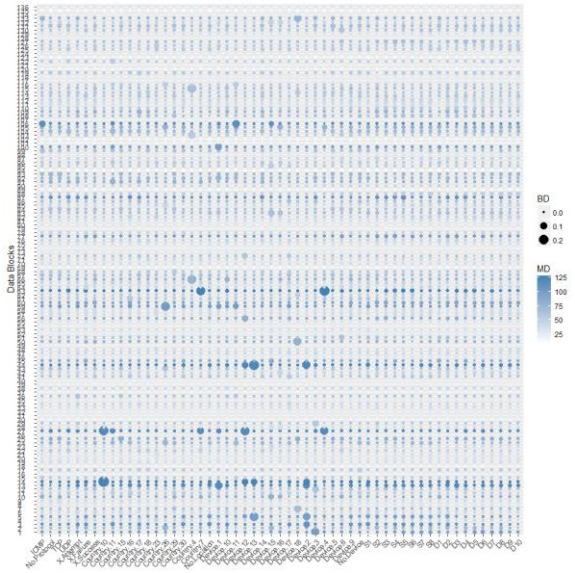


Figure 6. Histogram Matrix (HMAT)

HMAT for cyber log analysis uses the color to represent the raw value of how anomalous a specific column's value is in a block, color determined from Mahalanobis distances, and a further understanding of the relative BD value, size of the circle. Mahalanobis distance, a multivariate anomaly/outlier detection method, compares each observation, *x*, via

$$MD = \sqrt{(x - \bar{x})C^{-1}(x - \bar{x})} \quad (1)$$

with $x = (x_1, \dots, x_p)$, a vector of p observations, $\bar{x} = (\overline{x_1}, \dots, \overline{x_p})$, a the mean vector of the data, and C^{-1} being the inverse data covariance matrix [35]. Based on their raw scores, *MD* values can be used to find anomalies by relative magnitudes.

However, *MD* does not directly provide a rationale for what is or is not an anomaly; thus, we proposed the use of BD in [3], which is computed as,

$$BD_i = \left| \frac{(x_i - \overline{x_i})}{\sqrt{C_{ii}}} \right| \quad (2)$$

where x_i is a given variable, $\overline{x_i}$ is the mean of the variable, and C_{ii} is the variance of x_i, which are the diagonal values of *C*. BD scales the distance of each variable from the mean by their variance which thus indicates the relative contribution of each variable to a given *MD* score [3].

When considering the example dataset with HMAT, one sees Figure 6. Here, block numbers (y-axis) and column names (x-axis) are abstractly represented to enable a big picture view to find potential anomalous behavior. From these results, a user would select specific blocks for further analysis. Briefly, one would look for darker shaded circles, indicating a column in a block contains anomalous behavior for this log file, and then look at the size of the circles whereby larger circles indicate that a given variable is increasing the MD for that particular block. As discussed in [3], in Figure 6, the clearest anomalies are blocks 14, 27, 44 and 63.

4.2.2. Within-Blocks Analytics. Within-blocks analysis considers spillage of top attributes across neighboring time blocks to understand how particular columns are anomalous given the neighboring blocks. For this approach, the top five columns are examined via histograms to: 1) observe the data columns that cause the block to be anomalous and 2) examine any regional temporal relationship between the top five anomalous columns and the neighboring time blocks [3]. The example in Figure 6 shows Country 10 and Device 12 as appearing as anomalous only in Block 14; and Device 2 appearing only in Block 14 and adjacent blocks. Thus, these features might indicate a particular event in or around Block 14 associated with these attributes.

