Semantic feature selection for network telemetry event description

Thomas Feltin, Parisa Foroughi, Wenqin Shao, Frank Brockners, Thomas Heide Clausen

Semantic feature selection for network telemetry event description

Thomas Feltin*†, Parisa Foroughi ‡†, Wenqin Shao†, Frank Brockners † and Thomas H. Clausen*
*École polytechnique
{thomas.feltin,thomas.clausen}@polytechnique.edu
†Cisco Systems
{wenshao,fbrockne}@cisco.com
‡Telecom Paris
parisa.foroughi@telecom-paris.fr

***Abstract*—Model driven telemetry (MDT) enables the real-time collection of hundreds of thousands of counters on large-scale networks, with contextual information to each counter provided in the telemetry data structure definition. Explaining network events in such datasets implies substantial analysis by a domain expert. This paper presents an semantic feature selection method, to find the most important counters which describe a given event in a telemetry dataset, and facilitate the explanation process. This paper proposes a metric for estimating the importance of features in a dataset with descriptive feature names, to find those that are most meaningful to a human. With this estimation, this paper presents a cross-entropy based metric describing the quality of a selection of counters, which is combined with the data behavior to define an optimization goal. The computation of optimal selections distills intelligible and precise selections of counters with adjustable verbosity, and describes events with a few selected counters outlining the root cause of network events.**

***Index Terms*—Network management, Decision support, Selection process**

I. Introduction

Understanding the state of a router commonly involves a domain expert who interprets a selected set of operational data retrieved through SNMP or CLI. The emergence of model driven telemetry (MDT) further allows the automated and frequent retrieval of all the available operational counters on a router, in a semantically consistent way through a collection of YANG modules [1]. Routers in large-scale networks generate hundreds of thousands of individual operational counters, each describing a particular aspect of device behaviour. In network telemetry datasets of such dimensionality, distilling the information which best describes an event can be challenging. Events refer to any network or hardware-related events which cause the global state of the router or network to change, *e.g.*, network loops, black holes, interface failures, memory leaks. Because of the dependencies between the different operational counters, the majority of counters are highly impacted in value when events occur in the network. For example, as shown in Figure 1, the dataset [2] used in this paper contains 6622 individual counters, four thousand of which change in value when an interface failure occurs. Among all the thousands of counters changing in value, only a few describe the cause of the event. The majority of counters are either (i) frequently changing in value independently of the event, or (ii) only describing the consequences of the actual event. An interface failure will cause packet losses, route re-convergence, TCP connection changes, traffic changes, etc. which constitute the majority of changing counters, while the counters describing the actual root cause, *e.g.*, interface counts, will only be a few instances among the several thousands. An approach that takes all available counters into account and distills those which are most descriptive of an event in an automated data-driven way is still missing.

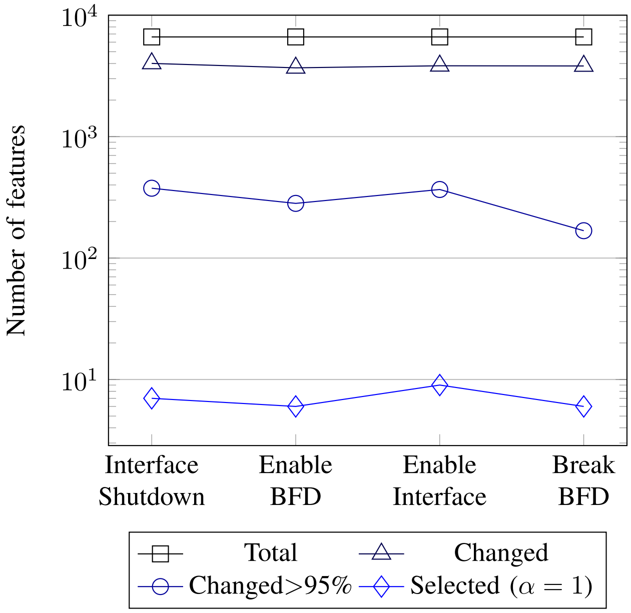

Figure 1. Number of selected features with $\alpha = 1$ compared to the original number of features and the number of changing features (logarithmic scale) around given events in the dataset [2]. ”Changed” describes the number of features which change in value around the event, and ”Changed>95%” are the ones which see a change of more than 95% of their absolute value.

