Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Telecommunication Engineering

Hierarchical density-based clustering and interpretation for network measurements

Doctoral Thesis

Ing. Pavol Mulinka

Ph.D. programme: P2612 Electrical Engineering and Information Technology
Branch of study: 2601V013 Telecommunication Engineering
Supervisor: Dr. Lukáš Kencl

Prague, October 2020

I hereby declare that I elaborated this doctoral Thesis independently and used only sources in the bibliography.

Abstract

Automatic detection and interpretation of network traffic anomalies through machine learning is a well-known problem, for which no general solution is available. Supervised learning solutions are often employed when there is a clear idea of the traffic patterns to be detected, whereas anomaly detection (i.e., detection of outliers) is the preferred solution when only the normal behaviour of the monitored system is known. Both approaches require prior knowledge about the monitored system, either the normal operation profiles or the specific anomalies patterns. As a consequence, both approaches have clear limitations when it comes to detecting, and interpreting unknown events.

In this work we present Hi-Clust, an universally applicable hierarchical density-based clustering approach for unsupervised network traffic analysis, which can both detect and characterize anomalous behaviours in a completely black-box manner, without relying on any ground-truth. Hi-Clust tackles the combined detection and interpretation of anomalies in multi-dimensional network data as an unsupervised machine learning task, relying on hierarchical clustering techniques for pattern discovery and interpretation. Hi-Clust can be applied to the unsupervised analysis of any kind of nested or hierarchically structured multi-dimensional data. We apply this approach to two distinct classification scenarios: (i) transit Internet traffic and (ii) active Cloud latency measurements. We describe the application procedure in both scenarios, benchmarking different methods for automatic identification of relevant features describing the detected events and propose methods of feature extraction and their suitable sets in both scenarios. The main contribution of Hi-Clust is its ability to discover novel data patterns. Consequently, we demonstrate Hi-Clust's capability to carry out network traffic classification by interpretation of patterns with a structural approach. We also indirectly show how Hi-Clust discovers additional anomalies in comparison to traditional methods of labeling datasets.

Index terms: Networks, Cloud, Latency, Classification, Interpretation, Unsupervised machine learning

Abstrakt

Automatická detekce a interpretace anomálií síťového provozu s pomocí strojového učení je známý problém, pro který však není známé žádné všeobecné řešení. Řešení využívající strojové učení s učitelem jsou často používána, když existuje jasná představa o provozních vzorech, které mají být detekovány, zatímco detekce anomálií je preferovaným řešením, známe-li pouze normální chování monitorovaného systému. Oba přístupy vyžadují předchozí znalosti o sledovaném systému, buď o jeho normálních provozních profilech nebo o specifických vzorech anomálií, což efektivně znemožňuje jejich využití pro detekci a interpretaci neznámých provozních vzorů. V této práci představujeme Hi-Clust, automatický a univerzálně použitelný způsobanalýzy síťového provozu založený na hierarchickém klastrování, který dokáže detekovat a charakterizovat anomálie pomocí tzv. "black-box" přístupu, tedy bez nutnosti spoléhat se na jakékoli pravdivostní hodnoty trénovacích vzorků. Hi-Clust řeší kombinovanou detekci a interpretaci anomálií ve vícerozměrných síťových datech pomocí strojového učení bez učitele a při detekci a interpretaci vzorů se spoléhá na metody hierarchického klastrování. Hi-Clust lze použít na analýzu jakéhokoliv druhu vnořených nebo hierarchicky strukturovaných vícerozměrných dat. Aplikaci navrženého přístupu demonstrujeme na dvou odlišných klasifikačních scénářích: (i) tranzitní Internetový provozu a (ii) aktivní měřeních latence Cloudu. Pro oba případy popisujeme postup aplikace, porovnáváme různé metody automatické identifikace měření popisujících zjištěné události a navrhujeme metody extrakce prvků a jejich vhodných sad.. Hlavní výhodou Hi-Clustu je jeho schopnost objevit zcela nové vzory v datech. Tu prakticky demonstrujeme na využití Hi-Clust přístupu, založeného na strukturované interpretaci vzorů, pro klasifikaci síťového provozu. Také, i když nepřímo, ukazujeme jak Hi-Clust nalézá zcela nové vzory v porovnání se standartními metodami založenými na anotaci datasetů.

Klíčová slova: Počítačové Sítě, Cloud, Latence, Klasifikace, Interpretace, Strojové učení bez učitele

Always pass on what you have learned.

–YODA, REVENGE OF THE SITH

Acknowledgments

I would like thank to Dr. Lukáš Kencl for his guidance and advice during my doctoral study, for his support in research work and most of all, for his patience with me and my transition from Network Engineer to Data Scientist. I still find it hard to believe he did not give up on me. I would like to thank him for introducing me to Alessandro D'Alconzo, Ph.D. and Dr. Pedro Casas from AIT Vienna.

My special thanks go to Dr. Pedro Casas for being a great supervisor and a friend during and after my internship at AIT Vienna and also for introducing me to Souneil Park, Ph.D. and Diego Perino, Ph.D. from Telefonica I+D in Barcelona and to Dr. Kensuke Fukuda from NII Tokyo.

I am thankful to Souneil Park, Ph.D and Diego Perino, Ph.D. for the opportunity to gain experience in different research field under their supervision in Barcelona.

I also thank Dr. Kensuke Fukuda for guidance during my stay at NII Tokyo and prof. Václav Hlaváč for giving me advice and an opportunity to apply for the internship.

Last, but not least I thank prof. Boris Šimák, doc. Ing. Jiří Vodrážka, Ph.D. and the whole department of Telecommunications at CTU, specifically my university colleagues Ondřej Tománek, Ph.D. and Jan Staněk, Ph.D. for support when I needed it and to all people working for institutions that provided me research grants and opportunity to follow my dream of becoming a Data Scientist. This Thesis would never have been possible without all the support from following research grants and institutions : Czech Technical University SGS and Mobilita-Akce 200, Cisco Systems Collaborative Research Program, AKTION Czech Republic-Austria, AIT Vienna, Fundación Universidad-Empresa, Telefonica I+D Barcelona and NII International Internship Program.

A Jedi uses the Force for knowledge and defense, never for attack.

–YODA, THE EMPIRE STRIKES BACK

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ARI	Adjusted Rand Index
BGP	Border Gateway Protocol
Big-DAMA	Big Data Analytics for network traffic Monitoring and Analysis
CLAudit	Cloud Latency Auditing platform
CDbw	Compose Density between and within clusters
DARPA	Defense Advanced Research Projects Agency
DBCV	Density-Based Clustering Validation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial Of Service
FTP	File Transfer Protocol
FTPS	FTP over SSL
GML	Generic Machine Learning
HDBSCAN*	Hierarchical DBSCAN
HTTP	HyperText Transfer Protocol
HTTPS	HTTP over TLS/SSL
ICMP	Internet Control Message Protocol
IMAP	Internet Message Access Protocol
IMAPS	IMAP over SSL
IQR	Inter Quartile Range
ISCX	Information Security Center of Excellence
KDD	Knowledge Discovery and Data mining

KDE	Gaussian Kernel Density Estimation
KPI	Key Performance Indicator
LAP	Laplacian Score
LAR	Least Angle Regression
MAWI	Measurement and Analysis on the WIDE Internet
MCFS	Multi-Cluster Feature Selection
NDFS	Nonnegative Discriminative Feature Selection
NETBIOS	Network Basic Input Output System
NETBIOS-NS	NETBIOS Name Service
NETBIOS-DGM	NETBIOS Datagram Service
NETBIOS-SSN	NETBIOS Session Service
NDFS	Nonnegative Discriminative Feature Selection
NSL-KDD	Network Security Lab - KDD
OPTICS	Ordering Points To Identify the Clustering Structure
PCA	Principal Component Analysis
PDU	Protocol Data Unit
POP3	Post Office Protocol version 3
RBF	Radial Basis Function
RLOGIN	Remote Login
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SVD	Singular Value Decomposition
SSH	Secure SHell
SPEC	Spectral Feature Selection
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLP	Transport Layer Protocol
UDFS	Unsupervised Discriminative Feature Selection
UDP	User Datagram Protocol

UNIDS	Unsupervised Network Intrusion Detection System
UUCP	Unix to Unix Copy
WIDE	Widely Integrated Distributed Environment

Chapter 1

Introduction

The problem we address in this thesis is detection and interpretation of unknown network traffic patterns. A spectrum of methods exists for monitoring of network status, ranging from passive collection of network traces and device logs to active probing of the connections. Usually, domain experts define dictionaries of network patterns and design tailored machine learning models for their detection. Everything else is considered anomalous or suspicious behavior. Our concern is the detection and interpretation of the unknown network patterns. We solve this problem by hierarchical unsupervised density-based clustering analysis and structured interpretation of results. We validate our results by clustering performance evaluation metrics and compare the results to previously identified patterns in the case of transit Internet traffic classification. We distinguish between general, suspicious network behaviour and outliers, each of which is defined by their ability to form a cluster and the size of the cluster, i.e. the largest cluster defines general behaviour and outliers are not forming clusters.

In the following, we introduce our motivation and solution and we summarize our contributions at the end of the section.

1.1 Motivation and problem statement

Network traffic classification, and anomalous behaviour detection and interpretation are often considered as two separate problems in monitoring and threat detection. The former is usually addressed by supervised, and the latter by unsupervised, machine learning methods. In general, no established guidelines exist for detection and interpretation of unknown network traffic patterns. Each pattern definition is dependent on its

monitoring solution and gathered dataset. This is a time consuming task which is often simplified by definition and detection of well known traffic patterns, e.g. DoS (Denial of Service), port scan, network scan, SYN flood, etc. Clustering is a well-known and highly useful technique when no ground-truth data is available, which is the usual case when dealing with real, "in the wild" network measurements. Based on the observation that anomalies occur, by definition, less frequently than normal operation, their unsupervised detection and analysis consists of identifying major behaviour clusters and less frequent suspicious behaviour clusters and outliers, i.e. instances that are remarkably different from the majority. Clustering is per se a very challenging task. Various clustering algorithms have different underlying assumptions and are capable of identifying clusters with very diverse structures; in addition, they usually have multiple configuration parameters which are difficult to set and might highly impact the results.

1.2 Solution

We present an approach overcoming this problem by uniting both (i) network traffic classification and (ii) anomalous behaviour detection into hierarchical density-based clustering analysis and interpretation. In a nutshell, our hierarchical model Hi-Clust uses clustering to firstly detect normal and abnormal behaviour in the multi-dimensional collected (and pre-filtered) dataset, and to then correlate and jointly analyse multiple dimensions of the communications stack to interpret the potential causes behind the detected behaviours.

We demonstrate benefits of the approach in two distinct scenarios, by solving (i) the transit Internet network traffic classification problem from the *macroscopic* view of overall network traffic analysis of the MAWI (Measurement and Analysis on the WIDE Internet) [1] traces and by (ii) the active Cloud latency measurements classification from the geographically distributed vantage points of CLAudit (Cloud Latency Auditing Platform) [2].

1.3 Summary of contributions

- We introduce a set of features suitable for transit Internet traffic classification;
- We present a novel approach to unknown pattern detection and interpretation in network traffic;
- We show a new approach for interpretation of clusters in active Cloud latency measurements;
- We present novel methodology and guidelines for selection and evaluation of density-based clustering algorithms in absence of ground truth, based on the state-of-the-art performance evaluation metrics;
- We present a new structured approach for unsupervised detection and interpretation of hierarchical clusters;
- We analyse and evaluate multiple clustering algorithms on the two distinct real world network traffic datasets;
- We introduce a new version of Hi-Clust [3].

1.4 Outline

- In Chapter 2, we present an essential overview of state-of-the-art network/Cloud traffic monitoring and classification solutions, types of machine learning and more specifically unsupervised machine learning methods and performance evaluation metrics we will use;
- In Chapter 3, we dive into the methodology of our proposed solution to unknown network traffic pattern detection and interpretation, and explain the procedure in detail in four successive steps;
- In Chapter 4, we apply the procedure to two distinct classification scenarios, show the essentiality of each step on real world datasets and provide the interpretation of the detected clusters;
- In Chapter 5 we summarize the results and speculate about the questions opened by our solution to unsupervised detection and interpretation of unknown network traffic patterns;
- In Appendix A we include a list of Ph.D. candidates publications related to this

work;

- In Appendix B we include a list of Ph.D. candidates publications related to other projects;
- In Appendix C we include a list of Ph.D. candidates projects related to this work;
- In Appendix D we include other Ph.D. candidates results;
- In Appendix E we include all results we used for cluster interpretation.

Chapter 2

State of the Art

In this section, we provide an explanation of essential terms in the context of machine learning followed by an overview of the state-of-the-art research performed in three major domains covered by our study:

1. Internet traffic classification;
2. Cloud monitoring;
3. Unsupervised machine learning.

Under machine learning we understand a research field that intersects computer science, AI (Artificial Intelligence) and statistics. It extracts knowledge from the data for predictive analytics (supervised machine learning, i.e. learning with labeled data) or intrinsic properties (unsupervised machine learning, i.e. learning with no labeled data) [4]. In between, stands semi-supervised machine learning that typically uses a small amount of labeled data to label the large amount of grouped unlabeled data. Regardless of machine learning type, each dataset consists of data points, i.e. samples, and their features. In the context of machine learning we denote a feature as an individual measurable property of the observed entity. Examples for each type of machine learning can be found in Fig. 2.1, an adjusted machine learning categorization by authors of [5] .

We focus on three subcategories of machine learning:

- **Classification** - prediction of data point labels from a predefined list of possibilities;
- **Clustering** - grouping of data points based on their similarity;
- **Outlier detection** - detection of observations which deviate from the rest of the dataset as if they were generated by different mechanism [6], sometimes called anomaly detection.

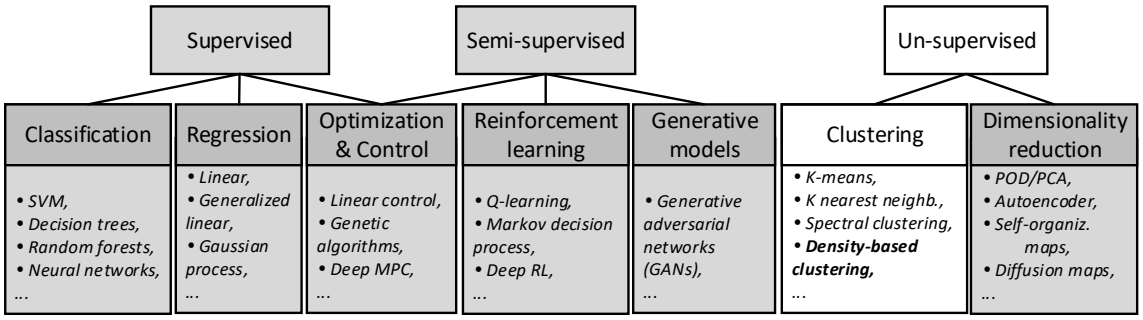


Figure 2.1: *Machine learning algorithms categorization.*

2.1 Internet traffic classification

There are a couple of extensive surveys on general domain outlier detection techniques [7] as well as network outlier detection [8, 9], including machine-learning-based approaches. The literature is a particularly extensive in the application of learning-based approaches for automatic traffic analysis and classification. We refer the interested reader to [10] for a detailed survey on the different machine learning techniques commonly applied to network traffic analysis.

Machine learning methods are most often evaluated on well-known labeled network traffic datasets, e.g. DARPA 1998/1999/2000 (Defense Advanced Research Projects Agency) [11, 12, 13], ISCX 2012 (Information Security Center of Excellence) [14], generated/artificial, etc.. KDD99 (Knowledge Discovery and Data mining cup in 1999) [15] is the most used dataset in machine learning for Network Intrusion Detection Systems in the last decade [16]. Recent studies suggest that KDD99 does not reflect the modern attack footprint or normal traffic scenarios [17], but researchers still tend to use the outdated KDD99 [18, 19, 20] or its cleaned up version NSL-KDD (Network Security Lab - Knowledge Discovery and Data Mining) [21]. We focus on applying machine learning methods to recently collected real world data and use MAWI transit Internet traffic traces.

We recently used the MAWI dataset for the application of supervised machine learning models to network security and anomaly detection problems [22, 23, 24, 25] and in [23] we introduced Big-DAMA, a big data analytics framework for large scale network traffic monitoring and analysis. In [22] we compared the performance of standard, off-line machine learning models for network security in fixed-line networks, further studying more complex and robust models based on ensemble machine learning techniques. Wireless network monitoring using similar techniques was studied in [25]. The knowledge and conclusions extracted from these papers motivated the introduction of the GML (Generic

Machine Learning) model for network measurements analysis [24], which achieved high accuracy for many different network analysis problems.

However, all these approaches are considered supervised analysis of network measurements, i.e. dataset was labeled.

2.2 Cloud monitoring

Many general purpose [26, 27, 28, 29] monitoring systems exist for local and wide area network performance analysis. In general, traffic monitoring systems use thresholds for suspicious behaviour detection and do not provide interpretation based on triggering events. Cloud monitoring is a specially tailored network and systems monitoring solution for a distributed computing environment with shared resources.

Cloud monitoring today is mainly about in-Cloud measurement, i.e., monitoring Cloud resources from within the Cloud infrastructure, without considering the end-to-end service perspective. Major Cloud service providers such as Amazon, Azure and others provide their own in-Cloud tools for the monitoring of Cloud resources, including tools such as Amazon CloudWatch [30], Azure Monitor [31], etc.. Cloud-specific [32, 33] third-party monitoring systems generally monitor Cloud resources assuming an internal knowledge of either the Cloud instances, Cloud traffic or the running Cloud applications. In addition, these systems do not take into account end-user experience related metrics such as end-to-end latency, and focus on lower level metrics such as service availability and uptimes. In [34], the authors present a Cloud monitoring survey providing good insights into approaches and metrics used for Cloud monitoring, which indicates that latency measurements are rarely used as the main Key Performance Indicator (KPI) for Cloud monitoring. As pointed out by a survey [35] on Cloud monitoring, while there is quite a vast variety of tools applied in practice for Cloud monitoring and benchmarking, there is a general lack of research approaches exploiting cross-layer monitoring and analysis of Cloud services.

We build on top of our previous work with cross-layer Cloud latency measurements from CLAudit [36, 37] and apply unsupervised machine learning methods on CLAudit measurements for clusters and outlier detection.

2.3 Unsupervised machine learning

There is quite a vast literature on clustering-based approaches for unsupervised traffic classification and anomaly detection, but most of them target the network security domain [38, 39, 40]. In [38] we introduced UNIDS, an Unsupervised Network Intrusion Detection System capable of detecting unknown network attacks through density-based, sub-space clustering techniques. We take the learnings from previous experiences working with clustering for anomaly detection to conceive Hi-Clust.

2.3.1 Clustering methods

Cluster analysis is a primary approach for data-mining which can be used for unsupervised data analysis and as a preprocessing step for other methods and applications. Authors of clustering algorithm surveys [41, 42, 43] classify algorithms based on their underlying approach: *Partition-Based, Density-Based, Grid-Based, Hierarchical.*

Hi-Clust uses density-based hierarchical clustering as the underlying unsupervised analysis technique, as the main target is to find dependencies and correlations among different nested dimensions in the analyzed data. As clustering a notion, it employs density-based analysis techniques, which identify clusters as highly dense regions separated by lower density ones.

2.3.2 Density-based hierarchical clustering

Density-based hierarchical clustering represents a subset of clustering approaches detecting correlations within nested data dimensions by decomposition of the analyzed dataset into a hierarchical structure. Well-known *Density-Based* algorithms include DBSCAN [44], as well as hierarchical-based extensions or improvements such as HDBSCAN* [45] and OPTICS [46]. Recently, authors of [47] revisited density-based clustering algorithms and summarized the process of data analysis. However, it lacks recommendations or methods for interpretation of results, just as performance evaluation by recently proposed clustering performance evaluation metrics, which we discuss next.

2.3.3 Clustering performance evaluation metrics

When talking about clustering, we need to consider the multiple techniques so far proposed for cluster evaluation and validity purposes. Cluster evaluation and validity

measures can be divided into internal - defined on internal properties of the clusters, and external - which assume the existence of ground truth. As explained earlier, Hi-Clust is a purely unsupervised approach, which considers unlabeled datasets for analysis. We therefore focus on internal cluster properties. Recently, authors in [48] introduced *DBCV (Density-Based Clustering Validation)*, a cluster-validity measure which encompasses density, shape and noise objects, which are properties of the density-based clustering algorithm results. *DBCV* improves over traditional measures such as Calinski-Harabasz [49], Dunn [50], Davies-Bouldin [51], Silhouette [52] score, SD and S_Dbw [53, 54] and the like, using more robust validity definitions. It builds upon density-based validation ideas presented by authors of CDbw (Compose Density between and within clusters) [55]. Based on previous studies on multiple cluster-validity metrics [54, 56, 57], we take *DBCV* as a main evaluation metric for the underlying algorithms benchmarked for Hi-Clust, while briefly exploring CDbw values.

We show that the Hi-Clust approach provides new insights compared to pattern recognition systems by comparing clustering results to ground truth extracted from MAWILab database [58]. This database contains anomaly traffic labels for the MAWI samplepoints B and F.

We followed the research community recommendations [59] to measure the similarity of the two independent label assignments and used a V-Measure [60] score to compare our clustering results to labels extracted from MAWILab database. Other popular scores include ARI (Adjusted Rand Index) [61], Fowlkes-Mallows index [62] or pairwise F-measure [63]. Recommendations suggest that V-measure is preferred when the dataset consists of more than one thousand samples and the expected number of clusters is ~ 10. We will confirm this claim in Chapter 4 when we describe the datasets in detail and apply the Hi-Clust approach.

2.4 Conclusion

In this section, we summarized state-of-the-art research in the of field supervised and unsupervised machine learning related to network traffic classification and explored Cloud monitoring alternatives to CLAudit. In the next section, we provide a detailed explanation of Hi-Clust and its methodology for clustering performance evaluation and interpretation of the results.

Chapter 3

Solution

In this section, we present building blocks of Hi-Clust. We define and explain each step of the hierarchical density-based approach to network traffic analysis described in Fig. 3.1. The approach is separated into four successive steps :

1. Dataset gathering and extraction ;
2. Features transformation for unsupervised machine learning analysis;
3. Algorithm selection and calibration;
4. Clustering results interpretation.

Tasks that are scenario-specific are marked by a hatched pattern, e.g. we do not perform feature selection and scaling on CLAudit latency measurements as it is important to keep all measurements unaltered for the interpretation.

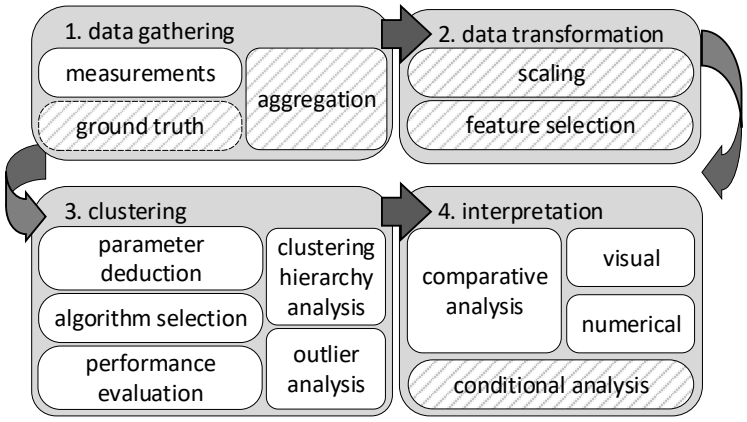


Figure 3.1: *Hi-Clust overview.*
An overview of the procedure for detection and interpretation of network patterns and suspicious events.

3.1 Data gathering

Data gathering is specific to each scenario and will be explained separately.

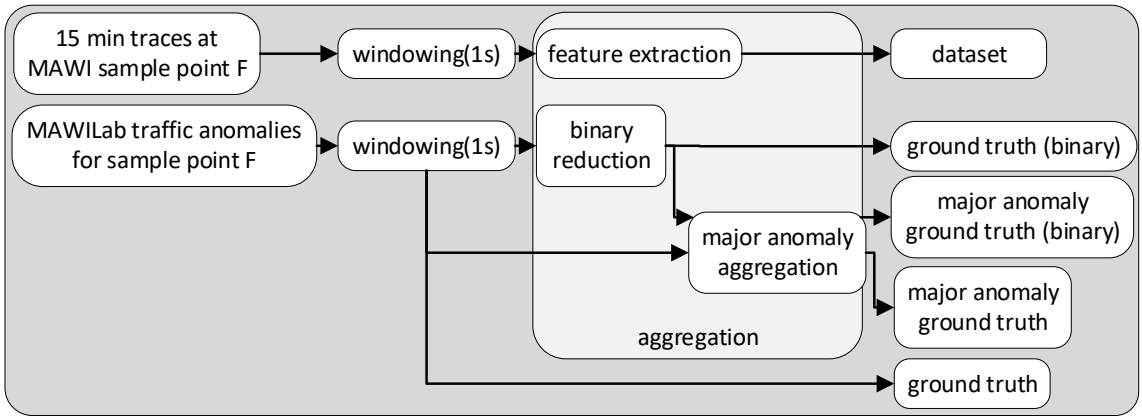
3.1.1 MAWI (Measurement and Analysis on the WIDE Internet)

MAWI daily traces are collected every day from 14:00 to 14:15 at samplepoint-F at the transit link of WIDE (Widely Integrated Distributed Environment), AS2500. Traces are inspected for anomaly patterns and stored in MAWILab database. We use MAWILab for a complementary traffic interpretation. Fig. 3.2 shows a MAWI specific process of dataset gathering along with an overview of WIDE environment where MAWI traces are collected. MAWILab detects predefined anomalies in MAWI traces. Features are extracted from all packets within the time window - i.e. we extract features from packets not flows. Tab. 3.1 summarizes extracted features. For TCP/UDP port features we used the subset of application protocol categories defined in *L7-filter* [64], see Tab. 3.2. MAWILab identifies anomalies with a precision of 1 second. We apply windowing with the same 1 second interval on daily traces and extract features. Detected anomalies are available in time range format - the time interval for each detected anomaly in each 15 minute trace. We transformed the 15 minute time range of each anomaly to binary vectors of 900 values, binary 1 representing an anomaly. As a result, each 1 second time interval was assigned a binary vector combining all anomaly types.