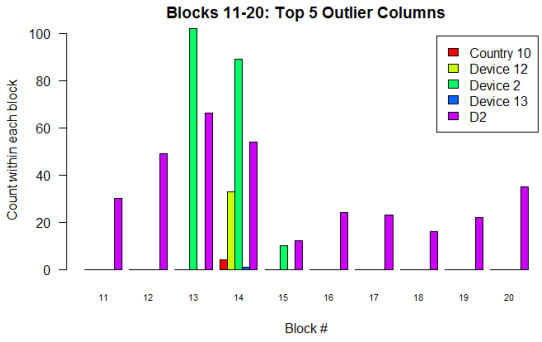


Figure 6. Block 14 Within-Blocks Analysis with Top 5 Anomalous Columns per Breakdown Distance

4.2.3. Across-Blocks Analytics. Basic across blocks analytics involves considering statistics for the entire state vector matrix. For this PCA or FA are used to aggregate data features and reduce dimensionality [3] [35]. PCA is a linear transformation approach that projects the original data by the eigenvectors of the data covariance matrix to explain variation in the data; FA extends upon PCA with a rotation, varimax used herein, of retained components, to better represent the data [35].

The dimensionality assessment process (number of components to retain) used herein was Horn's curve for to select *k* unrotated (PCA) features and then Kaiser's Index of Factorial Simplicity was used to assess the quality of the FA rotation [3]. When considering the example state vector matrix through this process, the dimensionality of columns is reduced from 91 variables to 6 retained components/factors.

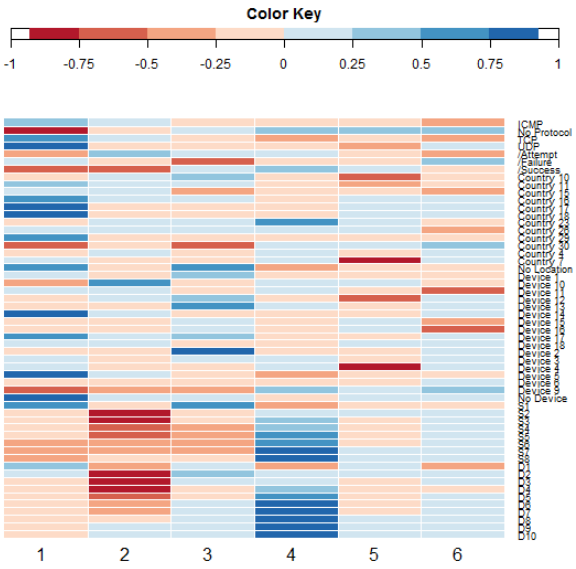


Figure 7. Heatmap of Loadings Matrix from Example Data with red = negative, blue = positive.

The result of across-blocks analytics is both a heatmap of the loadings and *x*-vs-*y* factor plots. A loadings heatmap, Figure 7, present a correlation matrix of the original data and transformed factors to provide an understanding of variable contributions to factors. As an example analysis, factor 1 in Figure 7 shows that devices and protocols were highly correlated with IP addresses from Countries 16, 17, 18, and 29. From here, analysts can incorporate contextual information, e.g. possibly linking activity in this log file to known TOR exit nodes. Further interpretation of Figure 7 shows a possible connection between rules triggered in Device 4 and Device 13 and IPs from Countries 7 and 10, which reinforces the interpretation of Figure 6 where Country 10 and Device 13 are seen active around Block 14.

Factor plots, Figure 8, consider the factor scores as plotted in a pairwise manner for the 6 retained factors of Figure 7. While the most variation is explained in decreasing amount by each retained component/factor, with this data, possible anomalous blocks are not apparent until examining rotated scores 3 and 5 with Based on Figure 8, potential anomalous time blocks, are blocks 27, 63, 14 and 44. These can then be evaluated further with HMAT, histograms, or graphical methods.

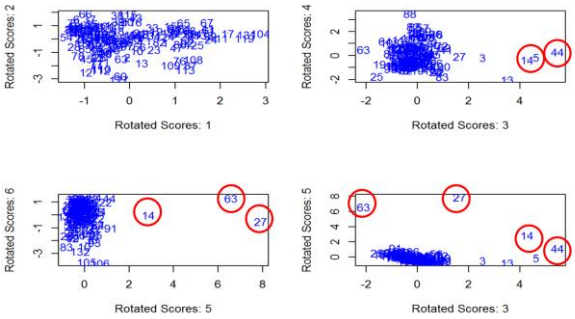


Figure 8. Factor plot using varimax rotated FA scores and identified outlier blocks