This paper proposes a method to find an intelligible selection of operational counters, *i.e.*, that can be understood and analyzed by a human, which best describes an event in

978-1-7281-4973-8/20/$31.00 © 2020 IEEE

network telemetry datasets. Labelling the data in this context implies the annotation of all the counters in the dataset by a domain expert. Not only would this process imply annotating thousands of counters for every possible event, it would also be too subjective. Every domain expert will look for different counters in their diagnosis, which implies that counters don't have an absolute importance value, *i.e.*, the ground truth can't be defined in this problem space.

The problem is formulated as a feature selection problem [3], *i.e.*, the extraction of the most important features in a dataset. The objectives of feature selection are further revised to fit this problem space. While the literature focuses on preserving the overall information contained in the original dataset, this method generates selections which are (i) descriptive of an event, *i.e.*, contain counters which see a significant change in value, and (ii) intelligible to a network engineer.

While the reaction of a counter to an event can be quantified from data behaviour, the intelligibility of a counter in a dataset, *i.e.*, how useful it is in helping a human explain the event, is strictly defined by domain knowledge. To compare the two notions, this paper presents a metric to quantify the abstract notion of how intelligible a counter is in the dataset, based on the counter's rareness in the feature set. This metric is extended into a cross-entropy based metric to describe the overall intelligibility of a selection of counters. The method then combines this estimation of intelligibility (domain knowledge) with a score for how strongly the features react to a change (data behaviour) to define an optimization score. This score allows the computation of *optimal* selections to help operators explain network events.

The remainder of this paper is organized as follows: Section II presents the related work on feature selection methods. Section III describes a method for estimating the relative importance of counters in the data based on the feature names only, and a measure of cross-entropy. Section IV presents the selection score and the intuition behind its definition. Finally, Section V describes the experiment setup and results for this score on a network telemetry dataset.

II. Related Work

A *feature* describes a measurable property of the object described in a dataset. The features are the columns of the dataset, *e.g.*, the counters in MDT data.

In the literature, *explanation* refers to the identification of the input features which contribute most to a model's decision [4], [5]. Although this paper presents a method with the same objective, it does not consider any model, but rather isolated events at a given moment in a multivariate time-series. In this study, the process is entirely data-driven, and is rather considered as a features selection problem.

Feature selection is the process of selecting the most important features in a dataset, in order to remove the irrelevant or redundant features [3]. Unsupervised feature selection performs this selection without the use of labels.

Unsupervised feature selection methods can be categorized in three groups [3], [6]: wrapper, filter, and hybrid methods. Wrapper methods [7] select the subset of features which optimizes the result of a specific clustering algorithm, and have shown to be computationally expensive for high dimensionality problems [3]. Filter methods rely only on properties of the data to assign a relevance score and rank each feature in the dataset. The score of each feature can either be computed in isolation (univariate filter method), or in conjunction with the other features in the dataset (multivariate filter methods).

The first filter method relying on information theory is sequential backward selection for unsupervised data [8], and is based on a measure of entropy with regard to the distances between features. SVD-Entropy [9] also uses the contribution of every feature to the entropy of the dataset (CE) to rank the features and find the subset with highest entropy. Similarly, [10] uses the concept of Representation Entropy to capture the amount of information contained in a selection. The idea behind [8]–[10] is that a high entropy translates to a balanced cluster structure, which implies that the features best represent the data. Multivariate methods such as Feature Selection using Feature Similarity (FSFS) [11] use statistical dependencies to further remove redundancy from a selection.