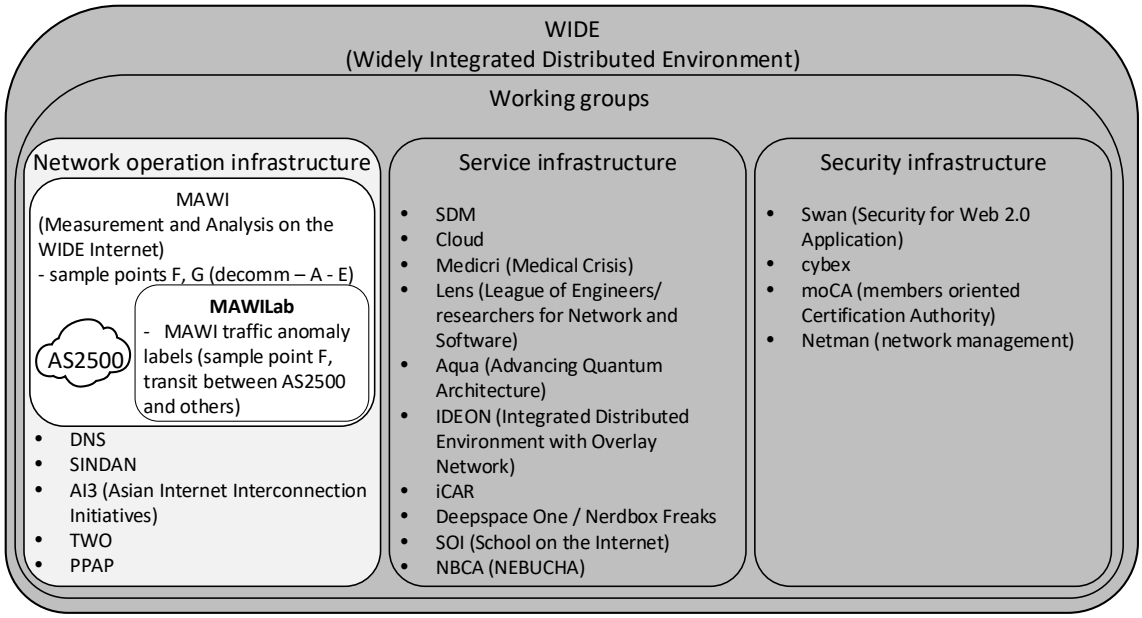
MAWILab anomaly taxonomy is hierarchical, built through an iterative process of network anomaly analysis by expert knowledge [65]. For major anomaly categories and their subcategories see Fig. 3.2 b). Some anomalies are concurrently in two major categories, eg. HTTP anomalies are included both in HTTP and Multipoint/Alpha Flow. We extracted four anomaly label datasets:

- Ground truth;
- Ground truth (binary);
- Major anomaly ground truth;
- Major anomaly ground truth (binary).

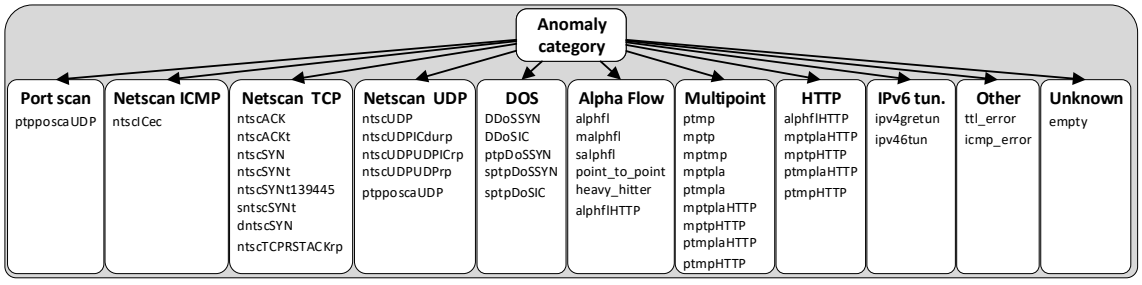
Binary datasets include only *the presence* of each anomaly type in a 1 second time slot. Non-binary datasets include the magnitude of each anomaly in a 1 second time slot, i.e. the anomaly could be detected from multiple IP sources - *Ground truth* and *Major*



(a) *MAWI Anomaly and Feature extraction*



(b) *MAWI WIDE structure*



(c) *MAWILab anomaly categories*

Figure 3.2: *Hi-Clust Step 1 : Data gathering - MAWI.*

anomaly ground truth datasets. *Major anomaly* datasets merge anomaly subcategories to major categories, i.e. *Unknown, Other, HTTP, Multipoint, Alpha flow, IPv6 tunneling, Port scan, Network scan ICMP, Network scan UDP, Network scan TCP, DoS*. We use major anomaly datasets to address the datasets general anomaly characteristics.

In the case of MAWI dataset, we perform clustering performance evaluation using internal and external evaluation metrics. External evaluation metrics require ground truth, i.e. a labeled dataset. Binary ground truth datasets can be seen as a collection of binary vectors. Each 1 second time slot is defined by a binary vector that shows which anomaly type was present. We mark each unique binary combination of anomaly types as a unique cluster and use the resulting dataset for external performance evaluation by V-measure. We calculate and analyse the V-measure for each clustering result with labels obtained from Ground truth (binary) and Major anomaly ground truth (binary) datasets as described above.

In the cluster interpretation step, we focus on average and distribution of cluster feature values compared to *General behaviour*. For complementary interpretation information, we analyse average and distribution of anomalies multitude in each cluster in Ground truth and Major anomaly ground truth datasets.

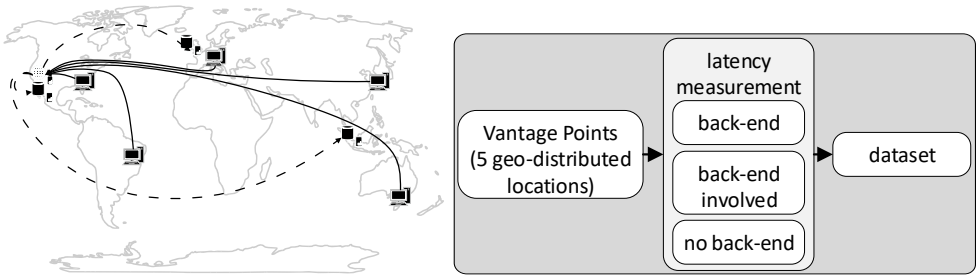
In summary, we use *binary* ground-truth datasets for performance evaluation and *non-binary* ground-truth datasets for additional information in cluster interpretation.

field	feature	description
total volume	# pkts	num. packets
	# bytes	num. bytes
PKT size	pkt_h	H(PKT)
	pkt_min,avg,max,std	min/avg/max/std of pkt. size
	pkt_p1,2,5,...,95,97,99	percentiles
IP PKT size	iplen_h	H(IPlen)
	iplen_min,avg,max,std	min/avg/max/std IPlen
	iplen_p1,2,5,...,95,97,99	percentiles
IP TTL	ttl_h	H(TTL)
	ttl_min,avg,max,std	min/avg/max/std TTL
	ttl_p1,2,5,...,95,97,99	percentiles
IP protocol	#/% icmp/tcp/udp/gre	num./frac. of IP protocols
IPv4/IPv6	% IPv4/IPv6	frac. of IPv4/IPv6 pkts.
	h_IP_src/dst_octet	H(IP_octets)
TCP PKT size	tcp_h	H(TCP)
	tcp_min,avg,max,std	min/avg/max/std TCP
	tcp_p1,2,5,...,95,97,99	percentiles
IP MF flag + IP frag offset	#/% of frag. pkts	num./frac. of fragmented packets
ICMP type	#/% of icmp type 0	num./frac. of echo reply packets
	#/% of icmp type 3	num./frac. of dest. unreachable pkts.
	#/% of icmp type 5	num./frac. of redirect message pkts.
	#/% of icmp type 8	num./frac. of echo request pkts
	icmp_type_h	H(icmp_type)
TCP WIN size	win_h	H(TCPW)
	win_min,avg,max,std	min/avg/max/std TCPW
	win_p1,2,5,...,95,97,99	percentiles
TCP flags (byte)	#/% of ACK/CWR/FIN/ ECN/NS/PSH/SYN/URG	num./frac. of TCP flags
TCP/UDP ports (src/dst)	#/% well-known	num./frac. of well-known
	#/% registered	num./frac. of registered
	#/% dynamic	num./frac. of dynamic
	#/% remote-access	num./frac. of remote-access
	#/% mail	num./frac. of mail
	#/% networking	num./frac. of networking
	#/% document retrieval	num./frac. of doc. retrieval

Table 3.1: *MAWI feature descriptions.*

category	TLP	port#	description
remote access	**tcp**	22	ssh
		23	telnet
		992	telnets
		513	rlogin
mail	**tcp**	25	smtp
		110	pop3
		143	imap
		993	imaps
		995	pop3s
	udp	512	biff
networking	**tcp**	43	whois
		67	dhcp
		68	dhcp
		137	netbios-ns
		138	netbios-dgm
		139	netbios-ssn
		161	snmp
		162	snmp-trap
		179	bgp
	udp	53	dns
		67	dhcp
		68	dhcp
		137	netbios-ns
		138	netbios-dgm
		139	netbios-ssn
		161	snmp
		162	snmp-trap
document retrieval	**tcp**	20	ftp-data
		21	ftp
		70	gopher
		80	http
		443	https
		445	smb
		540	uucp
		989	ftps-data
		990	ftps
	udp	69	tftp
		445	smb

Table 3.2: *MAWI application port features detailed*



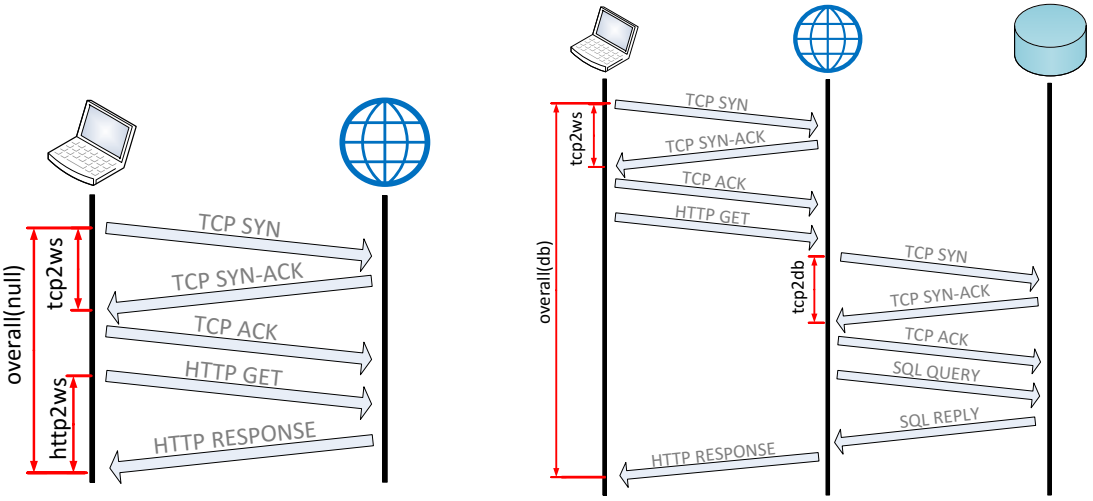
(a) *Worldmap* **(b)** *Vantage Points active measurements*

Figure 3.3: *Hi-Clust Step 1 : Data gathering - CLAudit.*

3.1.2 CLAudit (Cloud Latency Auditing Platform)

Platform Overview

We build upon measurements gathered by CLAudit, the globally-distributed cloud latency measurements platform, which is the *source* of cloud-latency data analyzed in this paper. CLAudit (Cloud Latency Auditing Platform) is a system for collecting and evaluating multidimensional measurements. CLAudit deployment is depicted in Fig. 3.3. By *measurements*, we mean RTTs of individual protocol exchanges, processing times and overall latency, as shown by the webpage retrieval processes in Fig. 3.4. By *multidimensional*, we mean measurements capable of being looked at from the point of view of Vantage Points, Data Centers and/or protocol layers. This section provides a brief overview of CLAudit components and deployment. For a detailed description, see [2].



(a) *Static webpage retrieve* **(b)** *Dynamic webpage retrieve*

Figure 3.4: *Static and dynamic webpage retrieves with time intervals included in discussed datasets denoted.*

measurement type	description	back-end involved
tcp2ws	Time elapsed between VP's TCP SYN sent and TCP SYN ACK from web server received	no
tracert	Time elapsed between VP's ICMP Echo Request sent and ICMP Time Exceeded from farthest reachable network hop received	no
tcp2db	Time elapsed between front-end web server's TCP SYN sent and TCP SYN ACK from back-end database server received	yes
http2ws	Time elapsed between VP's HTTP request sent and static web HTTP response from front-end web server received	no
overall(db)	Time elapsed between VP's TCP SYN sent and HTTP response from front-end web server incorporating back-end database data received	yes
overall(null)	Time elapsed between VP's TCP SYN sent and HTTP response from front-end web server received	no

Table 3.3: *CLAudit measurements descriptions.*

Components

Vantage Points. Remotely-controlled VPs emulate real client appliances for interacting with Cloud-hosted applications and services. They collect, at various protocol layers, latency measurements of what end-users perceive when utilizing the Cloud. VPs are geographically dispersed to obtain end-user perspectives from around the globe. To maintain comparability, VPs are homogeneous in terms of OS and software (i686 Fedora Core 8 PlanetLab hosts). For the sake of redundancy and validation, VPs are deployed in triplets in every region (two VPs in a single campus and the third, backup VP in different campus nearby).

Cloud front-end. Servers that serve client requests (in our context, those which respond to VP requests). Front-ends are geographically dispersed across Microsoft DCs, emulating a globally-distributed Cloud application. They are implemented as Azure PaaS Web Apps inside Free Tier (a single shared instance without backup per DC). The application container combines static and dynamic webpages.

Cloud back-end. Servers that provide data when front-end servers need them to compose the client response (e.g. dynamic webpage in our context). The back-end is, by definition, not involved in every interaction, although Cloud applications often consist of both the front-end and the back-end. In our setup, several back-ends are co-located with front-ends within a single DC, whereas other back-ends reside in DCs elsewhere. That is to take into account scenarios like remote storage, georedundancy, etc. Each back-end server hosts an Azure SQL database inside Standard Tier at S1 performance level, allowing for 90 concurrent requests.

Monitor. The monitor is a single *central* entity, that gathers data from all VPs; instructs and adjusts measurement; and publishes, archives and analyzes the past and near-real-time data. The monitor is hosted on a high-end server on the premises of the Czech Technical University in Prague.

The request-response nature of many existing protocols allows for measuring RTTs and deriving latency. That is just what CLAudit does - in a continuous, simultaneous, large-scale distributed manner. Several measurement types, described in Tab. 3.3, are used for active probing between every *(VP, front-end)* and *(front-end, back-end)* pair. There is neither a front-end, nor a back-end selection algorithm, as all the pairwise combinations are measured. We thus get an idea of Internet, intra-DC and inter-DC latencies.

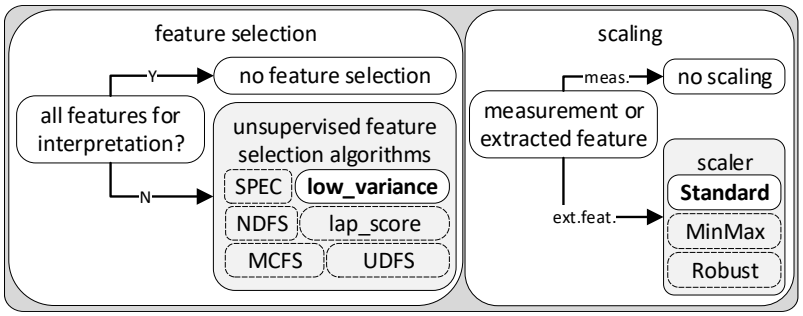


Figure 3.5: *Hi-Clust Step 2 : Data transformation.*

3.2 Data transformation

Our main focus is the interpretation of clustering results. Every feature/dimension of the dataset could potentially provide valuable information even if its values do not vary. Depending on the dataset characteristics, i.e. feature relation and dimensionality, we apply either (i) both feature selection and scaling techniques or (ii) neither and move to next step of Clustering. We view the feature selection and scaling tasks as optional, indicated by a grey color in the *Data transformation* diagram, Fig. 3.5. We explain the necessity of this task on the diverse high-dimensional MAWI dataset as opposed to lower dimensional CLAudit dataset, which includes only hierarchically related latency measurements.

The MAWI dataset consists of 197 features with various values. Abundant features complicate the interpretation; high dimensionality with a large range of values causes an issue with DBCV computation (see section 3.3.3) and diverse feature ranges could favor some features above the others in the clustering process ("curse of dimensionality" [66]). All of the mentioned issues (i) cluster interpretation, (ii) DBCV computation issues and (iii) curse of dimensionality can be solved by the reduction of the dataset dimensionality with feature selection and by scaling of the features. We cannot use standard dimensionality reduction methods like PCA (Principal Component Analysis) [67] or SVD (Singular Value Decomposition) [68] as they derive new feature subspace from existing features and we would loose the ability to interpret clusters by their features values, i.e. average and distribution. PCA and SVD are often referred to as feature extraction methods whereas we focus on feature selection methods. In general, there are two types of feature selection methods: (i) supervised and (ii) unsupervised. In our study we focus on the application of unsupervised machine learning methods and we analysed the effects of various unsupervised feature selection methods available in *Scikit-feature feature selection repository* [69] and scalers in Scikit [59]. We briefly evaluate the results in the next section. In

final algorithm evaluation, we use simple *Variance Threshold(low_variance)* feature selection and *Standard Scaler* feature standardization functions as we did not see significant improvement in the evaluation of clustering algorithms by use of more advanced methods.

The CLAudit dataset consists of 38 hierarchical latency measurements. Dimensionality reduction is not necessary and standardization of values would make interpretation step harder as we would lose *real world* values.

3.2.1 Feature selection algorithms

Feature selection is an effective preprocessing strategy to improve machine learning algorithm's performance by reducing high-dimensional data space to a simpler, more understandable dataset.

We a chose subset of unsupervised feature selection algorithms available in *"Scikit-feature" feature selection repository.* Feature selection algorithm results are parameter dependent with limited recommendations for their setting. In the following section, we analyse the algorithms ability to improve the performance of DBSCAN clustering. We compare internal evaluation metric DBCV between datasets with features filtered by feature selection algorithms and by standard procedure of the feature variance and correlation analysis. In the following, we give a brief description of the analysed feature selection algorithms, intuition behind them and their parameters. For detailed explanation, we refer the interested reader to original papers presenting the algorithms. For algorithms and their parameters summary see Tab 3.4.

Low variance

Simplest feature selection algorithm that removes features with variance below a given threshold.

Laplacian Score (LAP)

Similarity based unsupervised feature selection algorithm evaluating features based on their ability to partition the dataset. Algorithm can be summarized in three steps : (i) affinity matrix construction such that $S(i,j) = e^{-\frac{||x_i - x_j||^2}{t}}$ (ii) Diagonal $D(i,i) = \sum_{j=1}^{n} S(i,j)$ and Laplacian $L = D - S$ matrix calculation (iii) Laplacian score $L_r = \frac{\tilde{f_r}^T L \tilde{f_r}}{\tilde{f_r}^T D \tilde{f_r}}$ where $\tilde{f}_r = f_r - \frac{f_r^T D 1}{1^T D 1} 1$, $1 = [1, ..., 1]^T$. For detailed information we encourage the interested reader to see [70].

algorithm	input parameters
Low variance	**p** - variance threshold
LAP	**W** - input affinity matrix : (i) **k** - number of nearest neighbors (default = 5) (ii) **t** - weight function constant (default = 1)
SPEC	**style** - feature ranking function : (i) use all eigenvalues (ii) use all except the first eigenvalue (default) (iii) use the first k except first eigenvalue **W** - input affinity matrix (default = pairwise RBF kernel)
MCFS	**d** - number of features to select **k** - number of cluster **W** - (optional) input affinity matrix (same as LAP)
UDFS	**k** - number of clusters γ - objective function parameter (default=1) **p** - number of nearest neighbors
NDFS	α - objective function parameter (default=1) β - objective function parameter (default=1) γ - very large number used to force $F\hat{T}$ F = I **F0** - (optional) initialization of the pseudo label matrix F **clusters#** - number of clusters **W** - (optional) input affinity matrix

Table 3.4: *Feature selection algorithm input parameters*

Spectral Feature Selection (SPEC)

Extension of the LAP score that can be used also for supervised feature selection. SPEC uses RBF for vector similarity comparison. There are three available feature ranking functions using different Laplacian matrix eigenvalues, the default option (ii) is similar to LAP score, see Tab 3.4. For detailed information we encourage the interested reader to see [71]

Multi-Cluster Feature Selection (MCFS)

Algorithm uses Laplacian matrix constructed same as for LAP score. The algorithm computes the score as $MCFS(j) = \underset{k}{max}|a_{k,j}|$ where k is an algorithm parameter - number of clusters, and $a_{k,j}$ is the j-th element of the sparse coefficient vector a_k that we got from solving k l_1-regularized regression problems by Least Angle Regression (LAR) algorithm as defined by authors in [72].

Unsupervised Discriminative Feature Selection (UDFS)

Uses discriminative analysis and $l_{2,1}$-norm minimization to solve objective function $\underset{W^TW=I_d}{min} \sum_{i=1}^{n} tr[G_{(i)}^T H_{k+1} G_{(i)} - DS_{(i)}] + \gamma||W||_{2,1}$, where W is the linear classifier mapping input x_i to low dimensional space G_i, I is the identity matrix, G is the weighted label indicator matrix, , $G_{(i)}$ is a subset from G, γ is the regularization parameter, DS local discriminator score for x_i. For detailed information we encourage the interested reader

to see [73].

Nonnegative Discriminative Feature Selection (NDFS)

Algorithm uses spectral clustering simultaneously with feature selection. Redundant features are reduced by adding $l_{2,1}$-norm minimization constraint to objective function $\underset{G,w}{min}\, tr(G^T LG) + \beta(||XW - G||_G^2 + \alpha||W||_{2,1})$, such that $G^T G = I_n, G \geq 0$, where α and β are algorithm parameters, L is the Laplacian matrix derived from RBF, W is the linear transformation matrix and G is the weighted cluster indicator matrix. For detailed information we encourage the interested reader to see [74]

3.2.2 Scaling

Features extracted from network measurements have different scales depending on their type (eg. entropy vs average size of PDU headers) and usually contain outliers. Both characteristics lead to algorithm performance and computation speed degradation. We use well known feature scaling methods to overcome this issue. All three scalers transform feature ranges independently on each feature.

Standard

The most widely used scaler that removes feature average value and scales it to its variance.

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{3.2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.3}$$

MinMax

MinMax scales each feature to a given range, by default $(0, 1)$.

Robust

Scaler removes the feature median and scales it to IQR (Inter Quartile Range).

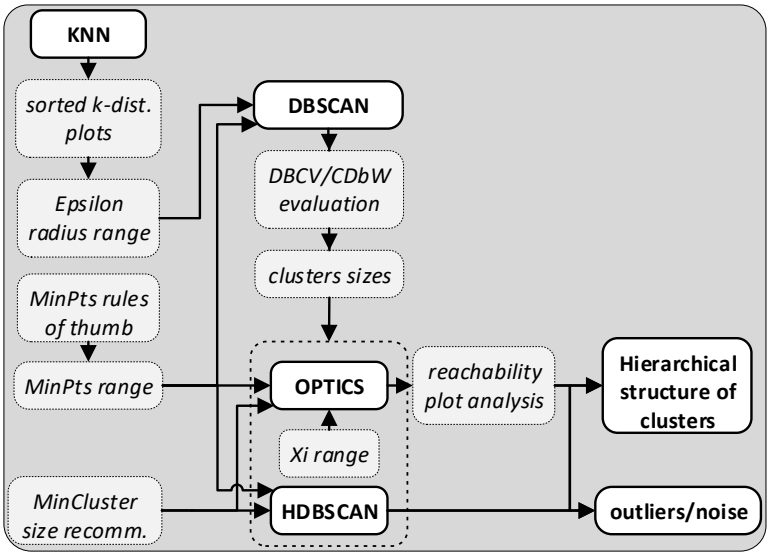


Figure 3.6: *Hi-Clust Step 3 : Clustering.*

3.3 Clustering

We use the original DBSCAN algorithm, its hierarchical variant HDBSCAN* and an extension OPTICS with *Xi* cluster extraction method. We do not use variants of algorithms to avoid confusion and to keep the number of algorithm parameters to a minimum. Fig. 3.6 shows an outline of the algorithm calibration process. Table 3.5 summarizes the algorithms parameters.