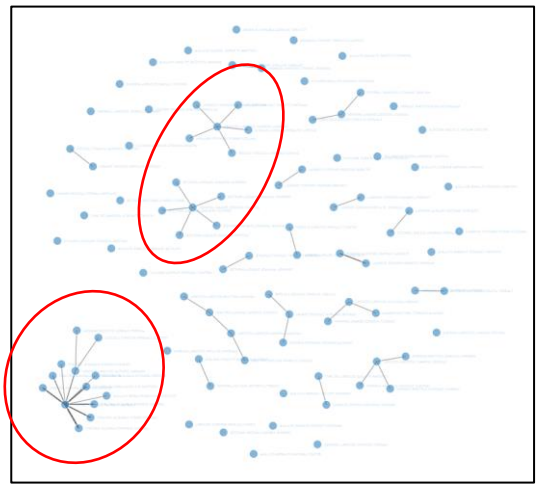


Figure 9. IP Network Graph of Block 14 with detected port scan

4.2.4. Graphical Analytics. Graphical analysis of the entire state matrix or a specific block are a further direction to interpret a log file. IP network graphs, e.g. Figure 9, visualize connections between source IP addresses and the destination IP(s) within a dataset allowing one to look for visual cues to understand possible interactions and trends. While Figure 9 which mostly shows singletons, i.e. single IP addresses, or simple pairs, two groups are of interest, an encircled in red. Both groups show clusters of source IP addresses linking to multiple destination IP addresses. A top level interpretation of these clusters is that a computer was performing a port scan, possibly attempting to fingerprint the network, and thus an analyst can revisit this time block to understand this event further.

4.2.5. TDA for Graphical and Similarity Across-Blocks Analytics. Missing in the process described above, and in [3], is an understanding of similarity across all blocks and feedback to repeatedly discover and query. For example, the HMAT might indicate which blocks have higher activity of Country 10, but not

what blocks are largely similar across this and all columns. Additionally, with this information of similarity, a user could select further blocks to refine and redo the analysis discussed in 4.2.1-4.2.4. Incorporating TDA can address both challenges.

TDA is performed on the entire state matrix of TFVs and multiple topological models are constructed with different combinations of lenses, metrics, resolutions, and gains. One example topological model, which will be discussed further, is presented in Figure 10a. Due to the breadth of options for TDA, discussed in Section 3, exhaustive combinations of metrics and lenses was not considered. For the model in Figure 10a, a cosine metric was used along with kurtosis and variance lenses. Cosine distances can be useful because they are translation variant but scale invariant, in contrast to Euclidean distances [37].

In Figure 10 we see four groupings of nodes with three of these groupings show connections, and thus similarities, of nodes; the fourth grouping is of singletons, which are not similar to other nodes based on this selection of metrics and lenses. Coloring is also possibly by the features used to generate the models, with the default coloring (10a) being a divergent color scheme (blue to red) based on block number (blue representing mostly low numbered blocks and red indicating a high concentration of higher numbered blocks). In the middle grouping, we see a large concentration of such dark red nodes in the middle and then there is a gradient as one moves left or right across this grouping.

Colored by Country 10 (Figure 10b), we see mostly blue nodes, indicating few if any events from Country 10 except in the middle grouping which has high concentrations of Country 10 activity on both the far right and left ends. In part, Country 10 thus separates this group from the rest of the groups. Revisiting Block 14, we find it in right most node of the middle grouping.

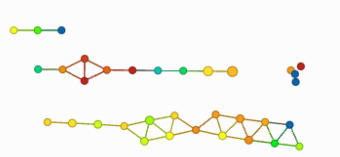
Colored by No Protocol (10c), we see the frequency of No Protocol events. A general separation is seen for all three groups and we can further investigates the FA loadings plot of Figure 7. The upper grouping is largely orange, indicating an average amount of No Protocol events. The lower grouping is mostly yellow, green, and blues; indicating few occurrence. However, the middle grouping presents a color gradient that moves from left to right in increasing No Protocol concentration.