What the methods discussed above have in common is their principal objective: finding the most relevant features, *i.e.*, features which contribute most to preserving the manifold structure of the original data, or features with highest or lowest correlation with the other features in the dataset [3]. The problem space of event description introduces a new constraint on the selected subsets, since they must (i) describe an event, and (ii) be intelligible enough for a human to interpret. In this problem space, relevant features are not only those relevant with respect to the information contained, but also those most affected by the change, and those with most significance to a network engineer. The objectives in this study differ from those of unsupervised feature selection, and are modified to include the two notions mentioned above.

In order to offer a complementary analysis to what is contained in the raw data, the assumption is taken that, like in most telemetry datasets, features are labelled with a feature name which gives some information on the feature and/or the group they belong to. Raw feature names can be used as an indication of what functionality or what subgroup the feature refers to, *i.e.*, its semantic meaning. This study also differs from the literature in the sense that it uses feature names to help selection, in addition to the data behavior.

In that respect, methods from information retrieval in text provide tools to exploit the feature names, *e.g.*, TF-IDF [12], which estimates the importance of words in a document with respect to a corpus of documents, based on the compared occurrences of the words.

III. Deriving information from feature names

As described in Section II, the notion of relevance in the literature is most often linked to a measure of information contained in the resulting selection of features. In this paper, the main objective is that a selection be as intelligible as possible for understanding an event. The ideal selection in this problem space is one that distills the counters that are both impacted by the event, and meaningful to an operator.

Since this notion of intelligibility is abstract and strictly depends on domain knowledge, this paper proposes an approximation, inspired by the computation of TF-IDF [12]. This estimation is based on the observed correlation between the occurrence of features in a dataset and their meaningfulness to a network engineer. Rare features in a dataset, *i.e.*, in cases where few features describe a given property, are observed to be more meaningful when selected than the most frequent ones. For example, BFD session counts (2 occurrences out of 6622 in the dataset [2]) are more meaningful to an operator than one of the many bytes sent counters on the router's interfaces (570 occurrences). Rareness is defined as how many of the same types of features exist in the entire dataset. This approximation allows the method to leverage the contextual information derived from the feature names, and offers a complementary analysis to what can be extracted from the data behavior.

To quantify rareness, frequencies are defined on a space describing the whole set of feature names, where a rare feature name has a low frequency value, and a highly occurring feature name has a higher frequency value.

A. Example distribution over the feature names

The definition of frequencies in this section must describe the rareness of feature names in a dataset, *i.e.*, frequently occurring feature names must have a higher frequency value than lower occurring ones. Any method for generating frequencies with such properties is valid – this paper will use the method described below.

A simple method for using descriptive feature names to quantify the rareness of a feature is to simply consider their occurrences in the dataset.

In the case of MDT, the feature names are the sensor paths [13]. With YANG, the feature names are part of a 3-layered hierarchical name space, and a specific name is a branch in that tree.

$$\underbrace{\text{tcp\_node\_statistics}}_{\text{token type 1}} : \underbrace{\text{interface\_1}}_{\text{token type 2}} : \underbrace{\text{bytes-sent}}_{\text{token type 3}}$$

Sensor paths can be parsed to extract three components: a module name, a key value array and a leaf name, as described in [1], and the rareness of a feature name is defined as the frequency of these individual components. These three components are labelled as *tokens*, each token being an instance within a *token type*. In the example above, sensor paths consist of three token types, and can be parsed into three tokens: ”tcp_node_statistics” is a token within the token type of module names, ”interface_1” is a token within the token type of key value arrays, and ”bytes-sent” is a token within the token type of leaf names.

More generally, feature names can be parsed when their format is consistent, in order to make token types correspond to precise attributes. This distinction between module name, key value array, and leaf name, can be generalized to the distinction of K token types as the different attributes parsed in a feature name (in this case, $K = 3$). Within each type, the rareness of a feature name is defined as the frequencies of its tokens within their type in the set (giving K frequency values for a single feature name).