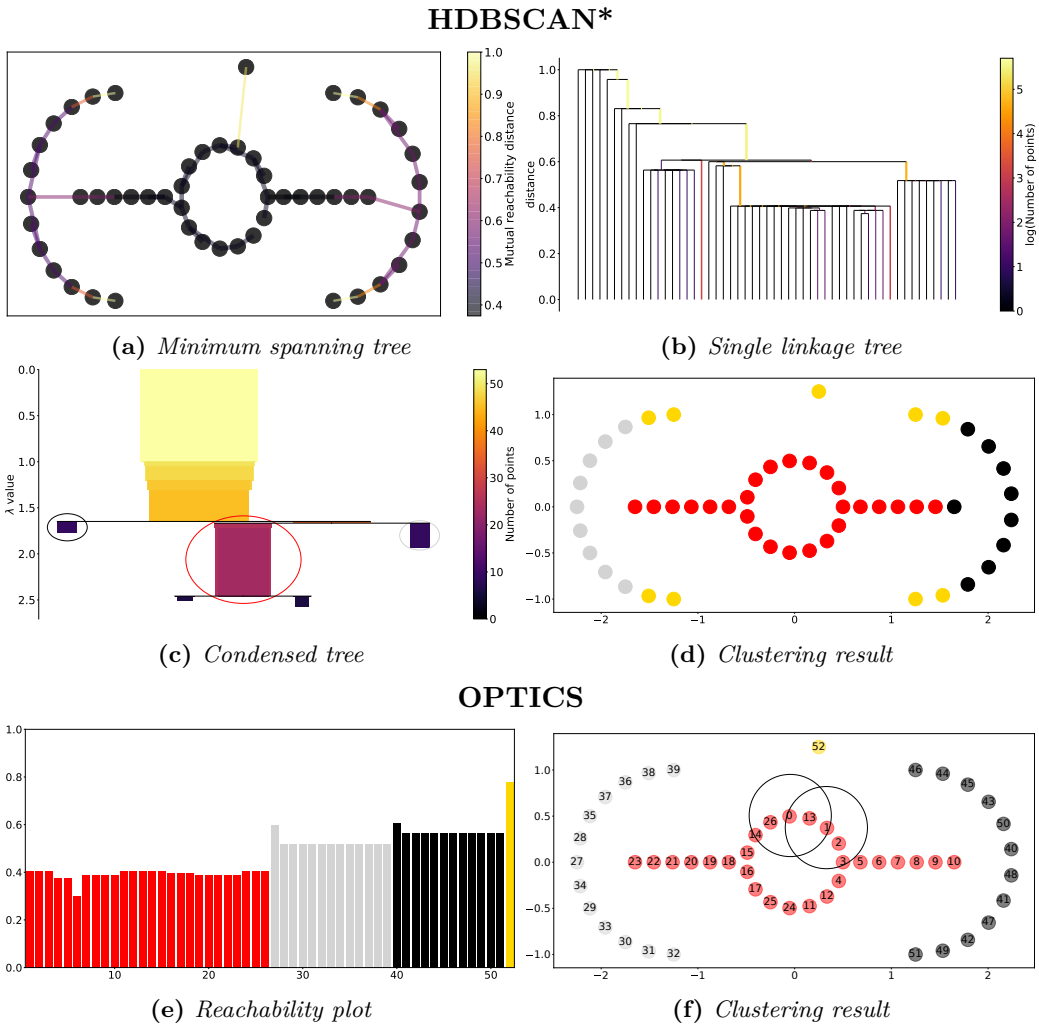
3.3.1 Algorithms

DBSCAN (Density-based spatial clustering of applications with noise)

Algorithm uses density level estimation by points with defined minimum number of neighbors - *MinPts*, within ϵ radius. Points adhering to this rule are called core points. Points that are not core, but are within radius of core point are *border points*. Everything else is noise.

algorithm	input parameters
DBSCAN	**MinPts** - min. number of points in the ϵ-neighborhood of a point ($minpts \in Z^+$) **ϵ(eps)** - maximum radius of the neighborhood to be considered
OPTICSXi	**MinPts** - same as DBSCAN **ϵ(eps)** - (optional) same as DBSCAN (∞ if not set) **ξ(xi)** - steepness req. for set of ordered points to be considered as beginning/end of the cluster **MinCl** - (optional) min. number of points in a set to become a cluster
HDBSCAN*	**MinPts** - same as DBSCAN **MinCl** - (optional) same as OPTICS

Table 3.5: *Algorithm input parameters*



(a) *Minimum spanning tree* **(b)** *Single linkage tree*

(c) *Condensed tree* **(d)** *Clustering result*

(e) *Reachability plot* **(f)** *Clustering result*

Figure 3.7: *Algorithms explanation*

HDBSCAN* (Hierarchical DBSCAN)

It is a hierarchical extension of DBSCAN that abandoned the concept of *border* points and uses the notion of minimum and single linkage spanning trees to identify the hierarchy of the most persistent clusters by an additional optional parameter of minimum extracted cluster size - *MinCl*. HDBSCAN* uses the Minimum Spanning tree build from mutual reachability distances between points to create a Single Linkage Tree (SLT) for all points. SLT is transformed to a Condensed tree by walking through the tree from top to bottom, keeping the largest clusters that adhere to MinCl parameter and rejecting the rest. The algorithm uses the notion of cluster stability $\sum_{p \in cluster}(\lambda_p - \lambda_{birth})$ where $\lambda = \frac{1}{mutual\ reachability\ distance}$, λ_p is a value of a point and λ_{birth} is a value at which the cluster was created. We demonstrate the algorithm notion on the example dataset in Fig. 3.7.

OPTICS (Ordering Points To Identify the Clustering Structure)

OPTICS is a density-based clustering algorithm extending DBSCAN [44], by producing a structure of augmented cluster-ordering of the database with respect to input parameters, up to generating distance ϵ. Generating distance is usually set to $max(\epsilon)$ or infinity. OPTICS does not explicitly generate clustering of the dataset. OPTICSXi extends the algorithm with parameter ξ, by defining ξ-*clusters* based on ξ-*steep areas* and ξ-*steep points* in the reachability plot (visualization of cluster ordering) of the dataset. We demonstrate the algorithm notion in Fig. 3.7 by displaying ordered reachability plot and clustered example dataset with corresponding order of samples in which they were processed and are displayed in the plot. For more information see [46].

3.3.2 Algorithms selection and parameters deduction

We start by setting MinPts value. Authors in [44, 75] recommend setting the value to $MinPts \geq 2{*}D$ whereas the research community often recommends $MinPts \geq D{+}1$ as a general rule (or rule of thumb). We narrow the range of MinPts parameter to two values. We denote them as *rule of thumb #1* ($MinPts = 2 * D$), *#2* ($MinPts = D + 1$). We set the $MinCl = 10$ to match the default CDbw value of cluster representative points. We set the range of ϵ values by plotting a sorted k-distance plot that shows the sorted distance of each point to k-th neighbor, as described by authors of DBSCAN, and according to their recommendations we search for a *knee* and explore the values between 70th and 95th percentile. We set the OPTICS ξ by analyzing values that are able to mark the start/end of interesting valleys identifying clusters in the reachability plot.

3.3.3 Performance evaluation techniques

V-measure

V-measure is a cluster labeling evaluation measure given a ground truth. We compare clustering results to ground truth extracted from MAWILab database in the case of MAWI dataset scenario. V-measure is a weighted harmonic mean of homogeneity and completeness. Homogeneity h indicates how well the clustering is able to assign member of a single class to a single cluster. Completeness c shows how well the clustering was able to assign all members of a single class to a single cluster. The measure is defined by formula :

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c} \tag{3.4}$$

, where β is set to 1 by default. If the β is set to > 1, the completeness is weighted more strongly than homogeneity, and vice versa if β is set to < 1.

DBCV (Density-Based Clustering Validation)

We use DBCV index [48] as a main indicator of cluster validity in the parameter calibration process of DBSCAN. DBCV produces values between -1 and 1 by comparing density within a cluster and between clusters. Higher values indicate better density-based clustering solutions. Negative values indicate lower density inside a cluster as compared to density separating it from other clusters. DBCV index of clustering $C = (C_i, N)1 \leq i \leq l$ containing l clusters, where n_i is the size of the i^{th} cluster, is based on definition of core density of the objects $a_{pts}coredist(o)$, interpretable as inverse density, defined by the formula :

$$a_{pts}coredist(o) = \left(\frac{\sum_{i=2}^{n_i} (\frac{1}{KNN(o,i)})^d}{n_i - 1} \right)^{-\frac{1}{d}} \tag{3.5}$$

$$V_C(C_i) = \frac{min_{1 \leq j \leq l, j \neq i}(DSPC(C_i, C_j)) - DSC(C_i)}{max(min_{1 \leq j \leq l, j \neq i}(DSPC(C_i, C_j)), DSC(C_i))} \tag{3.6}$$

$$DBCV(C) = \sum_{i=1}^{i=l} \frac{|C_i|}{|O|} V_C(C_i) \tag{3.7}$$

, where d represents the dimensionality of the dataset $O = o_1, ..., o_n$ containing n objects and $KNN(o, i)$ is the distance between object o and its i^{th} nearest neighbor. Density Sparseness of a Cluster (DSC) and Density Separation of a Pair of Clusters (DSPC) define density within and between clusters for Validity Index of a Cluster $V_C(C_i)$. The definition of $a_{pts}coredist(o)$ could create an issue in the implementation as of maximum supported float value, if the dimensionality is too high or objects are either too close or too far apart.

CDbw (Compose Density between and within clusters)

CDbw takes multiple points per cluster to consider their arbitrary shape and compute within and between cluster density. This creates a demand for additional mechanisms to choose the appropriate value that is common for all clusters. By default, the number of representative points is set to 10 according to authors' claim that it sufficiently captures the shape of the clusters [55]. According to [48], CDbw is the most used measure specific to density-based validation. The main idea of CDbw is that good clustering characteristics C should be reflected in Cohesion, Compactness and Separation of clusters by following formulas :

$$Separation(C) = \frac{\frac{1}{c}\sum_{i=1}^{c} min_{j=1,\dots,c;j\neq i}\{Dist(C_i, C_j)\}}{1 + Inter\_dens(C)} \quad ; c > 1, c \neq n \tag{3.8}$$

$$Compactness(C) = \frac{\sum_s Intra\_dens(C, s)}{s} \tag{3.9}$$

$$Cohesion(C) = \frac{Compactness(C)}{1 + Intra\_change(C)} \tag{3.10}$$

$$CDbw(C) = Cohesion(C)\ .\ Separation(C)\ .\ Compactness(C) \quad ; c > 1 \tag{3.11}$$

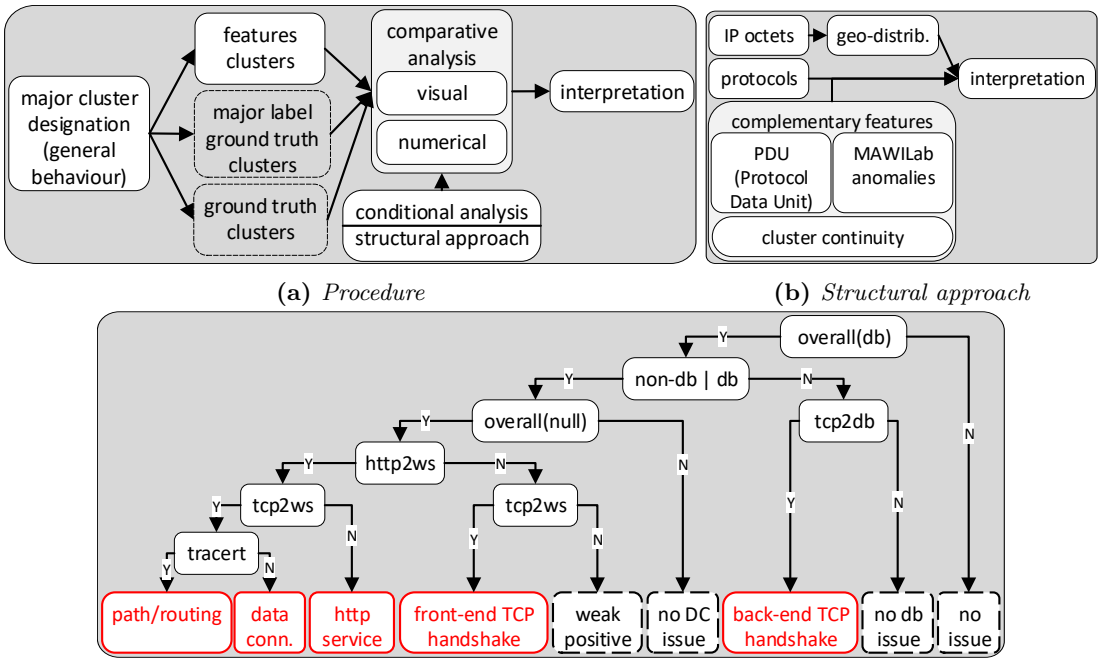
, where $Intra\_change(C)$ denotes a change of density within clusters, $Inter\_dens(C)$ a maximum density between the clusters, $Intra\_dens(C, s)$ relative density within cluster wrt. shrinking factor s and $Dist(C_i, C_j)$ a distance between clusters C_i and C_j. We consider this approach inferior to DBCV and use it as a complementary indicator of cluster validity.

We compute DBCV and CDbw only for DBSCAN as the unsupervised machine learning evaluation metrics are unable to address the hierarchical aspect of HDBSCAN* and OPTICS.

3.3.4 Clustering hierarchy and outliers analysis

We confirm the feasibility of the clustering results by analysing :

- **Total number of identified clusters.** Low number of DBSCAN clusters suggests underlying hierarchical structure, whereas high number suggests too restrictive choice of parameters resulting in many small, dense clusters [47];
- **Parent-child cluster relations.** Parent-child relations are visible in the OPTICS reachability plot by valleys inside a valley and in HDBSCAN* by branching of Con-



(a) *Procedure* **(b)** *Structural approach*

(c) *Conditional approach (Decision Tree)*

Figure 3.8: *Hi-Clust Step 4 : Cluster interpretation.*

densed tree. Identification of parent-child relations improves the interpretation by identifying main suspicious event characteristics and its *flavors* in child clusters;

- **Continuity of the clusters.** Continuity of the cluster shows if the event is recurring or exclusive;
- **Magnitude of identified outliers.** Outliers magnitude shows ability of algorithms to identify clustering structure.

We use the analysis results also for the interpretation of clusters by a structural approach when we do an *open interpretation* of the MAWI clustering results by traffic characteristics and expert domain knowledge. We discuss this in the following, last, Hi-Clust step of *Interpretation*.

3.4 Cluster interpretation

We identify *General behaviour* as the largest cluster that is not *noise* and compare the features of all remaining clusters with the general behaviour. Density-based clustering creates arbitrarily shaped clusters. We address it by comparing the distribution of suspicious features to *general behaviour* by box plots. We do not use features with significantly

wider distributions than *general behaviour* in Interpretation as they do not show an explicit pattern for a particular cluster, see *visual* in Fig. 3.8 a). We also compare features mean values of all remaining clusters with the general behaviour and mark the feature as suspicious if the value differs from the general behaviour by an empirically determined threshold, see *numerical* in Fig. 3.8 a). Eventually, we interpret the clusters by *structural* Fig. 3.8 b) or *conditional approach* Fig. 3.8 c).

Structural approach

The interpretation of MAWI clusters relies only on a structural approach. We use expert domain knowledge, i.e. internet traffic services characteristics, to interpret traffic categorized to clusters. IP octets entropy provides geo-distribution information of sources/destinations or possible multi-point traffic. PDU and protocol information reveals major attributes of the traffic and cluster continuity shows whether specific behaviour is recurring or happened in a burst. We use previously identified MAWILab anomalies as a complementary indicator of internet traffic characteristics.

Conditional approach

We extend the structural approach in CLAudit cluster interpretation by expert domain knowledge *Decision Tree*, crafted upon the inter-dependencies among the different layers/measurements considered. The Decision Tree takes into account the standardized protocol hierarchy and associated inter-dependencies between network and remote computation components to explain the potential causes. Clusters are interpreted only if marked suspicious features adhere to a set of conditions defined by the Decision Tree. Fig. 3.8 c) depicts the interpretation tree used by CLAudit. For example, abrupt latency changes due to path or routing issues would necessarily be reflected at all the different layer measurements, from ICMP to HTTP, whereas HTTP-related service issues such as large page elaboration times would not necessarily be visible at the transport (TCP) or connectivity (ICMP) levels.

Conclusion

In this section, we provided a detailed description of the Hi-Clust approach. We provided an intuition behind each step and described how the approach can be applied, in general, to open and closed cluster interpretation. In the next section, we apply the

procedures to two distinct scenarios of MAWI and CLAudit. We describe the datasets and their features, analyse clustering algorithm results and interpret the clusters and results.

Chapter 4

Application and Evaluation

We present two scenarios of Hi-Clust methodology application. We used Cloudera distributed computing environment with PySpark for the data gathering and feature extraction and Python with complementary RStudio libraries [59, 76, 77] for clustering performance evaluation and analysis of the results. First, we discuss the clustering results difference from the ground truth datasets we extracted from MAWILab anomalies and then we apply the Hi-Clust methodology to both MAWI and CLAudit datasets.

4.1 Evaluation methodology

As we mentioned earlier, we focus on the evaluation and interpretation of unsupervised machine learning methods and we do not possess ground truth in a "true sense". We extracted and transformed the MAWILab anomaly database to serve as a reference point and for the interpretation of the clustering results. We computed anomaly binary datasets evaluation metrics and compared them to the best DBSCAN clustering results:

- **major anomaly ground truth (binary)** : DBCV = -0.881, CDbw = 8.85-e07;
- **ground truth (binary)** : DBCV = -0.812, CDbw = 1.13-e05;
- **DBSCAN** : DBCV = 0.697, CDbw = 3.56-e06.

CDbw values are informative, as we will show in following sections, CDbw does not provide conclusive evaluation results in the scenarios. From the definition of DBCV, we conclude that clusters formed by binary anomaly datasets form suboptimal partitioning, as negative values indicate lower cluster intra-density compared to inter-density. MAWILab contains a database of anomalies that were obtained using a graph-based methodology

comparing and combining various independent anomaly detectors. From the DBCV results we see that the density-based clustering approach identifies significantly better dataset partitioning.

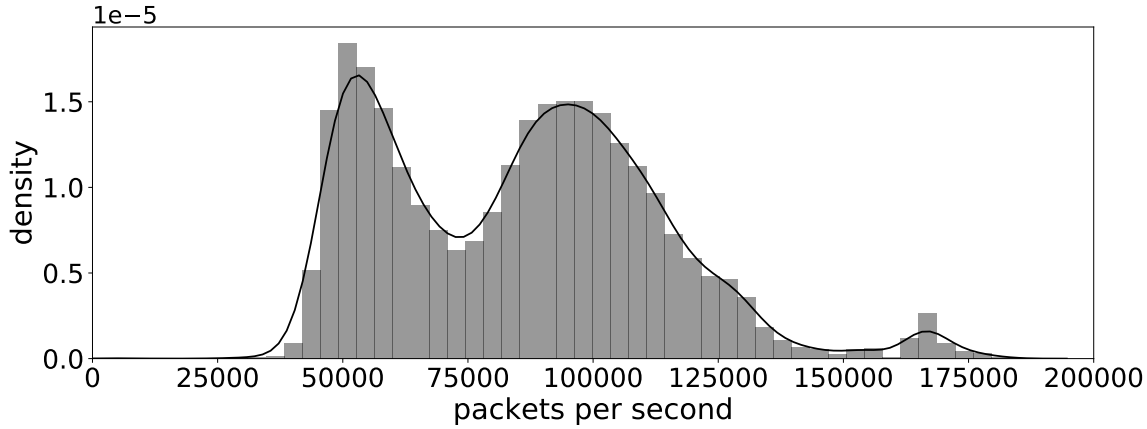


Figure 4.1: *MAWI number of packets per seconds.*

4.2 Scenario I - MAWI

4.2.1 Dataset

We extracted features from one month of MAWI traces. We chose specifically May 2016 based on the high diversity of detected MAWILab anomalies implying a possibility of a high number of interpretable clusters. The dataset includes 22500 points of 197 dimensions filtered to 106. The dataset is missing 6 days due to their unavailability. Each 106 dimensional point in the dataset represents $\sim$ 80000 traces captured in a second time interval. In Fig. 4.1, we show the distribution of the number of packets per second in a studied dataset. We used a histogram of nomalized packet counts, grey bins, that we approximated by nonparametric Gaussian Kernel Density Estimation (KDE), black line. The diagram gives us an understanding of the traffic magnitude represented by extracted features in each 1 second time interval. There are 900 points per day (15 minutes = 900 seconds), per 25 days. MAWI anomaly datasets consist of 24 (major anomaly dataset) and 88 (anomaly dataset) clusters. Each cluster represents a combination of anomalies.

Fig. 4.2 shows the distribution of the extracted features and their ability to characterize MAWI network traffic. We defined and extracted features that characterize network traffic in general. We see that most of the statistical features, i.e. percentiles, minimum, maximum and standard deviation, have narrow range of values and are not good choices for MAWI traffic characterization. Specific features increase in importance depending on the dataset and its traffic in a given time interval, e.g. fragmented packets, ICMP type fractions, etc. might increase in anomalous network environments like Darknet traces [78]. Features in a final dataset were filtered out based on their low variance and narrow distribution in the dataset. We followed the designed approach and applied

StandardScaler on the dataset to avoid prioritization of features.

We examined the MAWILab anomaly datasets. Anomaly graphs in Fig. 4.3 show that considering all anomaly categories results in large number of unique clusters with small sizes. We can see that anomalies from Multipoint, Alpha Flow and HTTP major categories are represented in the datasets concurrently from multiple source IPs in 1 second time intervals. In the anomaly durations graphs, we see that most of the anomalies, if present in a daily trace capture, are present for the whole 15 minute daily trace. In conclusion, we suspect this could give us a biased view on the interpretations of the detected clusters and we use the anomaly datasets only as complementary interpretation information.

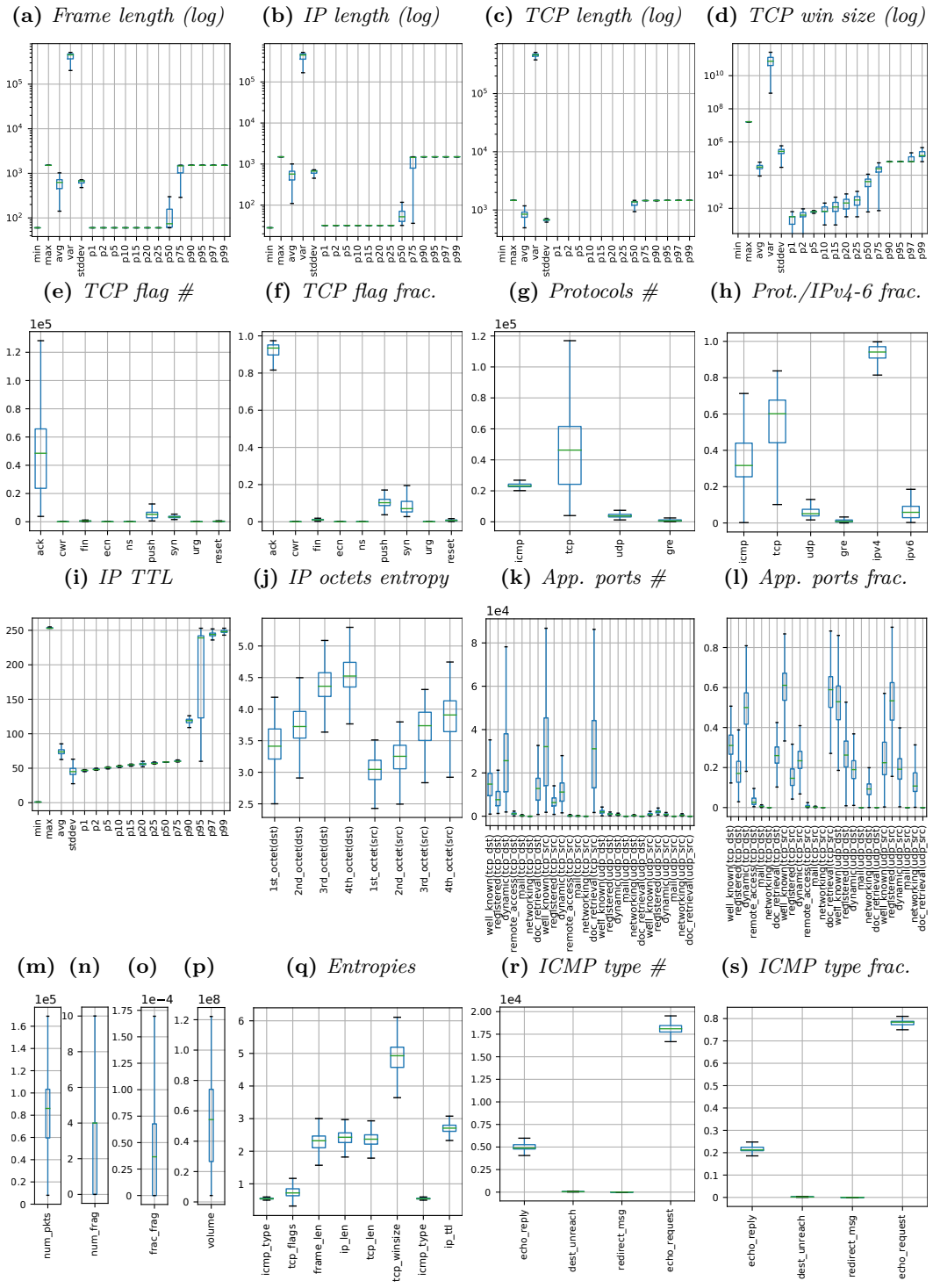


Figure 4.2: *MAWI feature distributions.*

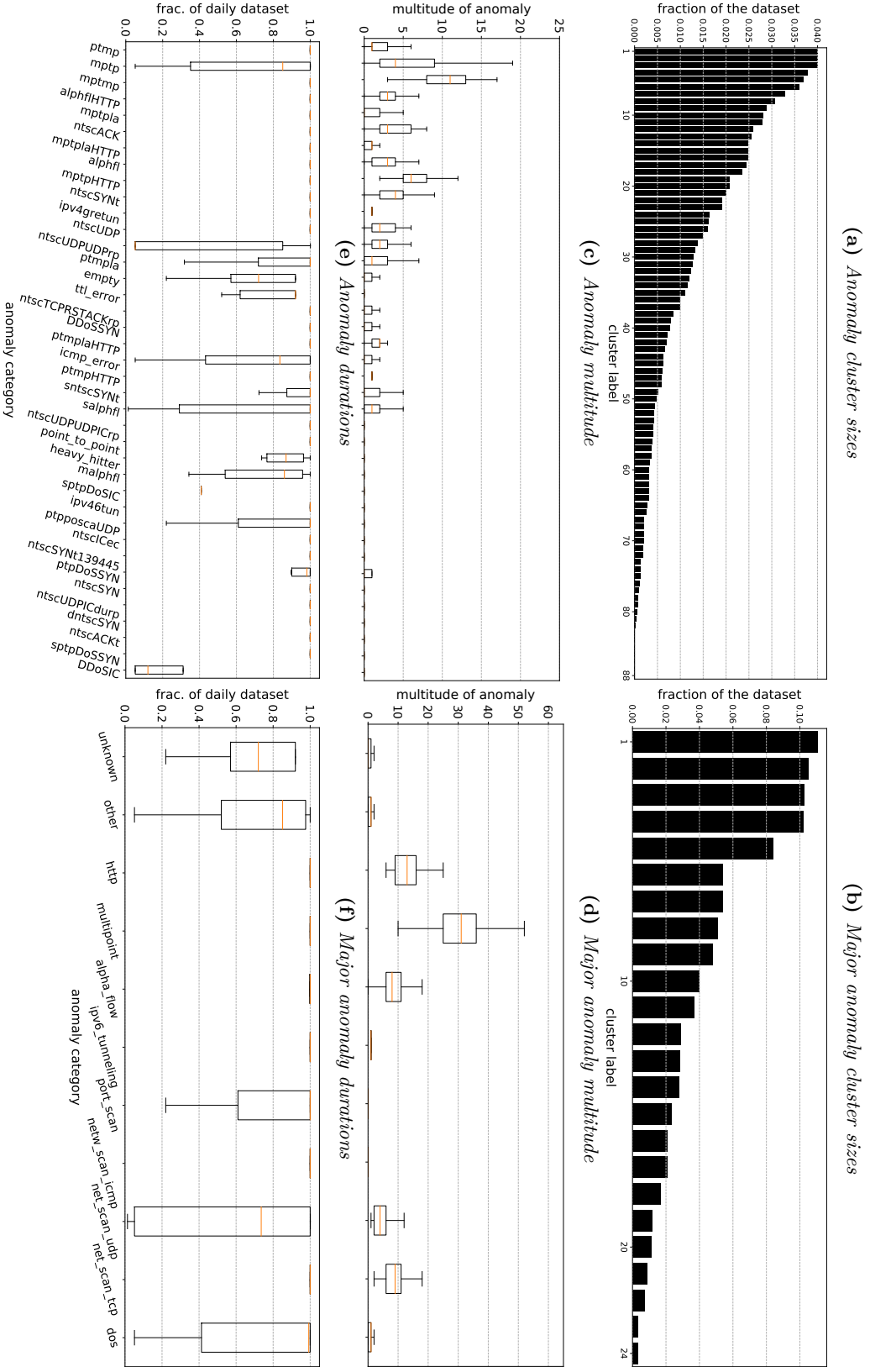


Figure 4.3: *MAWIlab anomalies.*

dataset	num. of features	DBCV
all features	197	0.08
LAP < 1	102	0.37
LAP < p50	91	0.38
LAP < p25	46	0.67
LAP < p10	19	0.73
pre-processed	**106**	**0.69**