4.3. Feedback with TDA

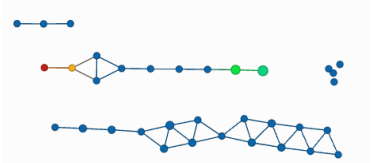
To further investigate what gathers together, we can click on a node and then find what blocks it contains. This functionality moves beyond capabilities available in Ayasdi Workbench and incorporates its functionality into the Big Data Platform. Here, we can enable analyst driven feedback whereby analysts can take the knowledge from the topological models and then examine each node within this model with the HMAT tool. From here, the process throughout Section 3 and 4 can be repeated to down-select blocks and find more novelty and understanding of behaviors.

An example is presented in Figure 11. Here, HMATs are presented for the three rightmost nodes of the middle grouping. These HMATs now enable one to find anomalies and trends within these subsets. Considering the rightmost node in $HMAT_1$ (bottommost HMAT in Figure 11), which contains the Block 14 discussed above, is found together with other potential blocks. Of particular interest appears Block 48, which has high levels of activity from Country 27, as well as the adjacent Block 49, which also appears. A set of adjacent blocks with strong anomalous scores also appears in the next node, $HMAT_2$ in the upper right of Figure 11, which shares Blocks 48 and 49 with the rightmost node.

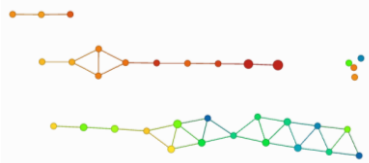
When we move over to the 3rd node from the right, we see $HMAT_3$ at the top left of Figure 11. Here, only 4 blocks are contained within this node. Most of these blocks do not contain much anomalous activity, but Block 16 appears to be active across most columns. Since the other 3 blocks within this node are similar, we might first investigate Block 16 and also look for similar, but lower, activity in the Blocks 4, 37, and 127 which appear similar. Interestingly, $HMAT_2$ and $HMAT_3$ both overlap on Block 16, but Block 16 did not appear notable in $HMAT_2$. While Block 16 would have likely been ignored just looking at the HMAT in Figure



a) Coloration by block number, with a divergent color scheme (blue indicating lower numbered, and red indicating higher numbered, blocks)



b) Coloration by Country 10 with blue colors representing low (or no) counts and red colors indicating high counts.



c) Coloration by No Protocol with blue colors representing low (or no) counts and red colors indicating high counts.

Figure 10. Custom topological model using Metric: Cosine, Lens 1: Approximate Kurtosis, Lens 2: Variance. Resolution: 30 (both lenses), gain: 2.50 (both lenses), with neither lens equalized.