For a token type $0 < k \leq K$, T_k is the total number of unique tokens found among token type k in the entire set of feature names. For $0 < i \leq T_k$, $t_{k,i}$ the i-th individual token among the tokens of type k, and $n_{k,i}$ is the number of times token $t_{k,i}$ appears as the k-th token type in a feature name. In other words, this value is counting the occurrences of every unique token among the tokens of the same type. Finally, for $0 < k \leq K$ the frequencies $\{p_{k,i}\}_{0<i\leq T_k}$ are defined as $p_{k,i} = n_{k,i}/N$, where N is the total number of features in the set.

For example, the frequency associated with the module name `tcp_node_statistics` (token type 1), the key value array `interface_1` (token type 2), or the leaf name `bytes-sent` (token type 3), will be the number of times the token appears divided by the total number of features in the dataset, giving one probability distribution p_k for each token type (3 in this case).

For each token type k, $p_{k,i}$ is a measure for how rare token $t_{k,i}$ is, within the token type k, and estimates how meaningful $t_{k,i}$ is to a network engineer (low values of $p_{k,i}$ translate to the token $t_{k,i}$ being most meaningful).

IV. Method

This section presents the feature selection method. For $0 < k \leq K$, $q_k = \{q_{k,i}\}_{1\leq i\leq T_k}$ refers to the distributions of tokens defined on the full set of feature names, and $p_k = \{p_{k,i}\}_{0<i\leq T_k}$ refers to the distribution of tokens in the subset selected by the feature selection method.

A. Quantifying the selection quality

Measuring the quality of a feature selection implies defining a goal metric for a selection. In the problem space described in Section I, the objective is to produce selections that are both intelligible and impacted by the event. The use of entropy as a goal metric in the literature translates to a balanced cluster structure in the data [3]. In this paper, an ideal selection is one that is *specific*, *i.e.*, an unbalanced cluster structure. The more specific to a given functionality, or to a given element of hardware, the more information a selection will provide to an operator, and the lower the entropy value. On the contrary, if the selection is very diverse and contains features describing many different functionalities, the entropy will be high. In other words, if the entropy is low, the selection will be more *intelligible*, because it will be focused on a specific functionality. Optimal selections in this problem space are of low entropy.

Entropy doesn't capture the difference pointed out in Section III, *i.e.*, if the selected counters are focused on a functionality which is rare in the original set, it will have the same score as if it was focused on a more frequent functionality.

The score for this feature selection method should be greater if the selection focuses on the rarest features in the dataset.

In that respect, cross-entropy, *i.e.*, the relative entropy of a distribution compared to a reference, quantifies how focused/specific a selection is, along with how much it differs from the reference dataset. Cross-entropy captures how specific the information is in the selection, with the original distribution as reference distribution. For a given token type $0 < k \leq K$, cross-entropy is expressed as follows:

$$H\left(p_k, q_k\right) = -\sum_{i \in P} p_{k,i} \log q_{k,i} \tag{1}$$

If $p_k = q_k$, the cross-entropy value will simply be the entropy of the original distribution q_k. This means that random selections will have a cross-entropy value of $H(q_k)$, while the scores of selections which focus on a specific functionality will be $H(p_k, q_k) > H(q_k)$. Additionally, the cross-entropy value will be greater if the focus is on a rare functionality, since the metric captures the difference in entropy between the two distributions.

Cross-entropy is very close to a divergence metric between two distributions. Compared to *e.g.*, the Kullback-Leibler (KL) divergence [14], cross-entropy also indicates the specificity of the selection. When the difference between cross-entropy and the entropy of the original distribution is computed (to remove the constant component $H(q_k)$), it can be expressed as the sum of the KL-divergence and the difference in entropy between the two distributions, *i.e.*, the information gain $IG(q_k|\mathcal{S})$.

$$\begin{aligned} H\left(p_k, q_k\right) - H\left(q_k\right) &= D_{KL}\left(p_k || q_k\right) + H\left(p_k\right) - H\left(q_k\right) \\ &= D_{KL}\left(p_k || q_k\right) - IG(q_k|\mathcal{S}) \end{aligned} \tag{2}$$

Not only does this score describe the distance from the original distribution (divergence), it also provides an indication on how concentrated the information is in the selection, compared to the original dataset (specificity).