Table 4.1: *MAWI datasets best DBCV scores.*

4.2.2 Feature selection evaluation

First, we filtered out single value features, i.e. features with zero variance, marked by red color in Tab. 4.2. Then, we analyzed the correlation of remaining features by heatmap, see Fig. 4.4. We removed an additional 77 features with a combination of variance, correlation and feature distribution values from Fig. 4.2. Resulting dataset includes 106 features and is marked as *pre-processed* in Tab. 4.1. Filtered features are marked by grey color in Tab. 4.2. We applied LAP, SPEC, MCFS, UDFS and NDFS feature selection algorithms with default parameter values on the dataset. As expected, SPEC resulted in comparable feature ordering as LAP score. Algorithms MCFS, UDFS and NDFS with default parameters resulted in zero scores for all features. We suspect that detailed fine-tuning of the algorithms is necessary, in order to obtain non-zero results. Unfortunately, there are no guidelines for parameter setting, including non-intuitive knowledge of the expected number of clusters in the results. We computed the DBCV score of the full feature dataset and its subsets and included the results in the Tab. 4.1. We calibrated DBSCAN for each feature subset, see Sec. 3.3.2. We included four feature subsets filtered by LAP score : (i) features with LAP score lower than 1 (max value), (ii) lower than 50th percentile, (iii) 25th percentile and (iv) 10th percentile. The dataset obtained by LAP score lower than p10 achieved slightly better DBCV result than *Low variance, distribution analysis, correlation*, but the resulting combination of features is not interpretable by expert domain knowledge. As mentioned previously in Sec. 3.3.3, we are often unable to compute DBCV due to the high dimensionality of the full feature dataset as the $a_{pts}coredist(o) = \left(\frac{\sum_{i=2}^{n_i} (\frac{1}{KNN(o,i)})^d}{n_i - 1} \right)^{-\frac{1}{d}}$, where the dimensionality of the dataset $d = 197$ creates an issue with maximum supported float value in operating systems. For further analysis, we used *Low variance, distribution analysis, correlation* dataset.

	features 1-40		features 41-80		features 81-120		features 121-160		features 161-197	
1	max(tcp_len)	0	frac_tcp_flag(syn)	5.00E-02	min(ip_ttl)	4.91E-01 121	num_dest_unr(icmp)	3.35E+01	num_echo_request(icmp)	1.37E+03
2	p2(frame_len)	0	entropy(icmp)	6.19E-02	entropy(tcp_winsize)	4.96E-01 122	num_tcp_flag(cwr)	5.36E+01	num_well_known(udp_src)	1.41E+03
3	min(frame_len)	0	frac_doc_ret(tcp_src)	6.34E-02	entropy_4th_ip_oct(dst)	5.14E-01 123	num_tcp_flag(ecn)	5.43E+01	num_tcp_flag(syn)	2.02E+03
4	max(frame_len)	0	frac_reg_ports(tcp_src)	6.38E-02	max(ip_ttl)	5.74E-01 124	p95(ip_ttl)	5.82E+01	num_tcp_flag(push)	2.65E+03
5	p2(tcp_len)	0	frac_dyn_ports(tcp_src)	7.04E-02	num_tcp_flag(urg)	5.87E-01 125	p20(tcp_len)	5.85E+01	num_echo_reply(icmp)	3.27E+03
6	p5(frame_len)	0	frac_well_known(tcp_src)	7.50E-02	p99(tcp_len)	6.11E-01 126	stddev(frame_len)	7.08E+01	p25(tcp_winsize)	3.42E+03
7	p10(frame_len)	0	frac_echo_reply(icmp)	7.80E-02	p50(ip_ttl)	7.45E-01 127	p75(tcp_len)	7.25E+01	num_reg_ports(tcp_src)	3.50E+03
8	p1(tcp_len)	0	frac_echo_request(icmp)	7.85E-02	p1(ip_len)	8.21E-01 128	p95(frame_len)	7.34E+01	num_icmp(ip_proto)	3.71E+03
9	max(ip_len)	0	frac_dyn_ports(udp_dst)	8.03E-02	p25(ip_ttl)	1.16E+00 129	stddev(ip_len)	7.80E+01	num_dyn_ports(tcp_src)	5.36E+03
10	min(tcp_len)	0	frac_reg_ports(tcp_src)	8.51E-02	p99(frame_len)	1.30E+00 130	p95(ip_len)	8.13E+01	num_reg_ports(tcp_src)	5.57E+03
11	p10(tcp_len)	0	frac_well_known(tcp_src)	9.86E-02	num_tcp_flag(ns)	1.32E+00 131	p25(frame_len)	9.59E+01	num_doc_ret(tcp_src)	6.36E+03
12	p5(tcp_len)	0	frac_netw(udp_src)	1.07E-01	p20(ip_ttl)	1.45E+00 132	p10(tcp_winsize)	1.09E+02	num_well_known(tcp_src)	6.60E+03
13	min(tcp_winsize)	0	frac_doc_ret(tcp_src)	1.10E-01	p15(ip_len)	1.53E+00 133	p25(ip_len)	1.10E+02	num_well_known(udp_dst)	6.93E+03
14	p1(frame_len)	0	frac_dyn_ports(tcp_src)	1.14E-01	p5(ip_ttl)	1.56E+00 134	p90(frame_len)	1.10E+02	p50(tcp_winsize)	1.09E+04
15	frac_mail(udp_dst)	4.65E-06	num_doc_ret(udp_src)	1.17E-01	p15(ip_ttl)	1.60E+00 135	p90(ip_len)	1.33E+02	num_reg_ports(udp_src)	1.19E+04
16	frac_tcp_flag(urg)	1.16E-05	frac_icmp(ip_proto)	1.17E-01	p10(ip_ttl)	1.66E+00 136	avg(tcp_len)	1.34E+02	num_dyn_ports(udp_src)	1.19E+04
17	frac_doc_ret(udp_src)	2.40E-05	frac_dyn_ports(udp_src)	1.20E-01	p20(frame_len)	1.67E+00 137	num_tcp_flag(reset)	1.37E+02	num_dyn_ports(tcp_src)	1.63E+04
18	frac_tcp_flag(ns)	4.01E-05	frac(ip_proto)	1.21E-01	p99(ip_len)	1.76E+00 138	p25(tcp_len)	1.47E+02	num_doc_ret(tcp_src)	1.83E+04
19	frac_netw(tcp_src)	3.11E-04	entropy(tcp_flags)	1.48E-01	num_mail(udp_src)	1.95E+00 139	avg(ip_len)	1.60E+02	num_well_known(tcp_src)	1.83E+04
20	frac_netw(tcp_src)	3.12E-04	frac(ip_proto)	1.48E-01	p2(ip_ttl)	2.23E+00 140	avg(frame_len)	1.61E+02	num_reg_ports(udp_dst)	1.87E+04
21	frac_mail(udp_src)	3.89E-04	frac_reg_ports(udp_src)	1.52E-01	p99(ip_ttl)	2.40E+00 141	num_mail(tcp_src)	1.93E+02	num(ip_proto)	1.98E+04
22	frac_tcp_flag(cwr)	1.35E-03	frac_well_known(udp_src)	1.58E-01	p1(ip_ttl)	2.64E+00 142	num_red_mes(icmp)	2.12E+02	num(ip_proto)	2.15E+04
23	frac_tcp_flag(ecn)	1.36E-03	frac_well_known(udp_dst)	1.59E-01	p97(tcp_len)	2.83E+00 143	num_rem_acc(tcp_src)	2.39E+02	num_tcp_flag(ack)	2.39E+04
24	frac_frag	3.53E-03	frac_reg_ports(udp_dst)	1.63E-01	p20(ip_len)	4.07E+00 144	p15(tcp_winsize)	3.08E+02	p75(tcp_winsize)	2.55E+04
25	frac_mail(tcp_src)	3.93E-03	entropy(tcp_len)	2.21E-01	p95(tcp_len)	4.71E+00 145	p50(ip_len)	3.14E+02	avg(tcp_winsize)	2.64E+04
26	frac_tcp_flag(reset)	4.24E-03	p15(frame_len)	2.58E-01	avg(ip_ttl)	5.10E+00 146	num_doc_ret(udp_dst)	3.36E+02	num_pkts(frame_len)	2.83E+04
27	frac_red_mes(icmp)	6.07E-03	entropy(ip_len)	2.76E-01	stddev(ip_ttl)	7.07E+00 147	num_dyn_ports(udp_dst)	3.53E+02	var(tcp_len)	3.23E+04
28	frac_dest_unr(icmp)	6.27E-03	entropy(ip_ttl)	2.80E-01	num_netw(tcp_src)	9.04E+00 148	num_tcp_flag(fin)	3.78E+02	p90(tcp_winsize)	5.47E+04
29	frac_tcp_flag(fin)	6.42E-03	entropy(frame_len)	2.90E-01	num_netw(tcp_src)	9.35E+00 149	num_frag	3.82E+02	var(frame_len)	8.29E+04
30	frac_mail(tcp_src)	7.77E-03	p5(ip_len)	3.13E-01	p90(tcp_len)	1.24E+01 150	p50(frame_len)	3.97E+02	var(ip_len)	9.12E+04
31	frac_doc_ret(udp_dst)	8.32E-03	p10(ip_len)	3.21E-01	p75(ip_ttl)	1.39E+01 151	p50(tcp_len)	4.84E+02	p95(tcp_winsize)	1.05E+05
32	frac_rem_acc(tcp_src)	9.49E-03	entropy_1st_ip_oct(src)	3.59E-01	p1(tcp_winsize)	1.66E+01 152	p75(frame_len)	5.16E+02	p97(tcp_winsize)	1.46E+05
33	frac_gre(ip_proto)	1.31E-02	p2(ip_len)	3.88E-01	p2(tcp_winsize)	1.69E+01 153	num_mail(tcp_src)	5.47E+02	p99(tcp_winsize)	4.50E+05
34	num_mail(udp_dst)	2.11E-02	min(ip_len)	3.92E-01	p15(tcp_len)	1.81E+01 154	num_netw(udp_dst)	5.57E+02	stddev(tcp_winsize)	5.10E+05
35	frac_rem_acc(tcp_src)	2.82E-02	entropy_2nd_ip_oct(src)	3.97E-01	p97(frame_len)	1.95E+01 155	p75(ip_len)	5.61E+02	max(tcp_winsize)	2.31E+07
36	frac_tcp_flag(push)	2.92E-02	entropy_4th_ip_oct(src)	4.05E-01	stddev(tcp_len)	2.48E+01 156	var(ip_ttl)	6.55E+02	vol(frame_len)	2.93E+07
37	frac_netw(udp_dst)	4.80E-02	entropy_3rd_ip_oct(src)	4.24E-01	p97(ip_ttl)	2.86E+01 157	p20(tcp_winsize)	7.61E+02	var(tcp_winsize)	1.84E+13
38	frac_ipv6	4.87E-02	entropy_1st_ip_oct(dst)	4.29E-01	p97(ip_len)	3.00E+01 158	num_gre(ip_proto)	8.95E+02		
39	frac_ipv4	4.87E-02	entropy_2nd_ip_oct(dst)	4.49E-01	p90(ip_ttl)	3.12E+01 159	num_netw(udp_src)	9.78E+02		
40	frac_tcp_flag(ack)	4.93E-02	entropy_3rd_ip_oct(dst)	4.91E-01	p5(tcp_winsize)	3.18E+01 160	num_rem_acc(tcp_src)	1.14E+03		

Table 4.2: *MAWI features sorted by their variance.*

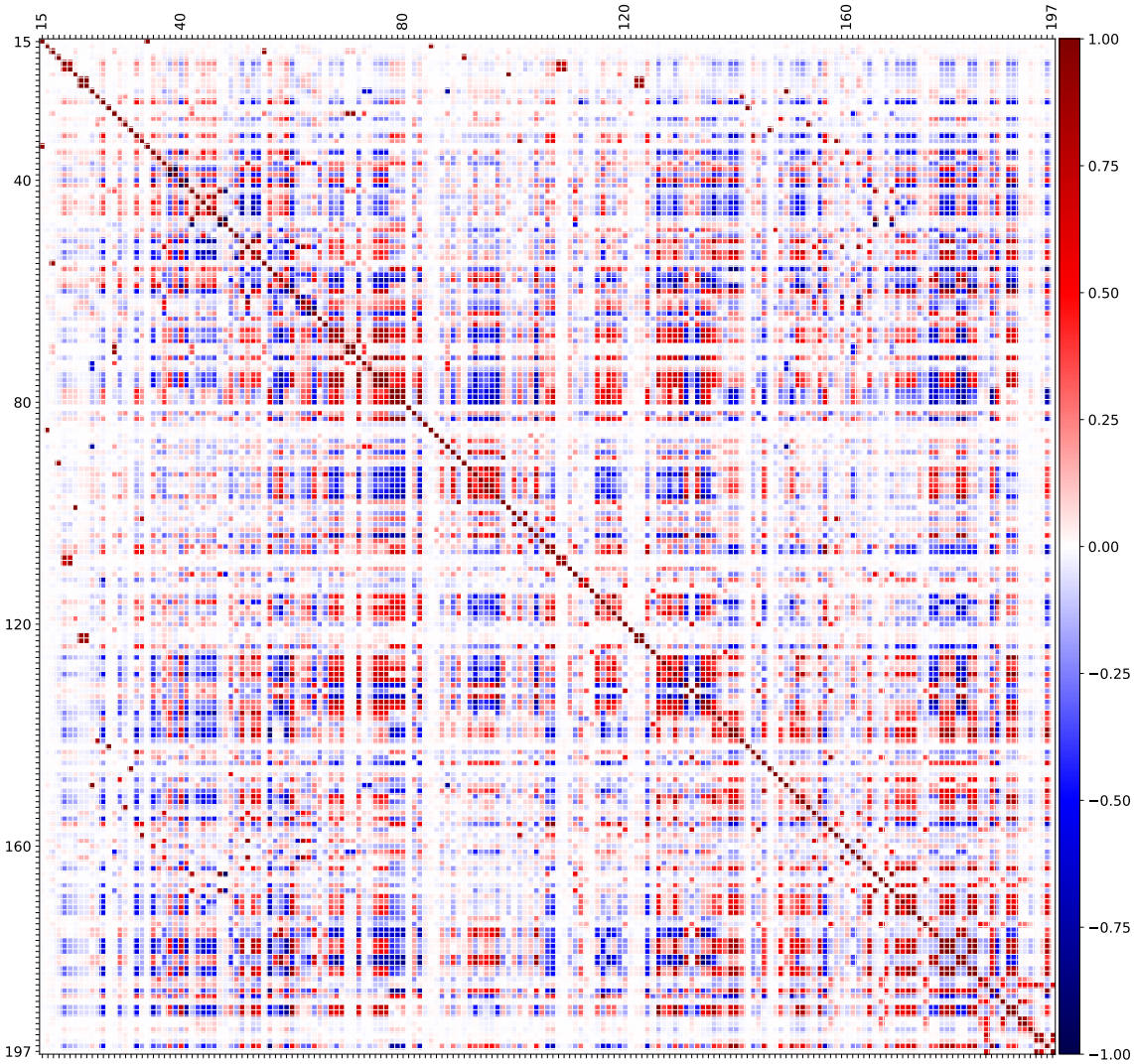


Figure 4.4: *MAWI features correlation heatmap.*
Features are ordered according to Tab.4.2. Single value features 1-14 are not included.

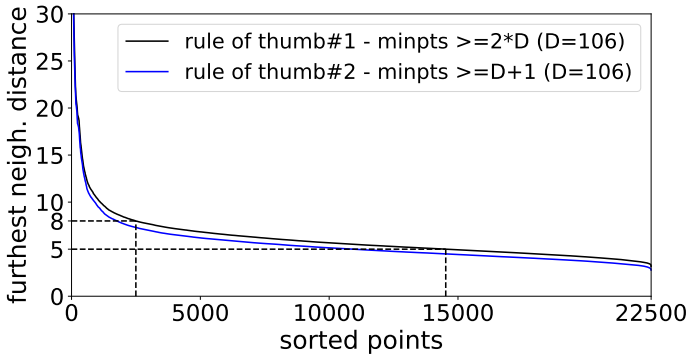


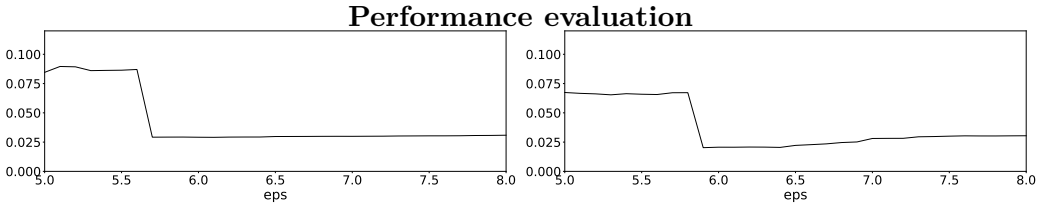
Figure 4.5: *MAWI DBSCAN k-dist plots.*

4.2.3 Clustering

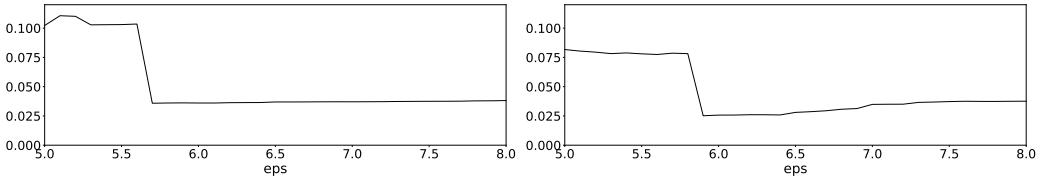
We calibrate, evaluate and interpret the clustering results. First, we analyse the effect of $MinPts$ and ϵ on DBCV and V-measure [60], then we proceed with hierarchical clustering analysis.

DBSCAN

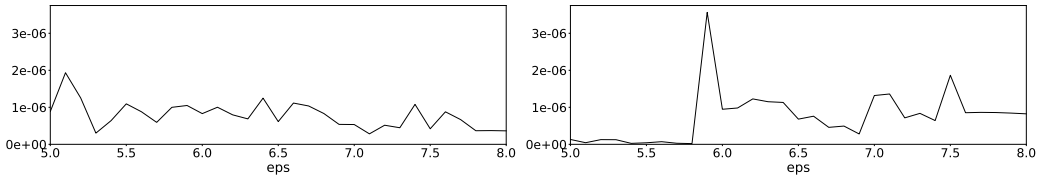
In Fig. 4.5, we show k-dist plots used to select range of ϵ values. We focus on the lower *knee* interval values $< 5; 8 >$ indicating 70th and 95th percentile. As we described in the previous chapter, we use two MinPts values defined as *rule of thumb #1 and #2*. In Fig. 4.6 a)-h) we analyze the impact of ϵ on clustering results through evaluation metrics. V-measure shows better results for small ϵ as the DBSCAN creates more clusters resembling the large number of clusters in the anomaly ground truth datasets, see Fig. 4.6 i). In general, V-measure and DBCV evaluation results for both anomaly datasets and both MinPts values are very similar. We observed that $MinPts$ does not have a significant impact on results, but the higher value (*rule of thumb #2*) gives slightly better results. We did not observe clear characteristics of the CDbw diagram and chose the best ϵ value according to DBCV metric, $\epsilon = 5.9, Minpts = 212$. In Fig. 4.6 j) we also see that the $\epsilon = 5.9$ is a "breaking point" for magnitude of marked outliers. The algorithm extracts three large clusters with parameters set to $\epsilon = 5.9, Minpts = 212$.



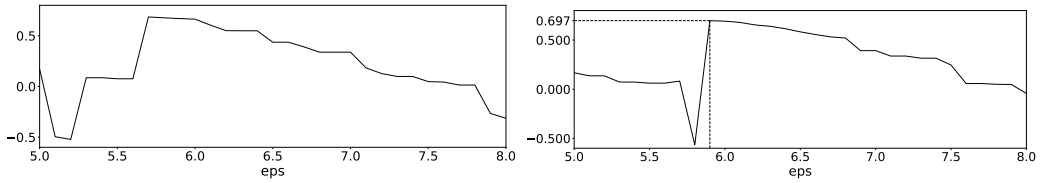
(a) *V-measure (major anomalies), MinPts=107* **(b)** *V-measure (major anomalies), MinPts=202*



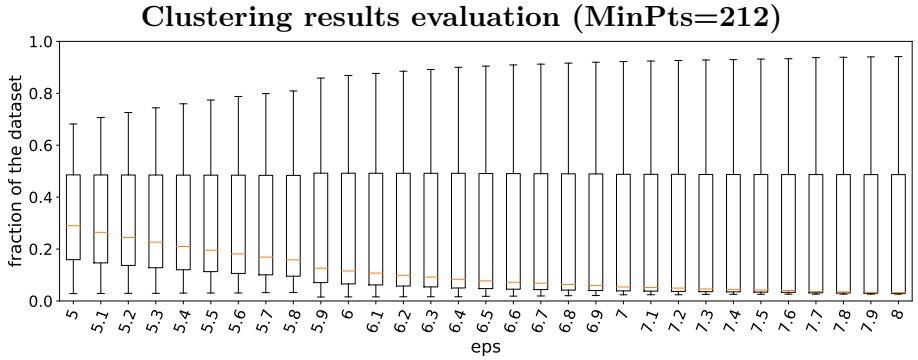
(c) *V-measure (anomalies), MinPts=102* **(d)** *V-measure (anomalies), MinPts=212*



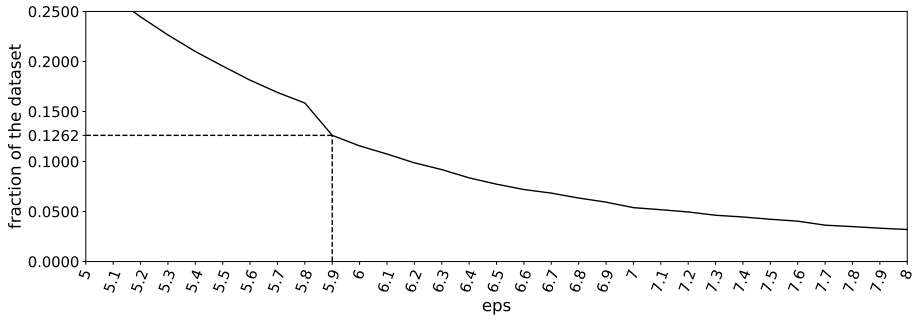
(e) *CDbW, MinPts=107* **(f)** *CDbW, MinPts=212*



(g) *DBCV, MinPts=107* **(h)** *DBCV, MinPts=212*



(i) *Cluster size distributions*



(j) *Outliers magnitude*

Figure 4.6: *MAWI DBSCAN evaluation.*

We summarize the identified cluster characteristics in comparison to the largest cluster, i.e. General behaviour. We use the default Scikit cluster numbering scheme, in which -1 denotes outliers, i.e. group of points remarkably different from the rest of the dataset that do not form clusters. Tab. 4.3 presents summary of DBSCAN results for $Minpts = 212, \epsilon = 5.9$. The general behaviour cluster is marked by green cells. We empirically differentiate two thresholds for upper and two thresholds for lower values in comparison to General behaviour. We transformed the feature dataset by StandardScaler which standardizes features by removing their average and scaling them to unit variance. Therefore, we expect the largest cluster to have average feature values close to 0. We set the upper threshold values to 0.25 and 0.5 denoted by light and dark red in Tab. 4.3a. We set the lower threshold values to -0.25 and -0.5 denoted by light and dark blue in Tab. 4.3a. For complementary interpretation by MAWILab anomalies, we use *non-binary* anomaly datasets that provide information about the distribution of each anomaly type in a cluster. The anomaly datasets were not standardized. We take the largest cluster as a baseline. We compare other clusters by their fractional increase or decrease in the average multitude of anomalies to the largest cluster and again empirically differentiate two thresholds of upper and two thresholds of lower values in comparison to General behaviour. We use the same color scheme as for the feature table with upper thresholds set to 1.05 and 1.10 and lower thresholds set to 0.95 and 0.90 representing 5/10% increase/decrease of average multitude compared to General behaviour. For the distributions of features and anomalies in each cluster, we refer the interested reader to Appendix Fig. E.1-6. Each plot includes cluster and general behaviour values for comparison. General behaviour is plotted by light grey color and cluster values by bright red for clear visual comparison. We summarize the tables and distributions by interpreting clusters in the final interpretations table Tab. 4.7.

We proceed to analyze possible hidden hierarchical clustering structures by HDBSCAN* and OPTICS.