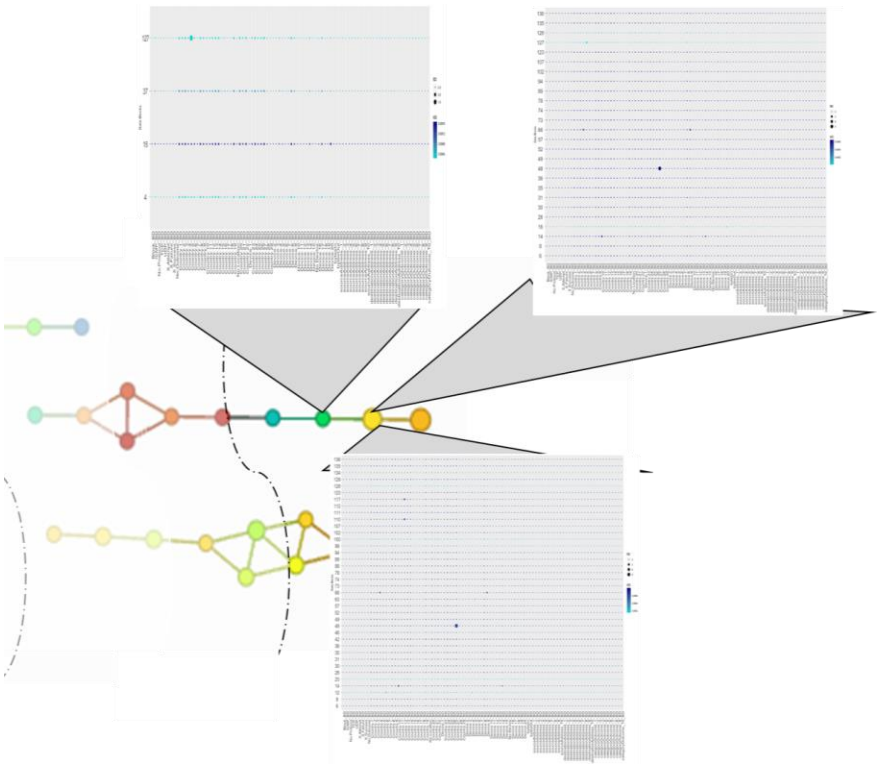


Figure 11. Topological Model from Figure 10a with incorporated HMAT visualizations for selected nodes.

6, the addition of TDA and feedback has enabled further discovery of potentially interesting patterns.

5. Conclusions

The authors presented a systematic experiential and data driven approach to cyber log analysis. For cyber log analytics, the authors employed a tabulated feature vector (TFV) approach which extracted numerical, and non-sparse, data from sparse log files. With the TFVs, the authors developed a human-machine approach that was simultaneously temporal, similarity, and statistically-based while also incorporating a feedback mechanism. This paper also provided the first application of Topological Data Analysis (TDA) to log analysis. In particular, it was seen that TDA enabled one to look across all blocks to find similarity of events for further interpretation. By enabling one to repeated query the dataset through the feedback mechanism, the authors showed how one can find anomalous behavior to examine which would have been missed by approaches in literature. Finally, this research illustrates how to develop an embedded analytics approach for anomaly detection, which is reproducible and generalizable to other datasets, and other types of log files, which have different data features.

Straightforward extensions include exploring the application of the presented method to other types of log files. Future work can also involve extending the TFV process to include sparse data fields, which are currently removed due to multicollinearity issues. Further future work can consider further automation of the general process in Figure 4 to bring human analysts; which could be accomplished by automating the novelty detection of TDA data products. Additional research can involve TDA in general, including better understanding the TDA hyperparameter space, comparison with other clustering methods, and selecting appropriate analytics given TDA outputs on the shape of the data.

6. References

[1] K. Harrison and G. White, "A taxonomy of cyber events affecting communities," *Hawaii International Conference on System Sciences*, pp. 1-9, 2011.

[2] A. Zobaa and T. Bihl, "Security Methods for Critical Infrastructure Communications," in *Big Data Analytics in Future Power Systems*, CRC Press, 2018, pp. 85-106.

[3] R. Gutierrez, et al., "Cyber anomaly detection: Using tabulated vectors and embedded analytics for efficient data mining," *Journal of Algorithms & Computational Technology*, vol. 12, no. 4, pp. 293-310, 2018.

[4] S. Schinagl, et al., "A framework for designing a Security Operations Centre (SOC)," *Hawaii International Conference on System Sciences*, pp. 2253-2262, 2015.

[5] T. Morris, et al., "Deterministic Intrusion Detection Rules for MODBUS Protocols," *Hawaii International Conference on System Sciences*, pp. 1773-1781, 2013.

[6] J. Stephen, et al., "Distributed real-time event analysis," *IEEE International Conference on Autonomic Computing*, pp. 11-20, 2015.

[7] H. J. Liao, et al., "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16-24, 2013.

[8] L. Zhang and G. White, "Analysis of payload based application level network anomaly detection," *Hawaii International Conference on System Sciences*, pp. 1-10, 2007.

[9] D. Davis, "Cybersecurity 101: The criticality of event logs," *CSO Online*, 21 Nov. 2016.

[10] E. Pilli, et al., "Network forensic frameworks: Survey and research challenges," *digital investigation*, vol. 7, no. 1-2, pp. 14-27, 2010.

[11] G. Singh, et al., "Topological methods for the analysis of high dimensional data sets and 3d object recognition," *SPBG*, pp. 91-100, 2007.

[12] A. Karpistsenko, "Networked Intelligence: Towards Autonomous Cyber Physical Systems," arXiv preprint arXiv:1606.04087., 2016.