B. Optimization problem definition

Considering a multivariate time-series of dimension $N > 0$, the frequencies describing the rareness of features $\{q_k\}_{0<k\leq K}$ and $\{p_k\}_{0<k\leq K}$ are defined as in Section III. At a given time, each feature i, *i.e.*, each uni-variate time-series, has an associated score σ_i that quantifies the amount of change. A simple example of this score is the normalized difference in mean value, on a small window before and after the time of the event (measuring the amplitude of baseline changes at the time of the event). This paper presents a method to find the subset of features $\mathcal{S}$ which describes best what is changing at a given time.

This can be expressed as an optimization problem, which aims at selecting features with a high change score, forming a subset of high cross-entropy, by maximizing the product of the two metrics (change score σ_i and cross-entropy $H(p_k, q_k)$). The idea behind this optimization process is that optimal selections will both picture the change around a given time, with the change score, and diverge from the original feature set with high specificity, with the cross-entropy.

The optimization objective to maximize is the simple product between the cross-entropy of the selection with regards to its original dataset, and the average change score (averaged in order to be independent of the size of the dataset). The simple product is taken as optimization score, in order to maximize the two metrics jointly. For $0 < k \leq K$ the optimization score $\mathcal{L}'_k$ is defined as follows:

$$\mathcal{L}'_k(\mathcal{S}, p_k, q_k) = H\left(p_k, q_k\right) \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \sigma_i \tag{3}$$

where $\mathcal{S}$ is the selected subset in the original dataset.

This process results in a fixed size selection. In order to relax the constraint and have the method offer configurable amounts of features which describe the event, a regularization term is added to the score to penalize very small selections and arrive at the final definition of $\mathcal{L}_k$:

$$\mathcal{L}_k(\mathcal{S}, p_k, q_k) = \left(1 - e^{-\frac{|\mathcal{S}|}{\alpha}}\right) H\left(p_k, q_k\right) \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \sigma_i \tag{4}$$

where α defines how much smaller subsets are penalized. A higher α leads to selections with higher cardinality.

The resulting scores can be aggregated by summing the values of cross-entropy, in order to take all token types into account, giving the final optimization goal $\mathcal{L}$, with $p = \{p_k\}_{0<k\leq K}$, and $q = \{q_k\}_{0<k\leq K}$,

$$\mathcal{L}(\mathcal{S}, p, q) = \left(1 - e^{-\frac{|\mathcal{S}|}{\alpha}}\right) \sum_{k=1}^{K} H\left(p_k, q_k\right) \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \sigma_i \tag{5}$$

C. Optimization process

The following algorithm is applied for the optimization process. The process is initialized by selecting a random set of features. At every iteration of the optimization process, the impact of the addition/removal of each feature in the selected set is computed. Only those additions/removals that improve the optimisation score ($\mathcal{L}$) are maintained/removed from the set. This process is repeated until no further additions/removals improve the score.

At the end of the optimization process, the selected features are ordered by their contribution to the score (computed by leave-one-out), in order to have the most important features first.

V. Experiments

The method proposed in this paper helps the explanation of events in a multivariate time-series and, explicitly, does not detect events. The times at which noticeable events happen are supposedly known at this point, and can result from any multivariate time-series change point detection algorithm [15].

A. Setup

The experiment was run on MDT datasets where the feature names follow the nomenclature of an associated YANG model [1]. The data was extracted from a router through MDT collections in a lab environment, where typical network events were inserted, such as interface failures, routing loops, traffic black holes, etc.

The results showcased below result from running the proposed algorithm on the dataset from [2], created by retrieving several MDT collections with a cadence of 10s over a period of 5 hours and 28 minutes in a lab environment. The resulting set consists of 6622 features describing the router's state, with triggered enables/disables on interface 10, and enables/disables of a bidirectional forwarding detection (BFD) session. The timestamps at which the events occur are known, and the change score (σ_i) for each uni-variate time series was computed as the normalized difference in a baseline on windows of 10 points before and after the time of the event.