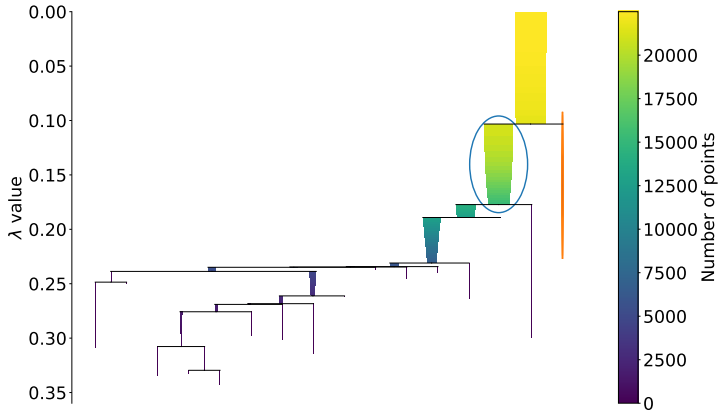
	label	cluster number		
		-1	0	1
PDU size	vol(frame_len)	0.03	-0.04	2.25
	avg(frame_len)	-0.23	0.02	0.83
	avg(tcp_len)	-0.56	0.13	-2.56
	avg(tcp_winsize)	0.5	-0.08	0.1
	avg(ip_len)	-0.18	0.01	1.07
	entropy(frame_len)	-0.79	0.18	-3.33
	entropy(tcp_len)	-0.83	0.13	-0.21
	entropy(tcp_winsize)	-0.8	0.09	1.34
	entropy(ip_len)	-0.76	0.18	-3.84
	var(frame_len)	-0.62	0.15	-3.17
	var(tcp_len)	-0.07	0.03	-1.29
	var(tcp_winsize)	0.09	-0.01	-0.01
	var(ip_len)	-0.59	0.14	-2.86
	stddev(frame_len)	-0.63	0.15	-3.52
	stddev(tcp_len)	-0.09	0.04	-1.28
	stddev(tcp_winsize)	0.13	-0.02	0.09
	stddev(ip_len)	-0.59	0.14	-3.17
	p50(frame_len)	0.11	-0.04	1.62
	p75(frame_len)	-0.29	0.05	-0.56
	p50(ip_len)	0.28	-0.08	2.32
	p75(ip_len)	-0.28	0.05	-0.42
	p50(tcp_len)	-0.58	0.13	-2.31
	p1(tcp_winsize)	0.37	-0.05	-0.36
	p2(tcp_winsize)	0.49	-0.07	-0.1
	p5(tcp_winsize)	0.71	-0.1	-0.1
	p10(tcp_winsize)	0.81	-0.12	-0.09
	p15(tcp_winsize)	0.81	-0.12	-0.02
	p20(tcp_winsize)	0.74	-0.11	0.03
	p25(tcp_winsize)	0.34	-0.05	0.06
	p50(tcp_winsize)	0.57	-0.09	0.14
	p75(tcp_winsize)	0.48	-0.07	-0.11
	p97(tcp_winsize)	0.55	-0.08	-0.02
	p99(tcp_winsize)	0.37	-0.05	-0.05
	entropy(tcp_flags)	0.52	-0.11	1.83
	entropy(icmp_type)	0.64	-0.1	0.07
Protocols frac.	tcp_flag_ack	-0.61	0.12	-1.72
	tcp_flag_push	0.46	-0.08	0.46
	tcp_flag_syn	0.62	-0.12	1.79
	tcp	-0.55	0.13	-2.8
	udp	0.63	-0.19	5.38
	gre	0.04	0.01	-0.82
	icmp	0.04	0.03	-1.89
	icmp_echo_request	-0.69	0.1	0.08
	icmp_echo_reply	0.66	-0.09	-0.08
	ipv4	0.4	-0.08	1.12
	ipv6	-0.4	0.08	-1.12
TCP port frac.	well_known(dst)	0.37	-0.09	1.85
	well_known(src)	-0.56	0.11	-1.42
	reg_ports(dst)	0.75	-0.11	-0.14
	reg_ports(src)	0.67	-0.11	0.93
	dyn_ports(dst)	-0.8	0.14	-1.11
	dyn_ports(src)	0.18	-0.05	1.15
	doc_retrieval(dst)	0.19	-0.05	1.23
	doc_retrieval(src)	-0.54	0.1	-1.12
	mail_tcp(dst)	0.04	-0.02	0.74
	mail_tcp(src)	0.32	-0.06	0.79
	remote_acc(dst)	0.57	-0.12	2.18
	remote_acc(src)	0.15	-0.01	-0.64
UDP port frac.	well_known(dst)	-0.13	0.07	-3.14
	well_known(src)	-0.1	0.04	-1.66
	reg_ports(dst)	0.28	-0.11	4.19
	reg_ports(src)	-0.08	0.04	-1.49
	dyn_ports(dst)	-0.31	0.09	-2.32
	dyn_ports(src)	0.23	-0.11	4.07
	networking(dst)	-0.12	0.05	-1.94
	networking(src)	-0.22	0.06	-1.33
IP oct. entropy	1st(dst)	-0.66	0.17	-4.03
	2nd(dst)	-0.77	0.19	-4.26
	3rd(dst)	-0.75	0.19	-4.78
	4th(dst)	-0.74	0.19	-4.79
	1st(src)	-0.99	0.23	-4.71
	2nd(src)	-1	0.23	-4.59
	3rd(src)	-0.98	0.22	-4.54
	4th(src)	-1.03	0.23	-4.22

(a) *Features table.*

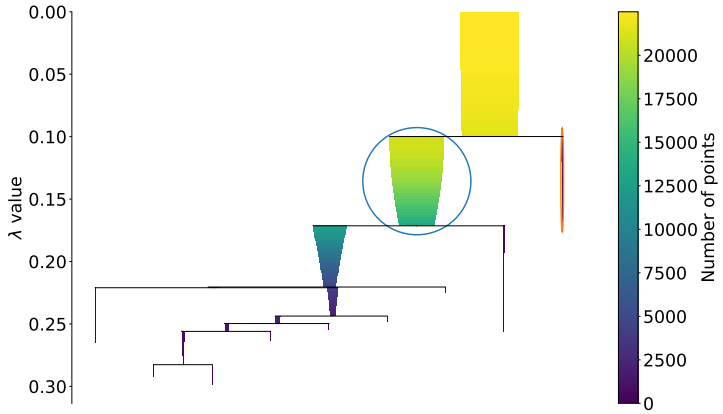
major category	category	cluster number		
		-1	0	1
dos		1.35	1	1.44
net_scan_tcp		1.06	1	1.24
net_scan_udp		1.41	1	2.18
netw_scan_icmp		0.68	1	0
port_scan		1.43	1	0
ipv6_tunneling		1.21	1	1.81
alpha_flow		1.13	1	0.95
multipoint		1.15	1	1.22
http		1.07	1	0.89
other		1.54	1	0.72
unknown		0.96	1	0
dos	DDoSIC	2.31	1	0
	sptpDoSSYN	0.23	1	0
	ptpDoSSYN	0.67	1	0
	sptpDoSIC	2.95	1	0
	DDoSSYN	2.02	1	3.59
net_scan_tcp	ntscACKt	0.14	1	0
	dntscSYN	0.28	1	0
	ntscSYN	0.79	1	0
	ntscSYNt139445	0.27	1	0
	sntscSYNt	1.15	1	4.17
	ntscSYNt	1.14	1	1.2
	ntscACK	0.95	1	0.27
	ntscTCPRSTACKrp	1.77	1	5.85
net_scan_udp	ntscUDPICdurp	1.31	1	8.67
	ptpposcaUDP	1.43	1	0
	ntscUDPUDPICrp	0.65	1	2.14
	ntscUDPUDPrp	1.56	1	2.84
	ntscUDP	1.38	1	1.46
net_scan_icmp	ntscICec	0.68	1	0
ipv6_tunneling	ipv46tun	1.31	1	0
	ipv4gretun	1.21	1	1.89
alpha_flow	malphfl	1.3	1	0
	heavy_hitter	1.31	1	2.67
	point_to_point	1.3	1	0
	salphfl	1.24	1	0.94
	alphfl	1.14	1	1.16
	alphflHTTP	1.04	1	0.73
http	ptmpHTTP	1.23	1	1.07
	ptmplaHTTP	1.12	1	1.15
	mptpHTTP	1.04	1	0.85
	mptplaHTTP	1.25	1	1.27
	alphflHTTP	1.04	1	0.73
multipoint	ptmpla	1.07	1	0.75
	mptpla	0.85	1	0.8
	mptmp	1.18	1	1.31
	mptp	1.33	1	1.76
	ptmp	1.1	1	1.36
	ptmpHTTP	1.23	1	1.07
	ptmplaHTTP	1.12	1	1.15
	mptpHTTP	1.04	1	0.85
	mptplaHTTP	1.25	1	1.27
other	icmp_error	1.46	1	0.89
	ttl_error	1.87	1	0
unknown	empty	0.96	1	0

(b) *Anomalies table.*

Table 4.3: *MAWI DBSCAN interpretations tables.*
Cluster −1 denotes outliers, green cells are general behaviour.



(a) *MinPts=107*



(b) *MinPts=212*

Figure 4.7: *MAWI HDBSCAN* condensed trees (MinCl=10).*

HDBSCAN*

We focus the HDBSCAN* clustering results analysis on Condensed trees as a Single Linkage tree would show 22500 branches, see Fig. 4.7. The size of the cluster is displayed by the size of the filled area between branching in the Condensed tree. We can see the algorithm produces comparable hierarchical structure for both MinPts values, but extracts only two clusters marked by a blue and orange circles. Noise is not included in the Condensed trees. Cluster marked by blue circle represents the most samples of the dataset and we denote it as a General behaviour in interpretation. We see a low number of extracted clusters as both strength and weakness of the algorithm. Algorithm parameters do not have a significant impact on the clustering results, but at the same time its hard to do fine-tuning to extract the variety of intrinsic hierarchical clustering structures. The V-measure of the obtained result in regard to major anomaly clustering is 0.027.

We again summarize the identified cluster characteristics for $MinPts = 212$ in comparison to the largest cluster, i.e. General behaviour, in Tab. 4.4. For the distributions of features and anomalies in each cluster we refer the interested reader to Appendix Fig. E.7-12.

	label	cluster number		
		-1	0	1
PDU size	vol(frame_len)	0.37	-0.08	2.24
	avg(frame_len)	0.08	-0.03	0.83
	avg(tcp_len)	-0.52	0.09	-2.53
	avg(tcp_winsize)	0.84	-0.05	0.09
	avg(ip_len)	0.13	-0.03	1.07
	entropy(frame_len)	-0.68	0.12	-3.24
	entropy(tcp_len)	-1.03	0.06	-0.29
	entropy(tcp_winsize)	-1.02	0.02	1.28
	entropy(ip_len)	-0.65	0.13	-3.75
	var(frame_len)	-0.59	0.11	-3.11
	var(tcp_len)	-0.41	0.05	-1.27
	var(tcp_winsize)	0.25	-0.01	-0.01
	var(ip_len)	-0.54	0.1	-2.81
	stddev(frame_len)	-0.61	0.12	-3.43
	stddev(tcp_len)	-0.43	0.06	-1.26
	stddev(tcp_winsize)	0.34	-0.02	0.1
	stddev(ip_len)	-0.55	0.11	-3.09
	p50(frame_len)	0.34	-0.06	1.64
	p75(frame_len)	-0.12	0.02	-0.55
	p50(ip_len)	0.53	-0.09	2.34
	p75(ip_len)	-0.11	0.02	-0.41
	p50(tcp_len)	-0.54	0.09	-2.29
	p1(tcp_winsize)	0.53	-0.02	-0.26
	p2(tcp_winsize)	0.59	-0.03	-0.02
	p5(tcp_winsize)	0.91	-0.05	-0.1
	p10(tcp_winsize)	1.24	-0.06	-0.09
	p15(tcp_winsize)	1.41	-0.07	-0.02
	p20(tcp_winsize)	1.33	-0.07	0.03
	p25(tcp_winsize)	0.63	-0.04	0.06
	p50(tcp_winsize)	0.89	-0.05	0.13
	p75(tcp_winsize)	0.56	-0.03	-0.09
	p97(tcp_winsize)	0.88	-0.05	-0.03
	p99(tcp_winsize)	0.63	-0.03	-0.01
	entropy(tcp_flags)	0.38	-0.06	1.75
	entropy(icmp_type)	0.59	-0.04	0.23
Protocols frac.	tcp_flag_ack	-0.97	0.09	-1.61
	tcp_flag_push	0.33	-0.03	0.33
	tcp_flag_syn	0.95	-0.09	1.67
	tcp	-0.41	0.09	-2.77
	udp	0.98	-0.19	5.31
	gre	-0.23	0.03	-0.82
	icmp	-0.46	0.07	-1.86
	icmp_echo_request	-0.49	0.03	-0.1
	icmp_echo_reply	0.38	-0.02	0.1
	ipv4	0.31	-0.04	1.09
	ipv6	-0.31	0.04	-1.09
TCP port frac.	well_known(dst)	0.18	-0.06	1.83
	well_known(src)	-0.59	0.07	-1.38
	reg_ports(dst)	0.81	-0.04	-0.11
	reg_ports(src)	0.78	-0.06	0.81
	dyn_ports(dst)	-0.73	0.07	-1.13
	dyn_ports(src)	0.13	-0.04	1.2
	doc_retrieval(dst)	-0.05	-0.03	1.27
	doc_retrieval(src)	-0.55	0.06	-1.11
	mail_tcp(dst)	0.12	-0.03	0.76
	mail_tcp(src)	0.67	-0.06	0.78
	remote_acc(dst)	0.65	-0.09	2.05
	remote_acc(src)	-0.24	0.03	-0.66
UDP port frac.	well_known(dst)	0.12	0.07	-3.14
	well_known(src)	0.05	0.04	-1.65
	reg_ports(dst)	0.19	-0.12	4.19
	reg_ports(src)	-0.26	0.03	-0.46
	dyn_ports(dst)	-0.63	0.09	-2.32
	dyn_ports(src)	0.27	-0.08	2.77
	networking(dst)	-0.09	0.05	-1.94
	networking(src)	-0.06	0.04	-1.33
IP oct. entropy	1st(dst)	-1.01	0.15	-3.97
	2nd(dst)	-1.14	0.17	-4.19
	3rd(dst)	-0.96	0.17	-4.71
	4th(dst)	-0.95	0.17	-4.72
	1st(src)	-1.17	0.18	-4.62
	2nd(src)	-1.18	0.18	-4.5
	3rd(src)	-1.17	0.18	-4.45
	4th(src)	-1.19	0.17	-4.14

(a) *MAWI HDBSCAN* features table.*

major category	category	cluster number		
		-1	0	1
dos		1.27	1	1.47
net_scan_tcp		0.99	1	1.23
net_scan_udp		1.32	1	2.16
netw_scan_icmp		0.82	1	0
port_scan		0.7	1	0
ipv6_tunneling		1.11	1	1.72
alpha_flow		1	1	0.97
multipoint		1.06	1	1.23
http		0.99	1	0.91
other		1.4	1	0.79
unknown		0.83	1	0
dos	DDoSIC	2.78	1	0.86
	sptpDoSSYN	0.19	1	0
	ptpDoSSYN	0.49	1	0
	sptpDoSIC	0.57	1	0
	DDoSSYN	1.98	1	3.39
net_scan_tcp	ntscACKt	0.19	1	0
	dntscSYN	0.45	1	0
	ntscSYN	1.42	1	0
	ntscSYNt139445	0.4	1	0
	sntscSYNt	1.47	1	4.33
	ntscSYNt	1.05	1	1.17
	ntscACK	0.77	1	0.27
	ntscTCPRSTACKrp	1.8	1	5.55
net_scan_udp	ntscUDPICdurp	1.59	1	8.25
	ptpposcaUDP	0.7	1	0
	ntscUDPUDPICrp	0.71	1	2.04
	ntscUDPUDPrp	1.43	1	2.83
	ntscUDP	1.31	1	1.49
net_scan_icmp	ntscICec	0.82	1	0
ipv6_tunneling	ipv46tun	1.43	1	0
	ipv4gretun	1.09	1	1.8
alpha_flow	malphfl	1.42	1	0
	heavy_hitter	1.64	1	3.16
	point_to_point	1.26	1	0
	salphfl	0.99	1	0.94
	alphfl	1.02	1	1.17
	alphflHTTP	0.9	1	0.74
http	ptmpHTTP	1.09	1	1.05
	ptmplaHTTP	1.01	1	1.1
	mptpHTTP	0.99	1	0.9
	mptplaHTTP	1.15	1	1.3
	alphflHTTP	0.9	1	0.74
multipoint	ptmpla	0.94	1	0.77
	mptpla	0.88	1	0.91
	mptmp	1.08	1	1.27
	mptp	1.23	1	1.78
	ptmp	0.94	1	1.37
	ptmpHTTP	1.09	1	1.05
	ptmplaHTTP	1.01	1	1.1
	mptpHTTP	0.99	1	0.9
	mptplaHTTP	1.15	1	1.3
other	icmp_error	1.48	1	0.99
	ttl_error	1.11	1	0
unknown	empty	0.83	1	0

(b) *MAWI HDBSCAN* anomalies table.*

Table 4.4: *MAWI HDBSCAN* interpretations tables.*
Cluster −1 denotes outliers, green cells are general behaviour.

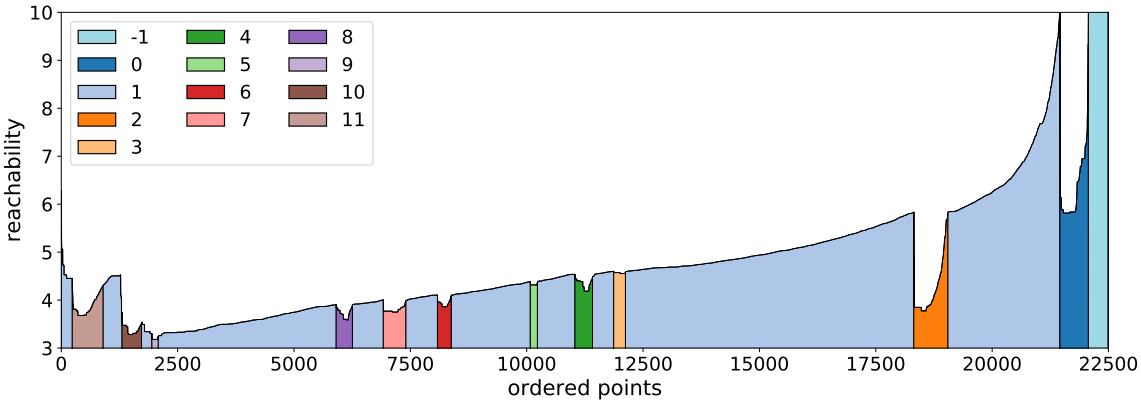


Figure 4.8: *MAWI OPTICS reachability plot* ($MinPts = 212, \xi = 0.002$).

OPTICS

Fig. 4.8 shows the reachability plot with extracted clusters by the most fitting parameter ξ that defines *steepness* denoting start/end of the cluster in the reachability plot. The plot is separated into valleys 0 and 1 and outliers not forming a valley marked as -1 . Each valley represents a set of points in close vicinity that form a cluster. Valleys inside valley 1 represent the child clusters to cluster 1. Cluster 1 represents the most samples of the dataset and we denote it as General behaviour. We see its child clusters as specific traffic patterns or "flavors" of normal traffic, whereas cluster 1 denotes suspicious or abnormal traffic. Deeper valleys represent clusters with significantly different subsets of feature values compared to the parent cluster but similar to cluster members, e.g cluster 2. We may think of them as high density sets of points. Shallow valleys represent small deviations to general behaviour, e.g. cluster 3. Reachability value shows how far apart the points are, i.e. cluster 1 points on the right side of the valley have a higher distribution of feature values compared to points on the left. In other words, the density of the cluster decreases as the reachability increases. The V-measure of the obtained result in regard to major anomaly clustering is 0.261.

As with DBSCAN and HDBSCAN* we summarize the identified cluster characteristics for $MinPts = 212, \xi = 0.002$ in comparison to the largest cluster, i.e. General behaviour, in Tab. 4.5 and Tab. 4.6. For distributions of features and anomalies in each cluster, we refer the interested reader to Appendix Fig. E.13-38.

	label	cluster number												
		-1	0	1	2	3	4	5	6	7	8	9	10	11
PDU size	vol(frame_len)	0.42	2.22		-0.92	-1.27	-0.47	0.04	0.12	0.3	0.36	0.5	-0.1	-1.07
	avg(frame_len)	0.15	0.81	0.04	-0.97	-1.53	-0.03	0.04	0.12	0.53	0.48	0.76	0.08	-1.02
	avg(tcp_len)	-0.84	-2.51	0.11	-0.34	-1.05	0.62	-0.38	0.49	0.36	0.06	0.65	0.33	-0.44
	avg(tcp_winsize)	1.34	0.18	-0.03	0.27	0.18	-0.09	0.07	-0.11	-0.13	-0.12	-0.36	-0.3	-0.14
	avg(ip_len)	0.16	1.05	0.02	-0.88	-1.42	-0.03	0.03	0.19	0.38	0.43	0.74	0.21	-1.06
	entropy(frame_len)	-0.72	-3.18	0.03	0.97	-1.1	0.47	0.89	0.27	1.18	0.81	0.59	0.13	-0.02
	entropy(tcp_len)	-1.31	-0.35	-0.01	0.18	-1.0	-0.21	0.14	0.99	0.77	0.54	0.67	0.28	0.01
	entropy(tcp_winsize)	-1.22	1.21	-0.03	-0.43	-0.62	-0.86	0.66	0.64	0.57	1.15	0.8	0.58	-0.22
	entropy(ip_len)	-0.7	-3.67	0.04	1.19	-0.84	0.74	0.7	0.27	1.08	0.54	0.56	-0.01	0.02
	var(frame_len)	-0.74	-3.06	0.14	-0.82	-1.28	0.29	0.42	0.51	0.49	0.67	0.78	0.51	-0.7
	var(tcp_len)	-0.72	-1.26	-0.01	0.75	1.01	-0.25	0.63	-0.21	-0.19	0.34	-0.35	0.15	0.66
	var(tcp_winsize)	0.67	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
	var(ip_len)	-0.71	-2.76	0.13	-0.77	-1.16	0.23	0.37	0.52	0.52	0.66	0.8	0.57	-0.79
	stddev(frame_len)	-0.77	-3.37	0.14	-0.73	-1.19	0.31	0.43	0.5	0.49	0.64	0.74	0.51	-0.62
	stddev(tcp_len)	-0.77	-1.26	-0.01	0.73	0.98	-0.23	0.62	-0.19	-0.17	0.35	-0.33	0.16	0.65
	stddev(tcp_winsize)	0.88	0.1	-0.02	-0.05	0.05	-0.11	0.08	-0.01	0.1	0.01	-0.06	-0.06	-0.09
	stddev(ip_len)	-0.74	-3.04	0.13	-0.68	-1.08	0.26	0.38	0.51	0.52	0.64	0.75	0.56	-0.7
	p50(frame_len)	0.42	1.64	0.03	-0.49	-0.5	-0.48	-0.47	-0.48	-0.2	-0.35	-0.25	-0.48	-0.5
	p75(frame_len)	-0.12	-0.55	0.04	-0.77	-2.07	0.51	0.55	0.55	0.57	0.6	0.61	0.59	-0.75
	p50(ip_len)	0.58	2.34	-0.02	-0.36	-0.41	-0.35	-0.35	-0.35	-0.29	-0.31	-0.28	-0.35	-0.4
	p75(ip_len)	-0.16	-0.41	0.04	-0.78	-1.85	0.54	0.6	0.6	0.63	0.65	0.66	0.64	-1.09
	p50(tcp_len)	-0.79	-2.26	0.07	-0.11	-1.39	0.54	0.07	0.52	0.48	0.4	0.54	0.52	-0.16
	p1(tcp_winsize)	0.61	-0.21	0.03	-0.03	0.22	0.6	-0.02	0.24	-0.33	0.34	0.25	-0.34	-1.41
	p2(tcp_winsize)	0.8	0.03	0.06	-0.28	0.81	0.25	-0.04	-0.42	-0.12	-0.37	-0.21	-0.72	-1.28
	p5(tcp_winsize)	1.31	-0.08	0.01	-0.19	1.19	0.27	-0.29	-0.33	-0.27	-0.34	-0.25	-0.3	-0.51
	p10(tcp_winsize)	1.94	-0.04	-0.01	0.06	0.32	0.19	-0.31	-0.34	-0.35	-0.33	-0.35	-0.34	-0.35
	p15(tcp_winsize)	2.43	0.03	-0.02	0.02	0.15	0.01	-0.22	-0.31	-0.31	-0.3	-0.33	-0.3	-0.32
	p20(tcp_winsize)	2.09	0.17	-0.02	-0.03	0.05	-0.04	-0.18	-0.27	-0.25	-0.23	-0.26	-0.23	-0.27
	p25(tcp_winsize)	0.74	0.3	-0.01	-0.02	0.03	-0.02	-0.08	-0.12	-0.1	-0.1	-0.11	-0.1	-0.12
	p50(tcp_winsize)	1.1	0.31	-0.02	0.09	-0.03	0.35	-0.02	-0.23	-0.19	-0.16	-0.28	-0.2	-0.17
	p75(tcp_winsize)	0.92	0.01	-0.02	0.91	-0.33	0.1	0.18	-0.25	-0.43	-0.27	-0.53	-0.41	-0.1
	p97(tcp_winsize)	0.83	0.06	0.0	-0.07	-0.14	-0.19	0.04	0.0	-0.08	-0.05	-0.15	-0.07	-0.09
	p99(tcp_winsize)	0.89	0.06	-0.02	0.39	0.1	-0.19	-0.11	-0.07	-0.18	-0.08	-0.21	-0.17	-0.02
	entropy(tcp_flags)	0.56	1.66	-0.11	0.79	0.84	-0.36	-0.32	-0.35	-0.12	-0.73	-0.79	-0.1	0.97
	entropy(icmp_type)	0.92	0.29	0.01	-0.27	-0.69	-0.51	-0.28	2.08	-0.36	-0.38	-0.46	-0.26	-0.39
Protocols frac.	tcp_flag_ack	-1.6	-1.56	0.1	-0.38	-0.65	-0.17	0.47	0.43	0.43	0.68	0.65	0.34	-0.74
	tcp_flag_push	0.32	0.26	-0.02	0.66	-0.12	-0.69	0.26	-0.07	0.08	-0.25	-0.53	0.02	0.01
	tcp_flag_syn	1.57	1.61	-0.11	0.46	0.81	0.13	-0.5	-0.44	-0.42	-0.7	-0.66	-0.31	0.69
	tcp	-0.44	-2.73	0.14	-0.96	-1.06	-0.14	0.69	0.25	0.57	0.82	0.86	0.3	-0.92
	udp	1.29	5.24	-0.22	0.71	-0.09	0.13	-0.15	-0.35	-0.13	-0.34	-0.31	-0.38	0.11
	gre	-0.24	-0.81	0.1	-0.07	-0.24	-0.61	-0.82	-0.53	-0.42	0.01	-0.49	-0.38	-0.21
	icmp	-0.74	-1.83	0.03	0.49	1.46	0.11	-0.61	0.11	-0.54	-0.69	-0.71	0.05	1.07
	icmp_echo_request	-0.55	-0.21	-0.02	0.29	0.58	0.44	0.42	-2.23	0.41	0.4	0.44	0.28	0.4
	icmp_echo_reply	0.26	0.22	0.03	-0.29	-0.58	-0.43	-0.44	2.26	-0.42	-0.4	-0.44	-0.26	-0.4
	ipv4	0.18	1.08	-0.07	0.69	0.88	0.34	-0.23	0.26	-0.83	-0.55	-0.27	0.38	0.13
	ipv6	-0.18	-1.08	0.07	-0.69	-0.88	-0.34	0.23	-0.26	0.83	0.55	0.27	-0.38	-0.13
TCP port frac.	well_known(dst)	0.42	1.79	-0.09	0.01	1.73	-0.6	0.34	-0.5	0.38	0.07	-0.55	-0.27	0.2
	well_known(src)	-0.87	-1.38	0.08	-0.43	-1.5	0.57	-0.53	0.62	0.05	0.16	0.73	0.25	-0.25
	reg_ports(dst)	0.94	-0.1	0.01	0.34	-0.25	1.07	-0.6	-0.04	-0.71	-0.59	-0.68	-0.38	-0.32
	reg_ports(src)	1.02	0.76	-0.06	0.76	0.54	-0.06	-0.3	-0.42	-0.63	-0.11	-0.81	-0.29	0.46
	dyn_ports(dst)	-0.98	-1.11	0.05	-0.26	-0.96	-0.4	0.22	0.36	0.28	0.39	0.87	0.46	0.11
	dyn_ports(src)	0.29	1.25	-0.06	-0.09	1.61	-0.73	1.02	-0.5	0.49	-0.12	-0.28	-0.09	-0.07
	doc_retrieval(dst)	-0.01	1.26	-0.04	-0.27	1.14	-0.94	0.73	-0.28	0.8	0.06	-0.26	-0.09	-0.17
	doc_retrieval(src)	-0.81	-1.12	0.07	-0.45	-2.03	0.71	-0.42	0.7	0.21	0.2	0.77	0.28	-0.29
	mail_tcp(dst)	0.37	0.73	-0.08	-0.36	-0.05	-0.25	-0.16	-0.2	-0.14	3.64	-0.22	-0.2	0.24
	mail_tcp(src)	0.79	0.77	-0.03	0	0.33	-0.44	-0.3	-0.04	0.04	-0.21	-0.05	-0.15	0.02
	remote_acc(dst)	1.13	1.97	-0.12	0.62	0.11	0.85	-0.71	-0.51	-0.49	-0.85	-0.65	-0.35	0.84
	remote_acc(src)	-0.31	-0.66	0.07	0.2	-0.03	-0.75	-0.55	-0.69	-0.53	-0.51	-0.58	-0.55	0.68
UDP port frac.	well_known(dst)	0.85	-3.13	-0.04	1.89	0.21	0.96	-2.13	0.33	1.1	-0.89	0.15	-0.09	0.7
	well_known(src)	0.17	-1.65	-0.13	2.84	-0.01	1.53	-0.75	-0.27	1.56	-0.4	-0.67	-0.41	0.63
	reg_ports(dst)	-0.56	4.18	-0.04	-1.18	-0.18	-0.86	1.05	-0.4	-0.64	0.55	-0.33	-0.06	-0.3
	reg_ports(src)	-0.37	-0.2	0.14	-2.36	-0.04	-1.42	1.21	0.43	-1.18	0.34	0.8	0.67	-0.23
	dyn_ports(dst)	-0.53	-2.31	0.16	-1.34	-0.04	-0.14	2.07	0.15	-0.87	0.63	0.36	0.3	-0.77
	dyn_ports(src)	0.24	2.42	-0.01	-0.76	0.06	-0.22	-0.54	-0.19	-0.56	0.1	-0.13	-0.31	-0.54
	networking(dst)	0.04	-1.94	-0.07	2.18	0.65	0.5	-0.81	-0.71	1.63	-0.04	-0.29	-0.2	0.3
	networking(src)	-0.05	-1.33	-0.16	3.32	0.2	1.56	-0.16	-0.45	1.77	-0.04	-0.43	-0.52	0.29
IP oct. entropy	1st(dst)	-1.33	-3.92	0.07	0.68	1.29	0.24	0.47	-0.51	0.21	0.21	-0.3	0.19	1.05
	2nd(dst)	-1.48	-4.14	0.09	0.51	1.16	0.29	0.5	-0.36	0.32	0.23	-0.22	0.18	0.99
	3rd(dst)	-1.1	-4.65	0.09	0.86	0.92	0.29	0.45	-0.26	0.24	0.19	0.01	0.06	0.81
	4th(dst)	-1.08	-4.66	0.09	0.91	0.93	0.25	0.37	-0.28	0.26	0.23	0.0	0.01	0.76
	1st(src)	-1.36	-4.54	0.15	-0.15	-0.4	0.15	0.45	0.74	0.36	0.68	0.72	0.52	-0.11
	2nd(src)	-1.38	-4.42	0.15	-0.2	-0.45	0.13	0.49	0.75	0.46	0.63	0.73	0.47	-0.13
	3rd(src)	-1.42	-4.37	0.13	0.09	-0.62	-0.02	0.77	0.51	0.69	0.97	0.79	0.39	-0.22
	4th(src)	-1.42	-4.07	0.13	0	-0.57	0.03	0.71	0.42	0.8	1	0.86	0.32	-0.29