[13] J. Navarro, et al., "Huma: A multi-layer framework for threat analysis in a heterogeneous log environment," I*nternational Symposium on Foundations and Practice of Security*, pp. 144-159, 2017.

[14] M. Coudriau, et al., "Topological analysis and visualisation of network monitoring data: Darknet case study.," *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1-6, 2016.

[15] Y. Chow, *Application of Data Analytics to Cyber Forensic Data*, Worcester Polytechnic Institute: BS Thesis , 2016.

[16] W. Lin, et al., "Streamscope: continuous reliable distributed processing of big data streams," *13th Symposium on Networked Systems Design and Implementation*, pp. 439-453, 2016.

[17] J. Sammons, *The basics of digital forensics: the primer for getting started in digital forensics*, Elsevier, 2012.

[18] G. Van Vleet, "2d Regional Cyber Center Opens," 30 Oct. 2013. [Online]. Available: https://www.army.mil/article/114105/. [Accessed 1 Jan. 2016].

[19] T. Bihl, et al., "Defining, understanding, and addressing big data.," *International Journal of Business Analytics (IJBAN)*, vol. 3, no. 2, pp. 1-32, 2016.

[20] A. K. Jain, et al., "Statistical pattern recognition: A review.," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 4-37, 2000.

[21] M. Zamani and M. Movahedi, "Machine Learning Techniques for Intrusion Detection," arXiv preprint arXiv, vol. 1312.2177, pp. 1-11, 2013.

[22] R. G. Abbott, et al., "Log analysis of cyber security training exercises," *Procedia Manufacturing*, vol. 3, pp. 5088-5094, 2015.

[23] J. R. Goodall, et al., "Supporting intrusion detection work practice," *Journal of Information System Security*, vol. 5, no. 2, pp. 42-73, 2009.

[24] M. R. Grimaila, et al., "Design and Analysis of a Dynamically Configured Log-based Distributed Security Event Detection Methodology," *The Journal of Defense Modeling and Simulation*, vol. 9, no. 3, pp. 219-241, 2011.

[25] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. 2, pp. 222-232, 1987.

[26] M. Ahmed, et al., "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016.

[27] C. Suh-Lee, et al., "Text mining for security threat detection discovering hidden information in unstructured log messages," *IEEE Conference on Communications and Network Security (CNS)*, pp. 252-260, 2016.

[28] J. Breier and J. Branišová, "A dynamic rule creation based anomaly detection method for identifying security breaches in log records," *Wireless Personal Communications*, vol. 94, no. 3, pp. 497-511, 2017.

[29] A. Lazarevic, et al., "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," *SIAM Conference on Data Mining*, pp. 25-36, 2003.

[30] J. Z. J. Zhang and M. Z. M. Zulkernine, "Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection," *IEEE International Conference on Communications*, pp. 2388-2393, 2006.

[31] A. Majeed, et al., "Near-miss situation based visual analysis of SIEM rules for real time network security monitoring," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1509-1526, 2019.

[32] R. Winding, et al., "System anomaly detection: Mining firewall logs," *Securecomm and Workshops*, pp. 1-5, 2006.

[33] M. Pirashvili, et al., "Improved understanding of aqueous solubility modeling through topological data analysis," *Journal of cheminformatics*, vol. 10, no. 54, pp. 1-14, 2018.

[34] M. Offroy and L. Duponchel, "Topological data analysis: A promising big data exploration tool in biology, analytical chemistry and physical chemistry," *Analytica chimica acta*, vol. 910, pp. 1-11, 2016.

[35] W. Dillon and M. Goldstein, *Multivariate analysis methods and applications*, Wiley, 1984.

[36] D. Manson, et al., "Is the open way a better way? Digital forensics using open source tools," *Hawaii International Conference on System Sciences* , pp. 1-10, 2007.

[37] T. J. Bihl, et al., "Cyber-Physical Security with RF Fingerprint Classification through Distance Measure Extensions of Generalized Relevance Learning Vector Quantization," *Security and Communication Networks*, 2019.