B. Results

From among the 6622 counters, the selection process returns the following subsets for $\alpha = 1$. The sensor paths in the rest of the paper were abbreviated for the sake of readability, but the detailed features names can be found in the publicly available dataset [2].

Admin Interface 10 Shutdown: (4007 counters changing)

- `bfd_summary::session-state_down-count`
- `bfd_summary::session-state_up-count`
- `interface-summary::admin-down-interface-count`
- `interface-summary::up-interface-count`
- `bfd_counters:HundredGigE0/0/0/10:hello-transmit-count`
- `bfd_session:HundredGigE0/0/0/10:negotiated-local-transmit-interval`
- `bfd_session:HundredGigE0/0/0/10:negotiated-remote-transmit-interval`

Breaking BFD Session: (3819 counters changing)

- `bfd_summary::session-state_up-count`
- `bfd_summary::session-state_down-count`
- `bfd_counters:HundredGigE0/0/0/16:hello-receive-count`
- `bfd_counters:HundredGigE0/0/0/16:hello-transmit-count`
- `bfd_session:HundredGigE0/0/0/16:negotiated-remote-transmit-interval`
- `bfd_session:HundredGigE0/0/0/16:negotiated-local-transmit-interval`

C. Discussion

The resulting selections for $\alpha = 1$ are both intelligible and descriptive of the corresponding events. In the case of the interface 10 shutdown, 4007 features changed in value, and 3819 did in the case of the broken BFD session. In both cases, the selection process has allowed the selection of less than 10 of these features, which can be categorized as meaningful to an operator (BFD sessions counts and interface counts). With the removal of the many counters that can obscure a user's analysis, it will be easier for an operator with this condensed view to infer an explanation of the event.

Figure 1 shows the size of the selections compared to the original number of features for the different types of events contained in the dataset. Figure 1 also displays the number of changed features around the time of the event, and the number of features whose values have changed by more than 95% in absolute value.

As a comparison, the simple method of extracting the top 10 features with highest value of σ_i (normalized difference in mean value on windows of 10 points before and after the timestamp) for the interface 10 shutdown would return:

Interface 10 Shutdown (Basic method for comparison)

- `fib-statistics:0/0/CPU0:incomplete-adjacency-packets`
- `bgp:ipv4:performance-statistics_vrf_update-generation-prefixes-count`
- `bgp_default-vrf_afs-process-info:ipv6:global_label-version`
- `bgp_default-vrf_afs-process-info:ipv6:global_last-rib-version`
- `bgp_default-vrf_afs-process-info:ipv6:local-paths-freed-num`
- `bgp_default-vrf_afs-process-info:ipv6:local-paths-malloced-num`
- `bgp_default-vrf_afs-process-info:ipv6:paths-freed-num`
- `bgp_default-vrf_afs-process-info:ipv6:paths-malloced-num`
- `bgp_default-vrf_afs-process-info:ipv6:rib-acked-table-version`
- `bgp_default-vrf_afs-process-info:ipv6:rib-bgp-version`

Although these features are actually caused by the event (re-routing caused by the interface shutting down), they are unrelated to the nature of the event itself. This specific dataset contains 514 (8%) VRF related sensor paths. As discussed in Section III, because of their high occurrence in the dataset, they are estimated to carry less meaning to the network engineers. As an indication, in the selection made by this paper's method with $\alpha = 1$, the change scores (σ_i) of the selected features ranged from 51th to 109th highest change score. This implies that the contribution of cross-entropy to the optimization score resulted in the selection of the features which carry most information (*e.g.*, interface counts) and the removal of those that did not, among the features changing most in value. This observation confirms the contribution of both components of the optimization score ($\mathcal{L}$): while the change score (σ_i) allows the method to select the counters that see a change in value around the event, the cross-entropy selects those which are most meaningful to a network engineer (*i.e.*, rare in the dataset).