Table 4.5: *MAWI OPTICS features interpretations tables. Cluster −1 denotes outliers, green cells are general behaviour.*

major category	category	cluster number												
		-1	0	1	2	3	4	5	6	7	8	9	10	11
dos		1.13	1.35	1	0	0.01	3.26	0	0.3	1.26	0	0.96	0	0.18
net_scan_tcp		0.93	1.17	1	0.22	0.37	1.13	0.22	0.89	0.66	0.87	0.69	1.68	0.83
net_scan_udp		1.22	1.96	1	0.19	0.96	1.26	0.2	0.63	0.19	0.43	0.45	0.73	0.33
netw_scan_icmp		0.41	0	1	0	0	7.7	0	0.37	0	0	0	0	0
port_scan		0.65	0	1	0	0	0	0	0	0	0	0	11.29	0.03
ipv6_tunneling		1.03	1.6	1	0	1.04	1.04	0	0.9	1.04	1.03	0.38	1.04	0.22
alpha_flow		0.92	0.93	1	0.12	1.42	1.46	0.19	0.9	0.84	1.14	0.59	1.22	0.18
multipoint		0.98	1.16	1	0.31	0.89	0.85	0.4	0.88	1.15	0.8	0.88	0.88	0.45
http		0.94	0.88	1	0.43	1.14	1.01	0.61	0.74	0.92	0.78	0.69	1.2	0.54
other		1.09	0.73	1	0	1.05	3.82	0	0.66	0	0.24	0.54	0.56	0.2
unknown		0.71	0	1	0	1.29	1.8	0	0.14	1.71	4.84	0.18	1.37	0.24
dos	DDoSIC	3.66	1.1	1	0	0	29.62	0	0.3	0	0	0	0	0
	sptpDoSSYN	0.05	0	1	0	0	0	0	0.94	0	0	0	0	0
	ptpDoSSYN	0.34	0	1	0	0	0	0	0.27	2.99	0.01	2.27	0.01	0.01
	sptpDoSIC	0.55	0	1	0	0	0	0	0	0	0	0	0	0
	DDoSSYN	1.77	2.99	1	0	0.01	3.66	0	0.24	0	0	0	0	0.4
net_scan_tcp	ntscACKt	0.12	0	1	0	0	0	0	16.92	0	0.07	5.2	0	0
	dntscSYN	0.59	0	1	0	0	0	0	0	42.72	0	0	0	0
	ntscSYN	2.55	0	1	17.64	0	0	0	0	0	0	1.18	0	0
	ntscSYNt139445	0.14	0	1	0	0	0	0	0	0	0	0	0	0
	sntscSYNt	1.62	4.19	1	0	0	0	0	0.54	1.08	2.03	0.02	4.07	0.08
	ntscSYNt	0.98	1.13	1	0.27	0.86	1.69	0.27	0.57	0.27	1.08	0.58	1.47	1.1
	ntscACK	0.63	0.26	1	0	0.04	1.02	0.25	1.16	0.76	0.51	0.94	1.52	0.62
	ntscTCPRSTACKrp	1.95	5.12	1	0	0.08	0	0	0.69	0	0	0.25	0.01	3.06
net_scan_udp	ntscUDPICdurp	1.37	6.93	1	0	0	0	0	0.9	0	0	0.85	0	0
	ptpposcaUDP	0.65	0	1	0	0	0	0	0	0	0	0	11.29	0.03
	ntscUDPUDPICrp	0.41	1.72	1	0	0	0	0	0.66	0	0.01	1.86	0	0.01
	ntscUDPUDPrp	1.31	2.58	1	0	0.9	1.6	0.01	0.48	0.45	0.73	0.1	0.8	0.49
	ntscUDP	1.25	1.34	1	0.4	1.19	1.2	0.4	0.78	0	0.24	0.59	0.4	0.26
net_scan_icmp	ntscICec	0.41	0	1	0	0	7.7	0	0.37	0	0	0	0	0
ipv6_tunneling	ipv46tun	1.51	0.03	1	0	0	0	0	0	0	0	0	0	1.25
	ipv4gretun	1	1.68	1	0	1.09	1.09	0	0.95	1.09	1.09	0.4	1.09	0.17
alpha_flow	malphfl	0.97	0.01	1	0	0.02	3.49	0	0.09	0	0	0	1.79	0.56
	heavy_hitter	1.98	3.06	1	0	3.46	0.12	0	0.12	0	5.63	0.08	0	0.28
	point_to_point	1.1	0	1	0	6	6.12	0	0	0	0.02	0.73	0	0.48
	salphfl	0.91	0.92	1	0	0.96	2.14	0	0.72	0	2.8	0.26	2.66	0.14
	alphfl	0.94	1.13	1	0	1.31	2.03	0	1.52	1.31	0.66	0.75	1.31	0.13
	alphflHTTP	0.8	0.68	1	0.28	1.38	0.56	0.45	0.58	0.84	0.84	0.62	0.84	0.18
http	ptmpHTTP	1.16	1.05	1	1.05	2.07	1.05	1.05	1	1.05	0	0.33	1.05	1.12
	ptmplaHTTP	1.01	1.07	1	0.64	1.91	0.64	1.27	1.19	0.64	1.27	0.36	1.23	0.24
	mptpHTTP	0.94	0.87	1	0.28	0.69	1.41	0.55	0.74	0.83	0.83	0.92	1.38	0.72
	mptplaHTTP	1.22	1.28	1	1.41	1.44	0	0	0.25	2.81	0	0	1.4	0.31
	alphflHTTP	0.8	0.68	1	0.28	1.38	0.56	0.45	0.58	0.84	0.84	0.62	0.84	0.18
multipoint	ptmpla	0.74	0.7	1	0	0	0.54	0	0.78	1.72	0.55	1.19	0.54	0.12
	mptpla	0.92	0.96	1	0	2.09	2.12	0	1.43	5.02	1.07	1.33	0	0.1
	mptmp	0.98	1.2	1	0.35	0.88	0.53	0.44	0.75	1.48	0.78	0.96	0.77	0.41
	mptp	1.06	1.61	1	0	0.5	0.94	0.16	1.14	0.46	1.18	0.95	0.65	0.15
	ptmp	0.94	1.27	1	0.52	1.53	0.52	0	1.2	0.01	0	0.47	0.52	0.98
	ptmpHTTP	1.16	1.05	1	1.05	2.07	1.05	1.05	1	1.05	0	0.33	1.05	1.12
	ptmplaHTTP	1.01	1.07	1	0.64	1.91	0.64	1.27	1.19	0.64	1.27	0.36	1.23	0.24
	mptpHTTP	0.94	0.87	1	0.28	0.69	1.41	0.55	0.74	0.83	0.83	0.92	1.38	0.72
	mptplaHTTP	1.22	1.28	1	1.41	1.44	0	0	0.25	2.81	0	0	1.4	0.31
other	icmp_error	1.11	0.93	1	0	0.01	4.85	0	0.84	0	0.31	0.68	0.71	0.11
	ttl_error	1.04	0.01	1	0	4.93	0	0	0	0	0	0	0	0.54
unknown	empty	0.71	0	1	0	1.29	1.8	0	0.14	1.71	4.84	0.18	1.37	0.24

Table 4.6: *MAWI OPTICS anomalies interpretations tables. Cluster −1 denotes outliers, green cells are general behaviour.*

cluster num.	size[%]	max.cont. [daily %]	interpretation (traffic description)
OPTICS			
-1	1.93	4.22	noise/outliers
0	2.72	1.89	high vol. UDP traffic to reg. ports between small num. of endpoints
1	78.24	86.22	general behaviour
2	3.24	11.33	small size and vol. UDP netw. traffic
3	1.12	0.78	small size and vol. TCP doc. retrieval traffic between small num. of src. and high num. of dst. endpoints
4	1.68	4.22	traffic to remote access TCP services + UDP netw. traffic
5	0.68	1.00	NA
6	1.31	4.56	ICMP echo reply traffic (DoS)
7	2.14	2.67	IPv6 UDP netw. traffic
8	1.55	1.89	large num. of endpoints to TCP mail serv.
9	0.60	1.00	large size/vol. TCP doc. retrieval traffic from large num. of endpoints
10	1.86	3.33	NA
11	2.93	3.89	small vol. and size traffic to large num. of endpoints, increased TCP SYN flags, ICMP echo request and remote access TCP traffic (admin traffic)
DBSCAN			
-1	12.62	5.33	noise/outliers
0	85.86	42.00	general behaviour
1	1.52	1.78	high vol. UDP traffic to reg. ports between small num. of endpoints
HDBSCAN			
-1	4.96	4.56	noise/outliers
0	92.69	64.56	general behaviour
1	2.35	1.89	high vol. UDP traffic to reg. ports between small num. of endpoints

Table 4.7: *MAWI clusters interpretations.*

4.2.4 Interpretation

Table 4.7 summarizes the interpretations of all algorithms results. We do an *open interpretation* as we describe the traffic in each cluster based on all available dataset characteristics. We focus on OPTICS results as the clustering process is easiest to calibrate, understand and interpret. We interpreted the clusters by analyzing the shift of the average feature values in clusters compared to general behaviour while taking into account their distributions. We also compared anomaly distributions to General behaviour. Interestingly, clusters marked with specific suspicious features, e.g. fraction of UDP packets, correlate with significantly higher resembling anomalies - i.e. *net scan UDP* (OPTICS cluster 0, Tab. 4.7). Ultimately, we focus on the interpretation by suspicious features, as the anomaly to feature correlation is ambiguous for some clusters, e.g. OPTICS cluster 7, Tab. 4.7 anomaly magnitudes are within *general behaviour* constraints. OPTICS clusters 5 and 7 did not resemble any traffic behaviour known to authors. General behaviour in MAWI traces is characterized by high volume of TCP traffic. We expected this behaviour. Interestingly, all algorithms successfully recognized it and separated TCP and UDP traffic. In the case of OPTICS, we see this in the reachability plot by dividing the dataset into TCP - cluster 1, and UDP - cluster 0, traffic. Results of DBSCAN and HDBSCAN* are comparable, but HDBSCAN is able to cluster a larger fraction of the dataset leaving less noise/outliers and does not need parameter adjustment and evaluation to achieve this result as DBSCAN.

4.2.5 Outliers analysis

In addition, we also analysed the ability of the algorithms to detect outliers and compared the detected sets by Jaccard index ("Intersection over Union"):

$$J(DBSCAN, HDBSCAN) = 0.392 \tag{4.1}$$

$$J(OPTICS, DBSCAN) = 0.153 \tag{4.2}$$

$$J(OPTICS, HDBSCAN) = 0.390 \tag{4.3}$$

The highest overlap is between the most related methods, DBSCAN and HDBSCAN*. OPTICS can be easily fine-tuned to detect clusters with lower density whereas HDBSCAN*, is by default, strict in conditions for cluster membership. Clusters detected by HDBSCAN* are well-defined, but the clustering results give us less insights than OPTICS.

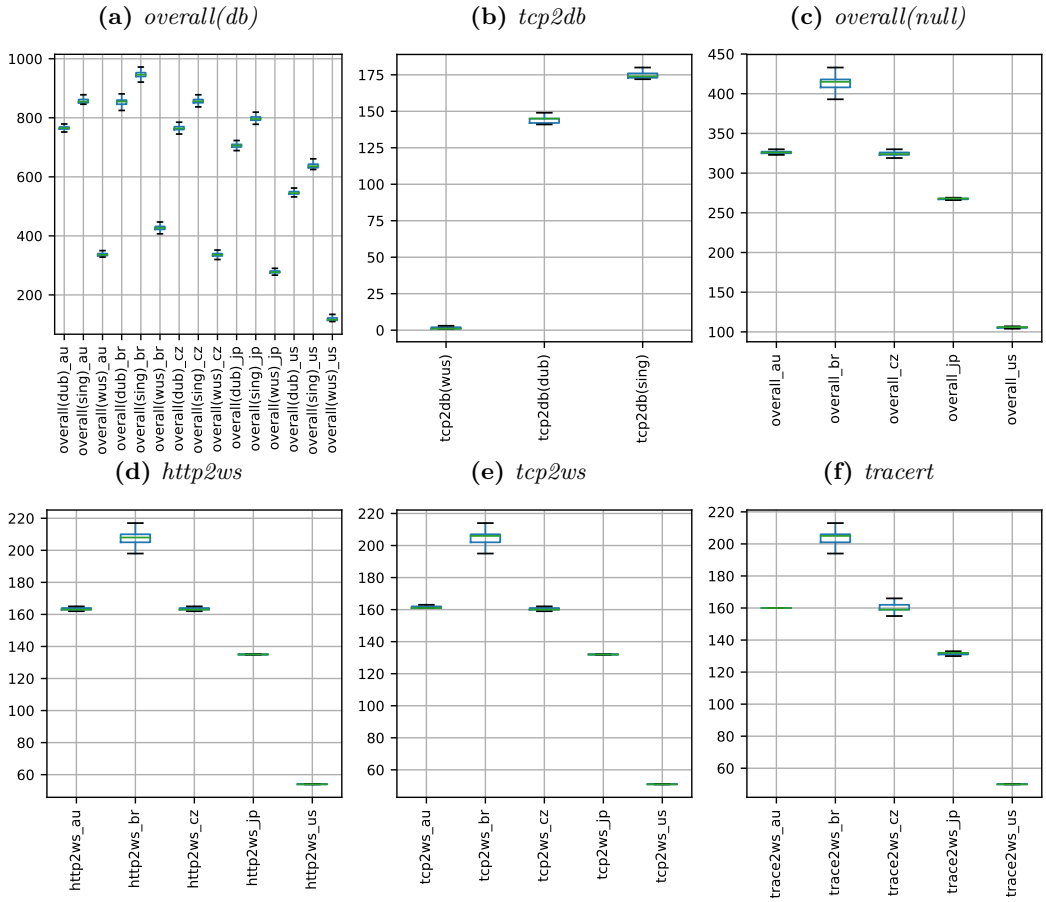


Figure 4.9: *CLAudit measurements distributions.*

4.3 Scenario II - CLAudit

4.3.1 Dataset

We use 4 weeks of the publicly available CLAudit dataset [79] as measurements for the study, ranging from January 10th 2016 to February 4th 2016. The dataset consists of 9360 points, of 38 dimensions. We consider a single Cloud instance located in California (WUS) for the Azure data-center, to have comparable results to our previous works presented in [3, 37]. We did not filter out features from the dataset as we use a Decision Tree to interpret the clusters. We focus on slight deviations in active measurements to detect and interpret suspicious network behaviour, see Fig. 4.9. Inputs to the clustering algorithm consist of all measurements coming from all Vantage Points to a given data center at time t, in the form of a multi-dimensional vector of dimension $q = 38$ (4 x non-db and 3 x overall(db), from 5 VPs; 3 x tcp2db) in our dataset. Vectors represent all measurement types at a given timestamp. Clustering of these input vectors results in timestamps-cluster ID pairs.

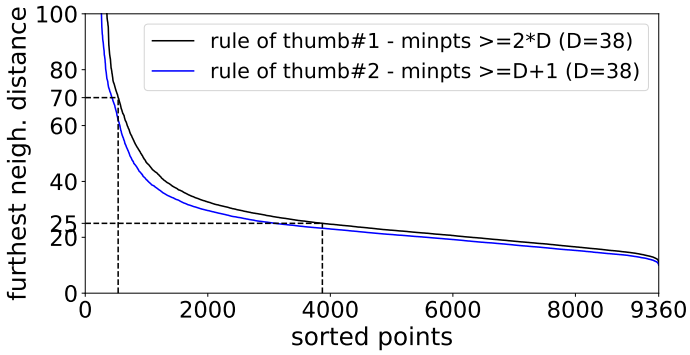


Figure 4.10: *CLAudit DBSCAN k-dist plots.*

4.3.2 Clustering

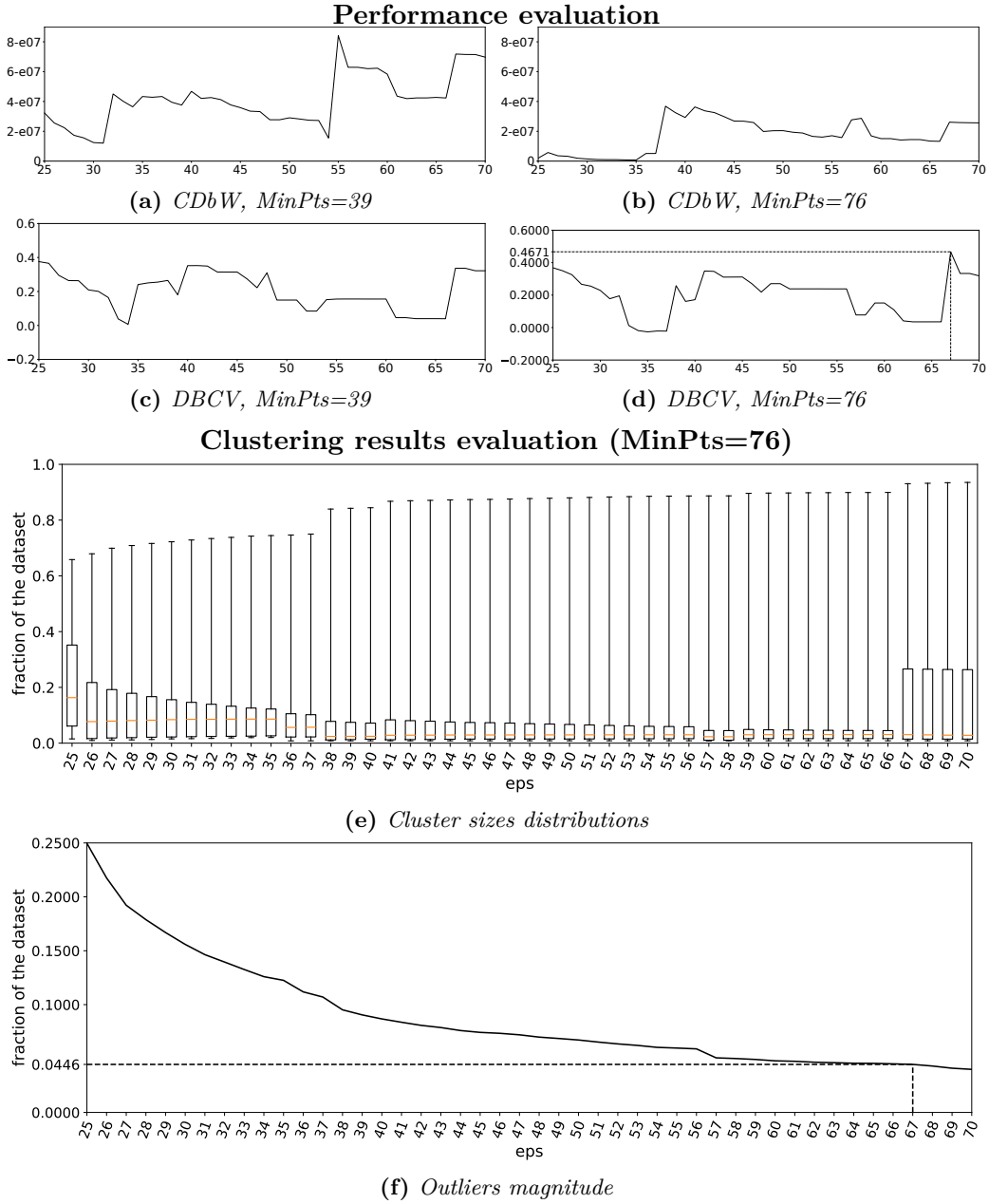
We repeat the same procedure as with MAWI dataset. In the CLAudit scenario, we do not possess anomaly labels and focus only on unsupervised performance evaluation metrics DBCV and CDbw.

DBSCAN

In ϵ selection process, we again focus on the lower knee values in k-dist plots with interval values $< 25; 70 >$ indicating 70th and 95th percentile, see Fig. 4.10. In Fig. 4.11 a)-d), we analyze the impact of ϵ on clustering results through DBCV and CDbw. In general, the values for both choices of $MinPts$ are comparable and we confirm that $MinPts$ does not have a significant impact on results, with higher $MinPts$ (rule of thumb # 1) slightly outperforming lower (rule of thumb # 2). We confirm that CDbw is unable to identify a parameter combination that would correlate with a change of clustering results characteristics. We achieved the best positive DBCV results at $Minpts = 76, \epsilon = 67$. The value correlates with the increased size of the detected clusters and low fraction of the dataset marked as outliers/noise. A high positive DBCV value and a small number of clusters indicate hidden hierarchical clustering structure. We proceed with dataset analysis by HDBSCAN* and OPTICS.

We summarize the identified cluster characteristics for $Minpts = 76, \epsilon = 67$ in comparison to the largest cluster, i.e. general behaviour, in Tab. 4.8. We mimic the same procedure as with MAWI dataset. We take the largest cluster as a baseline and compare other clusters to it by increased or decreased fraction of average latency measurements values. In this case, we empirically set one upper and one lower threshold to keep our procedure inline with our previous work on CLAudit measurements [3, 37]. We use green

cell color to denote General behaviour, Red to denote values above upper threshold set to 1.05 and blue for lower threshold set to 0.95 representing 5% increase/decrease of average measured latency in a cluster. For distributions of features and anomalies in each cluster, we refer the interested reader to Appendix Fig. E.39-42.



(a) *CDbW, MinPts=39*

(b) *CDbW, MinPts=76*

(c) *DBCV, MinPts=39*

(d) *DBCV, MinPts=76*

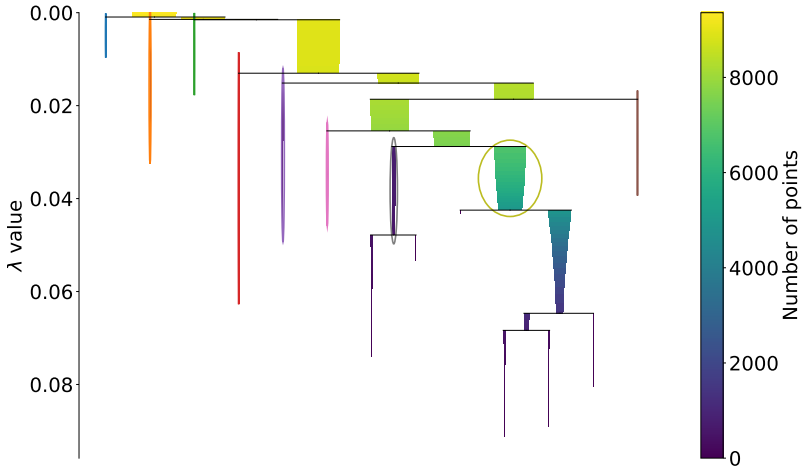
(e) *Cluster sizes distributions*

(f) *Outliers magnitude*

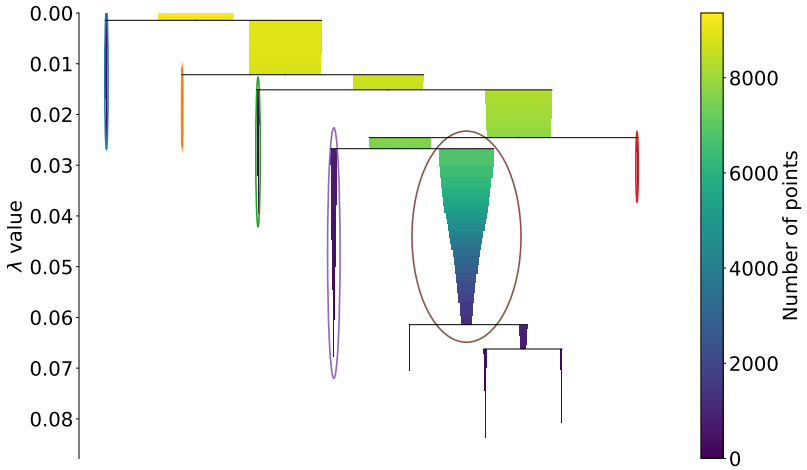
Figure 4.11: *CLAudit DBSCAN evaluation.*

label	cluster number			
	-1	**0**	**1**	**2**
overall(dub)_au	1.14	1	1	1.11
overall(sing)_au	1.17	1	1	1.11
overall(wus)_au	1.37	1	1.01	1.28
overall(dub)_br	1.33	1	1.01	1
overall(sing)_br	1.28	1	1.01	1.01
overall(wus)_br	1.69	1	1.01	1.02
overall(dub)_cz	1.11	1	1	0.99
overall(sing)_cz	1.1	1	1	1
overall(wus)_cz	1.27	1	1	1
overall(dub)_jp	1.22	1	1	0.99
overall(sing)_jp	1.43	1	1	1
overall(wus)_jp	1.47	1	1.01	0.99
overall(dub)_us	1.59	1	1	0.98
overall(sing)_us	1.52	1	3.5	0.99
overall(wus)_us	2.21	1	1.03	0.97
tcp2db(dub)	1.96	1	1	0.98
tcp2db(sing)	1.79	1	1	0.99
tcp2db(wus)	70.22	1	1.21	0.82
overall_au	1.31	1	1	1.29
overall_br	1.77	1	1.01	1.03
overall_cz	1.28	1	1	1
overall_jp	1.4	1	1	1
overall_us	5	1	1	0.99
http2ws_au	1.43	1	1	1.29
http2ws_br	2.55	1	1.01	1.03
http2ws_cz	1.57	1	1	1
http2ws_jp	1.54	1	1	1
http2ws_us	9.13	1	1	0.99
tcp2ws_au	1.43	1	1	1.29
tcp2ws_br	2.57	1	1.01	1.03
tcp2ws_cz	1.58	1	1	1
tcp2ws_jp	1.55	1	1	1
tcp2ws_us	9.59	1	1	0.99
trace2ws_au	1.44	1	1	1.29
trace2ws_br	2.29	1	1.01	1.03
trace2ws_cz	1.3	1	1.04	1.04
trace2ws_jp	1.37	1	1	1
trace2ws_us	2.69	1	0.99	1

Table 4.8: *CLAudit DBSCAN features interpretations table. Cluster −1 denotes outliers, green cells are general behaviour.*



(a) *MinPts=39*



(b) *MinPts=76*

Figure 4.12: *CLAudit HDBSCAN* condensed trees (MinCl=10)*

.

HDBSCAN*

Hierarchical clustering analysis of the dataset by HDBSCAN* confirms that DBSCAN uncovered only initial flat clustering. Condensed trees in Fig. 4.12 show the hidden structure of clusters. By default, the algorithm extracts only the clusters represented as leaves resulting in missing parent-child cluster relationships in interpretation. For interpretation, we proceed with clusters extracted by $MinPts = 76$ parameter setting, producing stable and uniformed hierarchical structure.

As with DBSCAN, we again summarize the identified cluster characteristics for $MinPts = 76$ in comparison to the largest cluster, i.e. general behaviour, in Tab. 4.9. For distributions of features and anomalies in each cluster, we refer the interested reader to Appendix Fig. E.43-49.

label	cluster number						
	-1	0	1	2	3	4	5
overall(dub)_au	1.06	1	1.11	1	1	1	1
overall(sing)_au	1.07	1	1.11	1	1	0.99	1
overall(wus)_au	1.15	1.01	1.28	1	1	0.99	1
overall(dub)_br	1.12	0.99	1	1	1	0.94	1
overall(sing)_br	1.1	1	1	1	1	0.94	1
overall(wus)_br	1.24	0.99	1.01	1	1	0.88	1
overall(dub)_cz	1.04	1	0.99	0.99	0.99	1	1
overall(sing)_cz	1.04	1	1	1	0.99	0.99	1
overall(wus)_cz	1.09	1	0.99	0.99	0.99	0.99	1
overall(dub)_jp	1.08	1	0.99	1	1	1	1
overall(sing)_jp	1.16	1	1	1	1	0.99	1
overall(wus)_jp	1.18	1.01	0.99	1	1	0.99	1
overall(dub)_us	1.22	1	0.98	1	1	1	1
overall(sing)_us	1.11	3.49	0.99	1	1	0.99	1
overall(wus)_us	1.49	1.03	0.97	1	0.99	0.98	1
tcp2db(dub)	1.35	1	0.98	0.99	1	1	1
tcp2db(sing)	1.29	1	0.99	1	0.99	0.99	1
tcp2db(wus)	25.14	1.11	0.78	0.9	0.72	0.63	1
overall_au	1.13	1	1.3	1	1	1	1
overall_br	1.27	0.99	1.02	1	1	0.88	1
overall_cz	1.1	1	1	0.99	0.99	1	1
overall_jp	1.15	1	1	1	1	1	1
overall_us	2.5	1.01	0.99	1	1	1	1
http2ws_au	1.17	1	1.29	1	1	1	1
http2ws_br	1.55	0.99	1.02	1	1	0.88	1
http2ws_cz	1.2	1	1	0.99	0.99	1	1
http2ws_jp	1.2	1	1	1	1	1	1
http2ws_us	4.01	1.01	1	1	1	1	1
tcp2ws_au	1.18	1	1.3	1	1	1	1
tcp2ws_br	1.56	0.99	1.02	1	1	0.88	1
tcp2ws_cz	1.21	1	1	0.99	0.99	1	1
tcp2ws_jp	1.2	1	1	1	1	1	1
tcp2ws_us	4.18	1.01	1	1	1	1	1
trace2ws_au	1.18	1.01	1.3	1.01	1	1	1
trace2ws_br	1.46	0.99	1.02	1.01	1	0.88	1
trace2ws_cz	1.05	0.97	1	0.13	0.56	1	1
trace2ws_jp	1.14	1	1	1	1	1	1
trace2ws_us	1.65	1.01	1	1	1	1	1

Table 4.9: *CLAudit HDBSCAN* features interpretations table. Cluster −1 denotes outliers, green cells are general behaviour.*

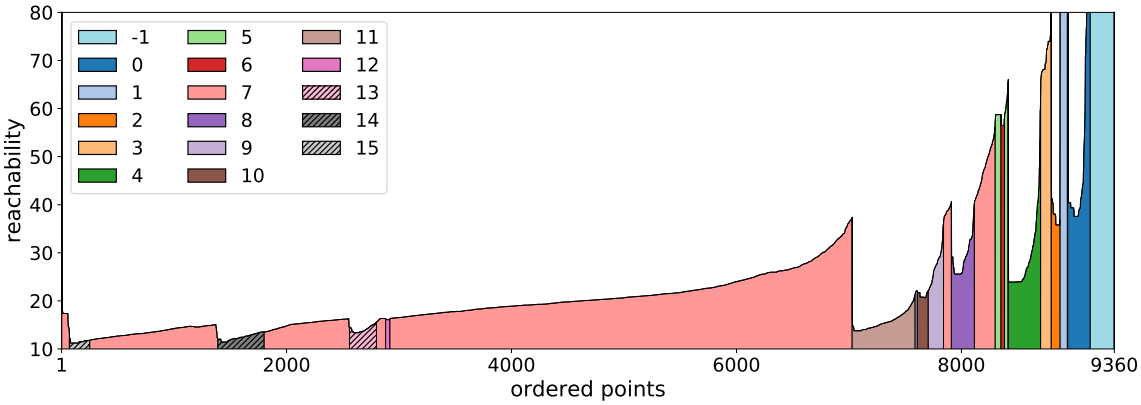


Figure 4.13: *CLAudit OPTICS reachability plot* ($MinPts = 76, \xi = 0.01$).

OPTICS

Fig. 4.13 shows OPTICS reachability plot with extracted clusters by the most fitting ξ, all residing in one valley (general behaviour), denoted as cluster 7 with outliers residing in cluster -1 on the right edge. Smaller ξ values extracted clusters not visible as valleys, whereas larger values are too restrictive. We expanded the clustering results by manually labeling clusters 13-15 (hatching pattern) to examine the properties of valleys not marked by ξ extraction method.

As with DBSCAN and HDBSCAN*, we summarize the identified cluster characteristics for $MinPts = 76, Xi = 0.01$ in comparison to the largest cluster, i.e. general behaviour, in Tab. 4.10. For distributions of features and anomalies in each cluster, we refer the interested reader to Appendix Fig. E.50-66.

label	cluster number															
	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
overall(dub)_au	1.27	1	1	1.11	1	1	1	0.99	1	1	0.99	0.99	1	1	0.99	0.99	1
overall(sing)_au	1.32	1	1	1.11	1	1	1.01	1	1	1	1	1	0.99	0.99	0.99	0.99	0.99
overall(wus)_au	1.69	1.01	1.01	1.28	1.01	1	1.01	1	1	1	1	1	0.99	0.99	0.99	0.99	0.99
overall(dub)_br	1.63	0.99	1	1	0.96	1	1	1.01	1	1	0.94	0.92	0.93	1	0.98	0.99	1
overall(sing)_br	1.54	1	1.01	1	0.97	1	1.01	1.02	1	1	0.94	0.94	0.94	1	0.98	0.99	0.99
overall(wus)_br	2.3	0.99	1.02	1.01	0.93	1	1.02	1.04	1	1	0.88	0.86	0.89	1	0.97	1	1
overall(dub)_cz	1.21	1	0.99	0.99	0.99	0.99	0.99	0.98	1	0.99	0.99	0.99	1	1.02	0.99	0.99	1
overall(sing)_cz	1.19	1	1	1	0.99	1	1	1	1	0.99	0.99	1	0.99	1.01	0.99	0.99	0.99
overall(wus)_cz	1.5	1	1	0.99	0.99	0.99	1	0.98	1	0.99	0.99	0.99	0.99	1.03	0.99	0.99	0.99
overall(dub)_jp	1.4	1	1.06	0.99	1	1	1	0.99	1	1	0.99	0.99	1	1	0.99	0.99	1
overall(sing)_jp	1.8	1	1.06	1	1	1	1	1	1	1	1	1	0.99	0.99	0.99	0.99	0.99
overall(wus)_jp	1.85	1.01	1.11	0.99	0.99	1	1	0.99	1	1	1.01	0.99	0.99	0.99	0.99	0.99	0.99
overall(dub)_us	2.11	1	1.03	0.98	1.01	0.99	1.05	1.07	1	1	0.99	0.98	1	1	0.98	0.98	1
overall(sing)_us	1.46	3.49	1.15	0.99	1.01	1	1.05	1.07	1	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99
overall(wus)_us	3.25	1.03	1.02	0.97	1.08	0.99	1.24	1.35	1	0.99	1	0.97	0.98	0.96	0.96	0.96	0.96
tcp2db(dub)	2.82	1	1	0.98	0.99	0.99	0.99	0.98	1	1	0.99	0.98	1	1	0.98	0.98	1
tcp2db(sing)	2.49	1	1	0.99	0.99	1	1	0.99	1	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99
tcp2db(wus)	125.39	1.09	1.07	0.78	0.65	0.89	0.83	0.5	1	0.81	0.69	0.53	0.63	0.56	0.53	0.57	0.52
overall_au	1.58	1	1.02	1.29	1.01	1	1.04	1.01	1	1	1	1	1	1	1	1	1
overall_br	2.44	0.99	1.02	1.02	0.94	1	1.02	1.04	1	1	0.88	0.86	0.89	1	0.98	1.01	1
overall_cz	1.52	1	1	1	0.99	0.99	1	0.99	1	0.99	0.99	1	1	1.05	1	1	1
overall_jp	1.72	1	1.11	1	1	1	1.01	1	1	1	1	1	1	1	1	1	1
overall_us	8.54	1.01	1.01	0.99	1.1	1	1.29	1.44	1	1	1	1	1	1	1	1	1
http2ws_au	1.8	1	1	1.29	1.01	1	1.02	1	1	1	1	1	1	1	1	1	1
http2ws_br	3.91	0.99	1.02	1.02	0.94	1	1.02	1.04	1	1	0.88	0.87	0.89	1	0.98	1	1
http2ws_cz	2.07	1	1	1	0.99	0.99	1	0.99	1	0.99	0.99	1	1	1.04	1	1	1
http2ws_jp	1.98	1	1.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
http2ws_us	16.36	1.01	1.01	1	1.1	1	1.28	1.43	1	1	1	1	1	1	1	1	1
tcp2ws_au	1.81	1	1	1.3	1.02	1	1.03	1.01	1	1	1	1	1	1	1	1	1
tcp2ws_br	3.94	0.99	1.02	1.02	0.94	1	1.02	1.04	1	1	0.88	0.86	0.89	1.01	0.98	1	1
tcp2ws_cz	2.09	1	1	1	0.99	0.99	1	0.99	1	0.99	0.99	1	1	1.04	1	1	1
tcp2ws_jp	2	1	1.11	1	1	1	1	1	1	1	1	1	1	1	1	1	1
tcp2ws_us	17.24	1.01	1.01	1	1.11	1	1.3	1.45	1	1	1	1	1	1	1	1	1
trace2ws_au	1.81	1	1.01	1.29	1.02	1.01	1.03	1.01	1	1.01	1.04	0.99	0.99	0.99	0.99	0.99	0.99
trace2ws_br	3.42	0.99	1.01	1.02	0.94	1.01	1.02	1.05	1	1.01	0.87	0.86	0.89	1.01	0.98	1.01	1.01
trace2ws_cz	1.77	0.97	0.94	1	0.37	0.13	0.93	0.94	1	0.56	0.99	0.99	1	1.08	1	1	1
trace2ws_jp	1.67	1	1.11	1	1	1	1.01	1.01	1	1	1	1	1	1	1	1	1
trace2ws_us	4.15	1.01	1.01	1	1.11	1	1.31	1.46	1	1	1	1	1	1	1	1	1

Table 4.10: *CLAudit OPTICS features interpretations table. Cluster −1 denotes outliers, green cells are general behaviour.*

cluster num.	size[%]	max. cont.[%]	interpretation
OPTICS			
-1	2.36	0.31	noise/outliers
0	2.12	0.02	weak positive
1	0.73	0.11	path/routing (local) + back-end TCP handshake
2	0.83	0.49	path/routing local
3	1.04	0.06	path/routing local
4	3.07	0.06	weak positive
5	0.89	0.22	path/routing local
6	0.33	0.10	path/routing local
7	68.53	1.03	general behaviour
8	2.18	0.04	weak positive
9	1.69	0.06	weak positive
10	1.03	0.11	weak positive
11	5.96	0.67	weak positive
12	0.40	0.10	weak positive
OPTICS - manually extracted clusters			
13	2.56	0.22	versions of cluster 13 (lower local db latency)
14	4.38	0.28	
15	1.92	0.05	
DBSCAN			
-1	4.46	0.31	noise/outliers
0	93.06	1.75	general behaviour
1	1.65	0.02	weak positive
2	0.84	0.49	path/routing (local)
HDBSCAN			
-1	12.24	0.75	noise/outliers
0	2.13	0.02	weak positive
1	0.84	0.49	path/routing local
2	2.96	0.06	weak positive
3	1.62	0.04	weak positive
4	8.27	1.00	path/routing (local); lower values
5	71.93	1.15	general behaviour

Table 4.11: *CLAudit clusters interpretations.*

4.3.3 Interpretation

Tab. 4.11 summarizes interpretation of clustering results from all algorithms, including manually labeled OPTICS clusters. We perform a *closed interpretation* as we use a Decision tree with a finite set of interpretations. Weak positives denote clusters not adhering to the Decision tree. We see that the manually marked clusters are special cases of active latency measurements with significantly lower latency to the local database (WUS). We can only speculate why the latency in 9% of the measurements (clusters $13-15$ combined). We conclude that in order to interpret *weak positive* clusters, we would either need more vantage points, or data center logs to give profound explanation of the events. Nevertheless, the interpretations table shows that the majority of identified clusters can be interpreted by the proposed Hi-Clust approach. We believe the approach can be used as a part of Cloud/Data Center benchmarking. Latency measurements from geographically distributed vantage points resemble diverse Cloud tenant locations, while latency event detection and interpretation provides valuable planning information for Cloud tenants.

4.3.4 Outliers analysis

In addition, we again computed Jaccard index of outliers/noise detected by algorithms:

$$J(DBSCAN, HDBSCAN) = 0.312 \tag{4.4}$$

$$J(OPTICS, DBSCAN) = 0.527 \tag{4.5}$$

$$J(OPTICS, HDBSCAN) = 0.190 \tag{4.6}$$

We can see that there is a significant, but not complete, overlap of unlabeled points marked as outliers. OPTICS can be easily fine-tuned to detect clusters that have lower density than the rest of the clusters, but higher density than surrounding points, e.g. clusters 1-2 in Fig.4.13. These clusters are marked as outliers by DBSCAN and HDBSCAN* (without additional fine-tuning). Another HDBSCAN* drawback compared to OPTICS is its default feature extraction process that does not extract both parent and child clusters.

Chapter 5

Conclusion

We presented the structural approach to application of density-based machine learning techniques to analyse datasets intrinsic hierarchical relations in two distinct scenarios - passive collection of network traces (MAWI dataset) and active latency probing (CLAudit dataset). We performed open (MAWI dataset) and closed (CLAudit dataset) interpretation. We identified and interpreted unknown patterns in network traffic by a blackbox approach. We focused on the structural approach, feature definition and fine-tuning of the algorithms.

We explored and analyzed both MAWI and CLAudit scenarios by various approaches. We applied batch and stream-based machine learning methods to anomaly detection on sets of features extracted from MAWI dataset, redefined previously used features and defined new ones. We analyzed and correlated results of statistical methods applied to fixed and variable length measurement windows of CLAudit latency measurements. We tested and analyzed clustering evaluation metrics, both recently defined and commonly used, to narrow the evaluation to two most recent DBCV and CDbw. We analyzed state-of-the-art research in the field of network and Cloud monitoring and anomaly detection, as well as unsupervised machine learning, and provided a solution to network and Cloud pattern and anomaly detection with interpretation without ground truth. Our results were summarized in various conference papers preceding this Thesis [3, 36, 37, 80, 81, 82, 83, 84].

This Thesis summarizes our research in application of unsupervised machine learning methods on network measurements for pattern and anomaly detection and provides the research community with methodology and guidelines that can be applied to any dataset. We restricted the application to network measurements due to our expert domain knowledge in networks, but with proper expertise it can be applied to any measurements.

The approach brings interesting network behaviour insights from a *macroscopic* view. We see primary application of our methodology and recommendations in network and Cloud monitoring systems. As we pointed out, most of the monitoring systems focus only on raising alarms defined by static thresholds. We provided an approach that is able to identify general behaviour, and provide guidelines for interpretation of suspicious events, regardless of the used dataset. Another use for this approach is suspicious activities identification by mobile operator networks monitoring traffic on antennas. This approach is universally applicable, either when we are able to define a suspicious event, i.e. cluster, characteristics through dataset features (MAWI), or when we identify direct relations between measurements (CLAudit).

This study falls into the category of applied unsupervised machine learning approaches. In general, unsupervised machine learning approaches are a less popular category of machine learning compared to supervised/semi-supervised because of the unclear definition of *the best* results. Performance of algorithms is often dependent on analyzed performance evaluation metrics. We tested our density-based clustering approach with state-of-the-art density-based metrics and showed that it is able to identify traffic patterns that are remarkably different than previously identified patterns. Our goal was to prove we are able to detect and interpret new patterns by novel methodology and approach. This conclusion is supported by high values of the (state-of-the-art) density-based clustering performance evaluation metric DBCV, which is able to assess quality of clustering of multi-dimensional datasets into arbitrary shaped clusters. The main contribution of this Thesis thus rests in *formulating a novel approach to unknown data pattern identification.*

The conclusions and results of this Thesis lead to further open research challenges:

1. suspicious events interpretation generalization by classification methods (taxonomy);
2. which unsupervised feature selection methods would facilitate interpretation;
3. which is the most suitable density-based algorithm for this task as we used fundamental algorithms and focused on the approach;
4. is there a way to replace the clustering with recently developed unsupervised deep learning methods e.g. generative models (GAN, DCGAN) or Auto-encoders;
5. what type of insights can we extract from datasets outside of the networking domain;
6. how to evaluate hierarchical density-based clustering results, i.e. include parent-child relations performance evaluation metrics.

In this Thesis we addressed the problem of detection and interpretation of unknown network traffic patterns. Our initial hypothesis was that if deviations in network measurements, passive or active, follow the hierarchical structure of network models, then they must be interpretable. Also, if they follow the hierarchical structure they must be detectable by hierarchical unsupervised machine learning models. We proved the hypothesis by state-of-the-art model evaluation and results interpretation. In summary, this work opens way to further advances in research in suspicious event-detection and brings novel insights to network pattern analysis.

Bibliography

[1] *MAWI(Measurement and Analysis on the WIDE Internet).* <https://mawi.wide.ad.jp/mawi/>.

[2] O. Tomanek and L. Kencl, "Claudit: Planetary-scale cloud latency auditing platform," in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, pp. 138–146, IEEE, 2013.

[3] P. Mulinka, P. Casas, and L. Kencl, "Hi-clust: Unsupervised analysis of cloud latency measurements through hierarchical clustering," in *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pp. 1–7, IEEE, 2018.

[4] A. C. Müller, S. Guido, *et al.*, *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016.

[5] S. Brunton, B. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *arXiv preprint arXiv:1905.11075*, 2019.

[6] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[8] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[9] W. Zhang, Q. Yang, and Y. Geng, "A survey of anomaly detection methods in networks," in *2009 International Symposium on Computer Network and Multimedia Technology*, pp. 1–3, IEEE, 2009.

[10] T. T. Nguyen and G. J. Armitage, "A survey of techniques for internet traffic classification using machine learning.," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1-4, pp. 56–76, 2008.

[11] R. P. Lippmann, I. Graf, D. Wyschogrod, S. E. Webster, D. J. Weber, and S. Gorton, "The 1998 darpa/afrl off-line intrusion detection evaluation," in *First International Workshop on Recent Advances in Intrusion Detection (RAID)*, 1998.

[12] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," *Computer networks*, vol. 34, no. 4, pp. 579–595, 2000.

[13] J. Korba, "Windows nt attacks for the evaluation of intrusion detection systems," tech. rep., MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 2000.

[14] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[15] *KDD99 dataset.* <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

[16] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," *arXiv preprint arXiv:1806.03517*, 2018.

[17] 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.

[18] S. Singhal and P. Yadav, "Evaluation of model using j-48 and other classifier on kddcup99 through performance metrics," in *International Conference on Advanced Informatics for Computing Research*, pp. 12–19, Springer, 2019.

[19] K. Ibrahimi and M. Ouaddane, "Management of intrusion detection systems based-kdd99: Analysis with lda and pca," in *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–6, IEEE, 2017.

[20] S. M. Almansob and S. S. Lomte, "Addressing challenges for intrusion detection system using naive bayes and pca algorithm," in *2017 2nd International Conference for Convergence in Technology (I2CT)*, pp. 565–568, IEEE, 2017.

[21] *NSL-KDD dataset.* <https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/>.

[22] J. Vanerio and P. Casas, "Ensemble-learning approaches for network security and anomaly detection," in *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, pp. 1–6, ACM, 2017.

[23] P. Casas, F. Soro, J. Vanerio, G. Settanni, and A. D'Alconzo, "Network security and anomaly detection with big-dama, a big data analytics framework," in *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, pp. 1–7, IEEE, 2017.

[24] P. Casas, J. Vanerio, and K. Fukuda, "Gml learning, a generic machine learning model for network measurements analysis," in *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–9, IEEE, 2017.

[25] P. Casas and J. Vanerio, "Super learning for anomaly detection in cellular networks," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, IEEE, 2017.

[26] *Nagios.* <https://www.nagios.org/>.

[27] *Zabbix.* <https://www.zabbix.com/>.

[28] *Cacti.* <https://www.cacti.net/>.

[29] *ThousandEyes.* <https://www.thousandeyes.com/>.

[30] *Amazon CloudWatch*. <https://aws.amazon.com/cloudwatch/>.

[31] *Azure Monitor*. <https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-azure-monitor>.

[32] *Monitis*. <https://www.monitis.com/>.

[33] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, "Gmone: A complete approach to cloud monitoring," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2026–2040, 2013.

[34] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.

[35] K. Alhamazani, R. Ranjan, K. Mitra, F. Rabhi, P. P. Jayaraman, S. U. Khan, A. Guabtni, and V. Bhatnagar, "An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art," *Computing*, vol. 97, no. 4, pp. 357–377, 2015.

[36] P. Mulinka and L. Kencl, "Learning from cloud latency measurements," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 1895–1901, IEEE, 2015.

[37] O. Tomanek, P. Mulinka, and L. Kencl, "Multidimensional cloud latency monitoring and evaluation," *Computer Networks*, vol. 107, pp. 104–120, 2016.

[38] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, 2012.

[39] J. Dromard, G. Roudière, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2016.

[40] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[41] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*, pp. 25–71, Springer, 2006.

[42] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE transactions on emerging topics in computing*, vol. 2, no. 3, pp. 267–279, 2014.

[43] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[44] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[45] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172, Springer, 2013.

[46] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod record*, vol. 28, pp. 49–60, ACM, 1999.

[47] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, p. 19, 2017.

[48] D. Moulavi, P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander, "Density-based clustering validation," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 839–847, SIAM, 2014.

[49] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[50] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[51] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[52] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[53] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality scheme assessment in the clustering process," in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 265–276, Springer, 2000.

[54] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: Finding the optimal partitioning of a data set," in *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 187–194, IEEE, 2001.

[55] M. Halkidi, M. Vazirgiannis, and C. Hennig, "Method-independent indices for cluster validation and estimating the number of clusters," in *Handbook of Cluster Analysis*.

[56] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*, pp. 911–916, IEEE, 2010.