The α parameter further allows the tuning of the verbosity of the selection. For example, with lower α parameters, the selection for the interface 10 shutdown becomes:

Admin Interface 10 Shutdown: ($\alpha = 0.47$)

- `bfd_summary::session-state_down-count`
- `bfd_summary::session-state_up-count`
- `interface-summary::admin-down-interface-count`
- `interface-summary::up-interface-count`

Admin Interface 10 Shutdown: ($\alpha = 0.1$)

- `bfd_summary::session-state_down-count`
- `bfd_summary::session-state_up-count`

A higher value of α can allow for a more detailed view (which can hint at the details of the events, *e.g.*, locality) whereas lower values can result in scarce selections.

VI. Conclusion and Future Work

This paper proposes a general method for event description in multivariate time-series using both data behavior, through the change score, and contextual information, through information retrieval on the feature names only. The proposed estimation of the intelligibility of a selection, through the rareness of its components, allows the identification of semantically important counters and the definition of an optimization goal. The method produces intelligible selections, which can ease the interpretation of events by a network operator, while allowing the exploitation of all the available counters in the dataset. This prevents operators from having to tediously hand pick which features to monitor, in cases where the data is of high dimension. The proposed method is equipped with a single value parameter, allowing the adjustment of the desired output verbosity. This allows to find the selection size which best fits a given use case, by distilling the right amount of information. This method has shown to generate selections of less than ten highly meaningful features out of more than six thousand in the dataset in [2], for events with different root causes.

Although a mean difference for the change score, and probabilities defined as above, may produce intelligible results for network telemetry datasets, they need to be further evaluated, in cases where (i) the changes in the data are more complex, or (ii) the relative rareness of a feature isn't the best indicator of its relevance. These assumptions were taken based on the general observation that it is usually the case in network telemetry datasets. These two metrics are nevertheless defined independently from the general method to compute optimal selections, and they can be adapted to any other problem space.

References

[1] E. M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," Internet Requests for Comments, RFC Editor, RFC 6020, October 2010. [Online]. Available: https://tools.ietf.org/html/rfc6020

[2] Cisco Innovation Edge, "Network anomaly telemetry datasets," https://github.com/cisco-ie/telemetry/tree/master/11, 2019.

[3] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artificial Intelligence Review*, 2019. [Online]. Available: https://doi.org/10.1007/s10462-019-09682-y

[4] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 647–665, 2014.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," *CoRR*, vol. abs/1602.04938, 2016. [Online]. Available: http://arxiv.org/abs/1602.04938

[6] S. Alelyani, H. Liu, and L. Wang, "The effect of the characteristics of the dataset on the selection stability," in *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, Nov 2011, pp. 970–977.

[7] J. Yao, Q. Mao, S. Goodison, V. Mai, and Y. Sun, "Feature selection for unsupervised learning through local learning," *Pattern Recognition Letters*, vol. 53, pp. 100 – 107, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865514003559

[8] M. Dash, H. Liu, and J. Yao, "Dimensionality reduction of unsupervised data," in *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*, Nov 1997, pp. 532–539.

[9] R. Varshavsky, A. Gottlieb, M. Linial, and D. Horn, "Novel Unsupervised Feature Filtering of Biological Data," *Bioinformatics*, vol. 22, no. 14, pp. e507–e513, 07 2006. [Online]. Available: https://doi.org/10.1093/bioinformatics/btl214

[10] V. M. Rao and V. N. Sastry, "Unsupervised feature ranking based on representation entropy," in *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, March 2012, pp. 421–425.

[11] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, March 2002.

[12] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Information Processing and Management*, vol. 39, no. 1, pp. 45–65, 2003.

[13] "Telemetry Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.1.x," 2019. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/telemetry/b-telemetry-cg-ncs5500-61x/b-telemetry-cg-ncs5500-61x_chapter_010.html

[14] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. [Online]. Available: http://www.jstor.org/stable/2236703

[15] "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 339–367, 2017. [Online]. Available: https://doi.org/10.1007/s10115-016-0987-z