[57] M. Hassani and T. Seidl, "Using internal evaluation measures to validate the quality of diverse stream clustering algorithms," *Vietnam Journal of Computer Science*, vol. 4, no. 3, pp. 171–183, 2017.

[58] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," in *ACM CoNEXT '10*, (Philadelphia, PA), December 2010.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[60] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 410–420, 2007.

[61] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[62] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[63] C. J. Van Rijsbergen, "Foundation of evaluation," *Journal of documentation*, vol. 30, no. 4, pp. 365–373, 1974.

[64] *L7-filter*. `<http://l7-filter.sourceforge.net/protocols>`.

[65] J. Mazel, R. Fontugne, and K. Fukuda, "A taxonomy of anomalies in backbone network traffic," in *Proceedings of 5th International Workshop on TRaffic Analysis and Characterization*, TRAC 2014, pp. 30–36, 2014.

[66] R. E. Bellman, *Adaptive control processes: a guided tour*, vol. 2045. Princeton university press, 2015.

[67] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[68] C. F. Van Loan, "Generalizing the singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 13, no. 1, pp. 76–83, 1976.

[69] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 94, 2018.

[70] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, pp. 507–514, 2006.

[71] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*, pp. 1151–1157, ACM, 2007.

[72] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[73] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "L2, 1-norm regularized discriminative feature selection for unsupervised," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[74] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[75] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[76] *fpc: Flexible Procedures for Clustering.* <https://rdrr.io/cran/fpc/>.

[77] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[78] J. Liu and K. Fukuda, "An evaluation of darknet traffic taxonomy," *Journal of Information Processing*, vol. 26, pp. 148–157, 2018.

[79] *CLAudit: Planetary-scale cloud latency auditing.* <http://claudit.feld.cvut.cz/>.

[80] P. Mulinka and P. Casas, "Stream-based machine learning for network security and anomaly detection," in *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, pp. 1–7, 2018.

[81] P. Mulinka and P. Casas, "Adaptive network security through stream machine learning," in *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, pp. 4–5, 2018.

[82] P. Mulinka, S. Wassermann, G. Marín, and P. Casas, "Remember the good, forget the bad, do it fast-continuous learning over streaming data," 2018.

[83] P. Casas, P. Mulinka, and J. Vanerio, "Should i (re) learn or should i go (on)? stream machine learning for adaptive defense against network attacks," in *Proceedings of the 6th ACM Workshop on Moving Target Defense*, pp. 79–88, 2019.

[84] S. Wassermann, T. Cuvelier, P. Mulinka, and P. Casas, "Adam & ral: Adaptive memory learning and reinforcement active learning for network monitoring," 2019.

Appendix A

List of topical publications

Publication I is a direct output of project VII (see Appendix. C). Publication II is a direct output of project III. Publication III is a direct output of project VII. Publication IV is the aggregate output of projects IX, X and XI. All authors contributed equally to the respective listed papers.

IF–journal papers

I) Multidimensional cloud latency monitoring and evaluation
Ondrej Tomanek, Pavol Mulinka and Lukas Kencl
Computer Networks 107 (2016) pp. 104–120. Elsevier, 2016
Citations except self–citations: 5 (WoS), 6 (Scopus), 14 (Google Scholar)

II) Adaptive and Reinforcement Learning Approaches for Online Network Monitoring and Analysis
Sarah Wassermann, Thibaut Cuvelier, Pavol Mulinka and Pedro Casas
IEEE Transactions on Network and Service Management, 2020
Citations except self–citations: 0 (WoS), 0 (Scopus), 0 (Google Scholar)

Scopus and WoS–indexed conference papers

I) HUMAN - Hierarchical Clustering for Unsupervised Anomaly Detection & Interpretation
Pavol Mulinka, Pedro Casas, Kensuke Fukuda, and Lukas Kencl
11th international Conferece on Network of the Future, 2020

II) WhatsThat? On the Usage of Hierarchical Clustering for Unsupervised Detection & Interpretation of Network Attacks
Pavol Mulinka, Kensuke Fukuda, Pedro Casas, and Lukas Kencl
The 5th International Workshop on Traffic Measurements for Cybersecurity, 2020

III) Should I (re)Learn or Should I Go(on)? Stream Machine Learning for Adaptive Defense against Network Attacks
Pedro Casas, Pavol Mulinka and Juan Vanerio
The 6th ACM Workshop on Moving Target Defense, 2019, London, UK
Citations except self–citations: 0 (WoS), 0 (Scopus), 1 (Google Scholar)

IV) Continuous and Adaptive Learning over Big Streaming Data for Network Security
Pavol Mulinka, Pedro Casas and Juan Vanerio
IEEE International Conference on Cloud Networking CLOUDNET, 2019 International Conference on, 2019, Coimbra, PT
Citations except self–citations: 0 (WoS), 0 (Scopus), 0 (Google Scholar)

V) ADAM & RAL: Adaptive Memory Learning and Reinforcement Active Learning for Network Monitoring
Sarah Wassermann, Thibaut Cuvelier, Pavol Mulinka and Pedro Casas
15th International Conf. on Network and Service Management, 2019, Halifax, CA
Citations except self–citations: 0 (WoS), 0 (Scopus), 0 (Google Scholar)

VI) Remember the Good, Forget the Bad, do it Fast: Continuous Learning over Streaming Data
Pavol Mulinka, Sarah Wassermann, Gonzalo Marín and Pedro Casas
@NeurIPS 2018 Workshops, Workshop on Continual Learning, 2018, Montreal, CA
Citations except self–citations: 0 (WoS), 0 (Scopus), 0 (Google Scholar)

VII) Hi-Clust: Unsupervised Analysis of Cloud Latency Measurements through Hierarchical Clustering
Pavol Mulinka, Pedro Casas and Lukas Kencl
IEEE International Conference on Cloud Networking CLOUDNET, 2018 International Conference on, 2018, Tokyo, JP
Citations except self–citations: 0 (WoS), 0 (Scopus), 1 (Google Scholar)

VIII) Stream-based Machine Learning for Network Security and Anomaly Detection
Pavol Mulinka and Pedro Casas
Proc. of the Workshop on Big Data Analytics and ML for Data Comm. Net., BigDAMA@SIGCOMM, 2018, Budapest, HU
Citations except self–citations: 0 (WoS), 4 (Scopus), 6 (Google Scholar)

IX) Learning from Cloud latency measurements
Pavol Mulinka and Lukas Kencl
Communication Workshop (ICCW), 2015 IEEE Int. Conf. on, 2015, London, UK
Citations except self–citations: 1 (WoS), 1 (Scopus), 2 (Google Scholar)

Posters

I) Adaptive Network Security through Stream Machine Learning
Pavol Mulinka and Pedro Casas
Proceedings of the ACM SIGCOMM '18 Posters and Demos, 2018, Budapest, HU
Citations except self–citations: 0 (WoS), 1 (Scopus), 0 (Google Scholar)

II) Stream-based Machine Learning for Network Security and Anomaly Detection
Pavol Mulinka and Pedro Casas
8th PhD School on Traffic Monitoring and Analysis (TMA), 2018, Vienna, AU
Citations except self–citations: 0 (WoS), 0 (Scopus), 0 (Google Scholar)

III) Cloud Latency Measurements Interpretation
Pavol Mulinka, Ondrej Tomanek and Lukas Kencl
4th PhD School on Traffic Monitoring and Analysis (TMA), 2014, London, UK
Citations except self–citations: 0 (WoS), 0 (Scopus), 0 (Google Scholar)

Appendix B

List of other publications

Scopus and WoS–indexed conference papers

I) Speaker identification by K-Nearest Neighbors: Application of PCA and LDA prior to KNN

Juraj Kacur, Radoslav Vargic, and Pavol Mulinka

Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on, 2011, Sarajevo, BiH

Citations except self–citations: 0 (WoS), 5 (Scopus), 15 (Google Scholar)

Appendix C

List of projects

Contributing to this thesis was work conducted in a scope of projects II, IV, V, VI, VII, VIII, IX. Project III was an independently conducted industry collaboration with partner research organizations.

I) J. Klemsa, L. Vojtech, P. Mulinka, J. Riha, P. Hnyk, I. Trumova, "Practical Privacy-Preserving Data Collection and Utilization using Provable Cryptographic Tools", SGS19/109/OHK3/2T/13, Co-Investigator role, 1/2019 - 1/2020, Czech Technical University in Prague

II) P. Mulinka, K. Fukuda, "Syslog causality analysis", NII Tokyo internship funded by NII International Internship Program Co–investigator role, 3/2019 – 9/2019

III) P. Mulinka, S. Park, D. Perino "Understanding the Quality of Service of Mobile Network depending on the socioeconomic factors", Telefonica I+D Barcelona internship funded by Fundación Universidad-Empresa, Co–investigator role, 11/2018 – 2/2019

IV) P. Mulinka, P. Casas, "Big-DAMA project internship", partially funded by scientific bilateral cooperation grant Aktion Österreich-Tschechien, AÖCZ-Semesterstipendien, ref.num. ICM-2017-08733 Co–investigator role, 3/2018 – 8/2018, AIT Vienna

V) P. Mulinka, J. Klemsa, O. Tomanek and L. Kencl, "Privacy Protection and Machine–Learning Utilization of IoT Data in Cloud", SGS18/077/OHK3/1T/13, Principal investigator role, 1/2018 – 1/2019, Czech Technical University in Prague

VI) P. Mulinka, O. Tomanek, J. Klemsa and L. Kencl, "Smart–home IoT and Cloud Telemetry Datamining", SGS17/091/OHK3/1T/13, Principal investigator role, 1/2017 – 1/2018, Czech Technical University in Prague

VII) O. Tomanek, P. Mulinka, Z. Kouba, V. Uhlir, E. Marku and L. Kencl, “Cloud Performance Analysis and Improvement”, SGS15/153/OHK3/2T/13, Co–Investigator role, 1/2015 – 1/2017, Czech Technical University in Prague

VIII) P. Mulinka and L. Kencl, “Metrics for Automated Detection of Cloud Anomalous Behavior”, Cisco Systems, Inc., Collaborative Research Program, Principal Investigator role, 1/2013 – 1/2014, Czech Technical University in Prague

IX) O. Tomanek, J. Stanek, P. Mulinka and L. Kencl, “Methods Enhancing Work with Cloud Data”, SGS13/139/OHK3/2T/13, Co–investigator role, 1/2013 – 1/2015, Czech Technical University in Prague

Appendix D

Other results

I) CLAudit Real Time Data and Detected suspicious events visualization
Pavol Mulinka
Available: <http://claudit.feld.cvut.cz/rtdata.php>

Appendix E

Complementary interpretation boxplots

This section includes boxplots showing distributions of features and anomalies for detected clusters in Sec. 3.4 :

1. MAWI DBSCAN feature boxplots E.1,2,5
2. MAWI DBSCAN anomaly boxplots E.2,4,6
3. MAWI HDBSCAN* feature boxplots E.7,9,11
4. MAWI HDBSCAN* anomaly boxplots E.8,10,12
5. MAWI OPTICS feature boxplots E.13,15,17,19,21,23,25,27,29,31,33,35,37
6. MAWI OPTICS anomaly boxplots E.14,16,18,20,22,24,26,28,30,32,34,36,38
7. CLAudit DBSCAN feature boxplots E.39-42
8. CLAudit HDBSCAN* feature boxplots E.43-49
9. CLAudit OPTICS feature boxplots E.50-66

We did not include the boxplots in the main text due to their size and rather summarized the information in the final interpretation tables for MAWI Tab. 4.7 and CLAudit Tab. 4.11 scenarios.

Figure E.1: *MAWI DBSCAN cluster -1 feature interpretation*

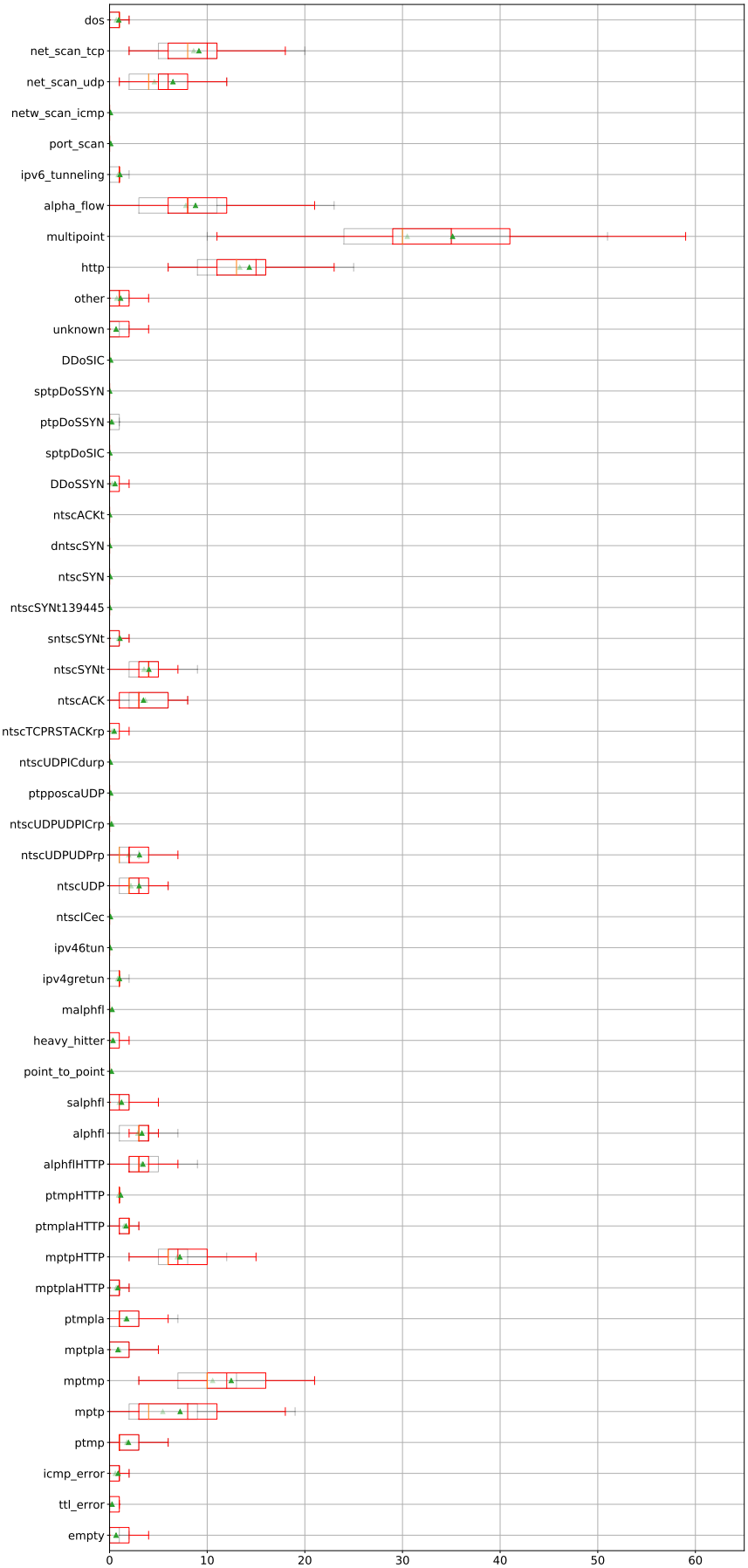


Figure E.2: *MAWI DBSCAN cluster -1 anomaly interpretation*

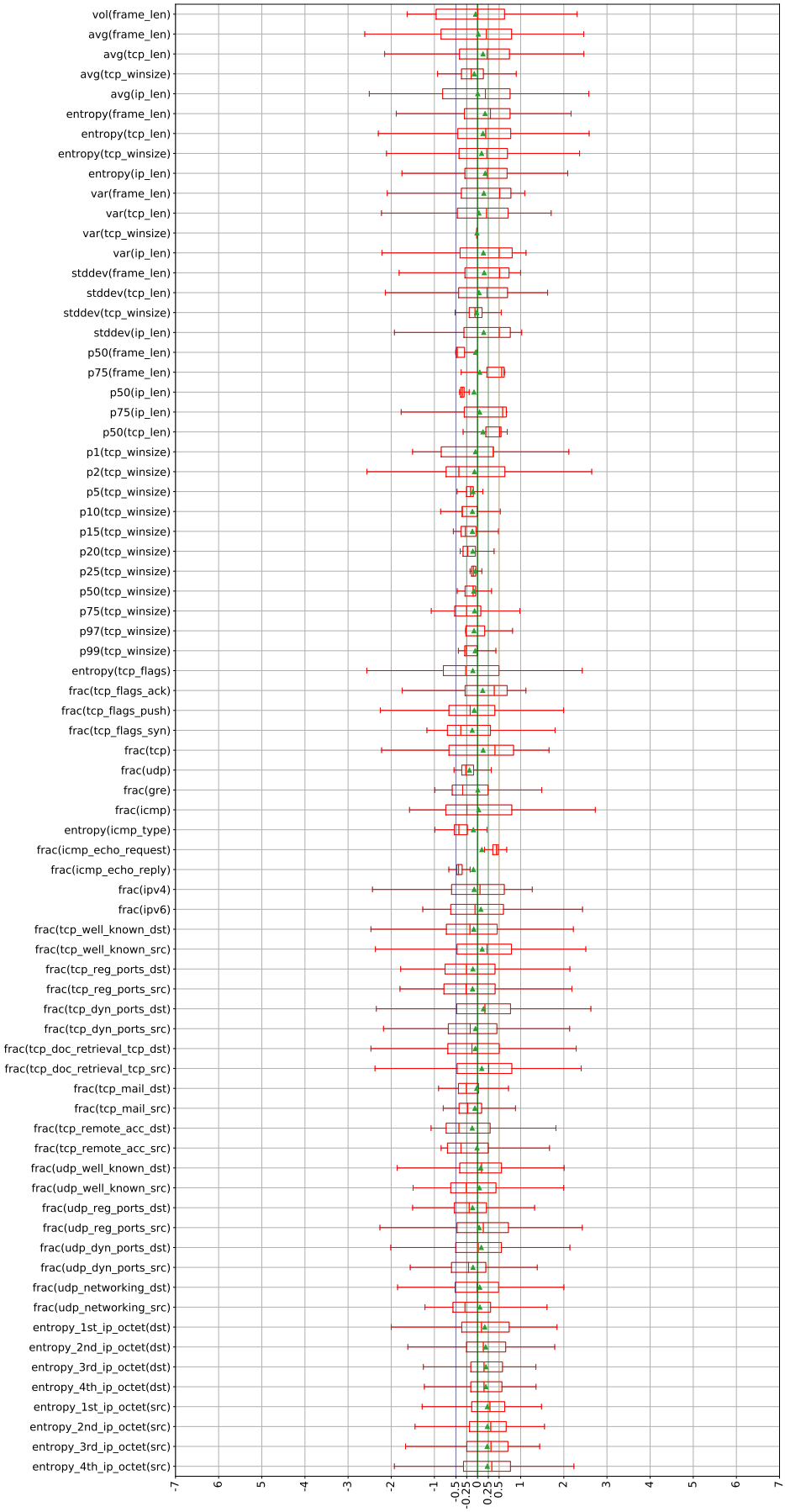


Figure E.3: *MAWI DBSCAN cluster 0 feature interpretation*

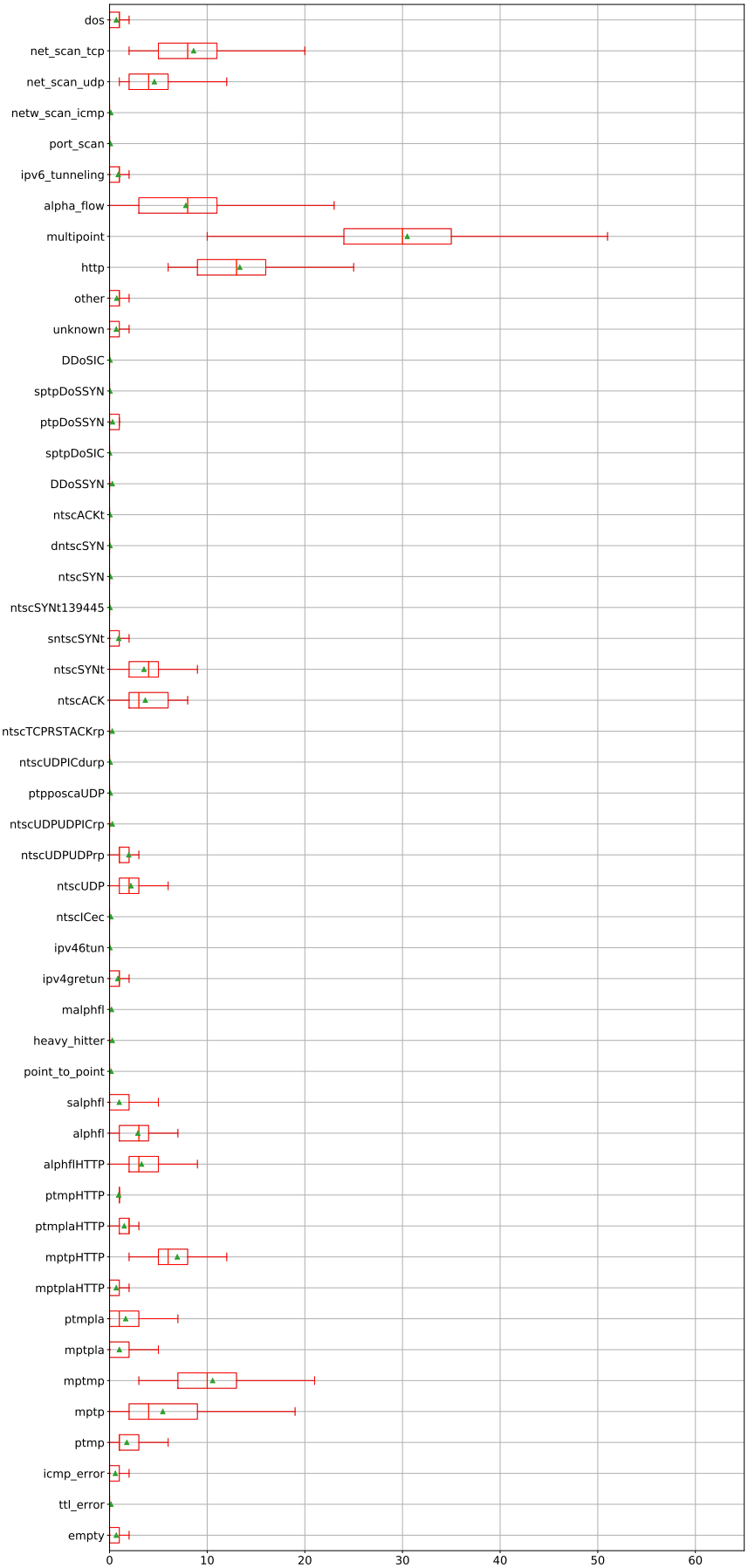


Figure E.4: *MAWI DBSCAN cluster 0 anomaly interpretation*

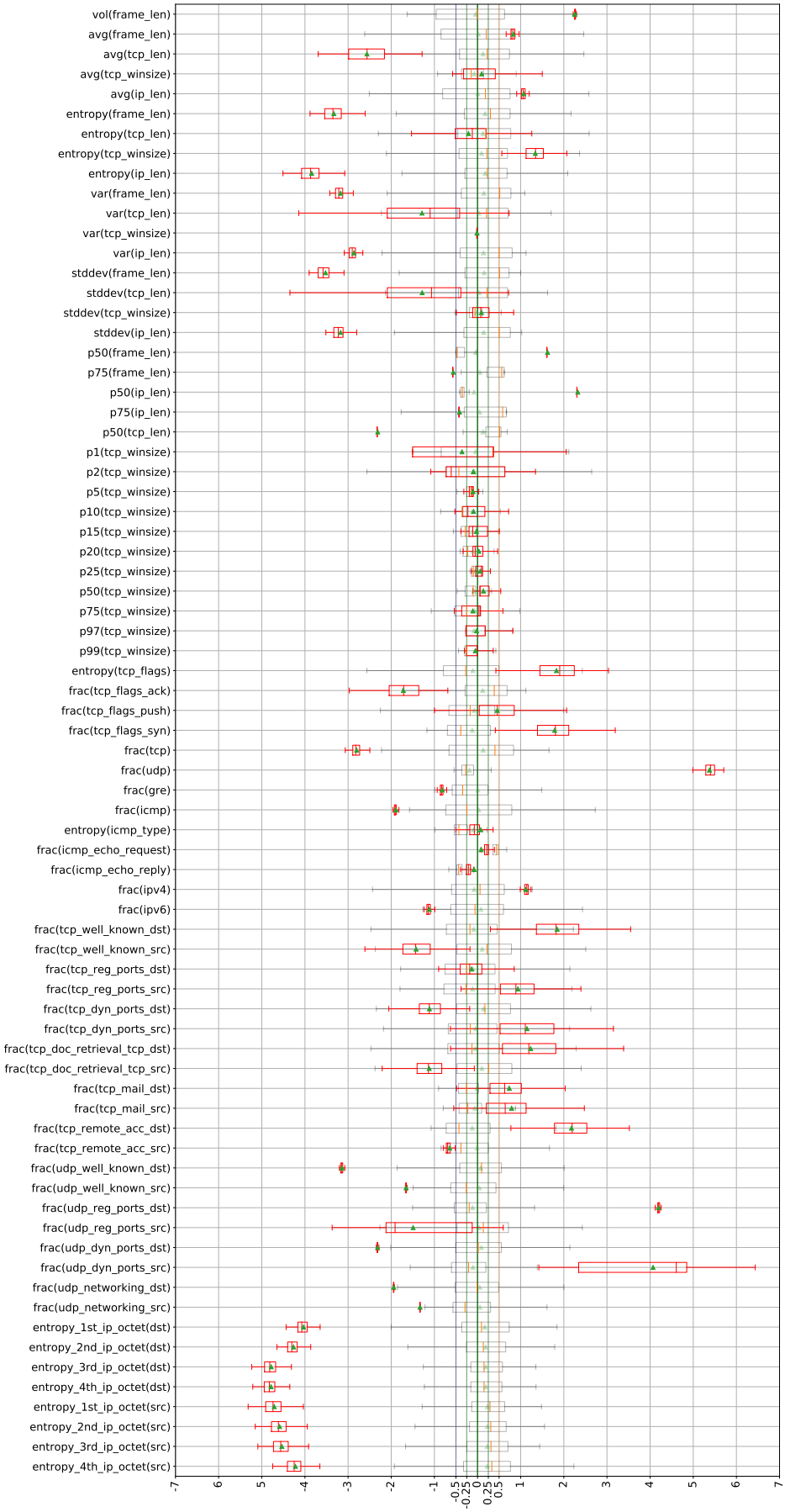


Figure E.5: *MAWI DBSCAN cluster 1 feature interpretation*

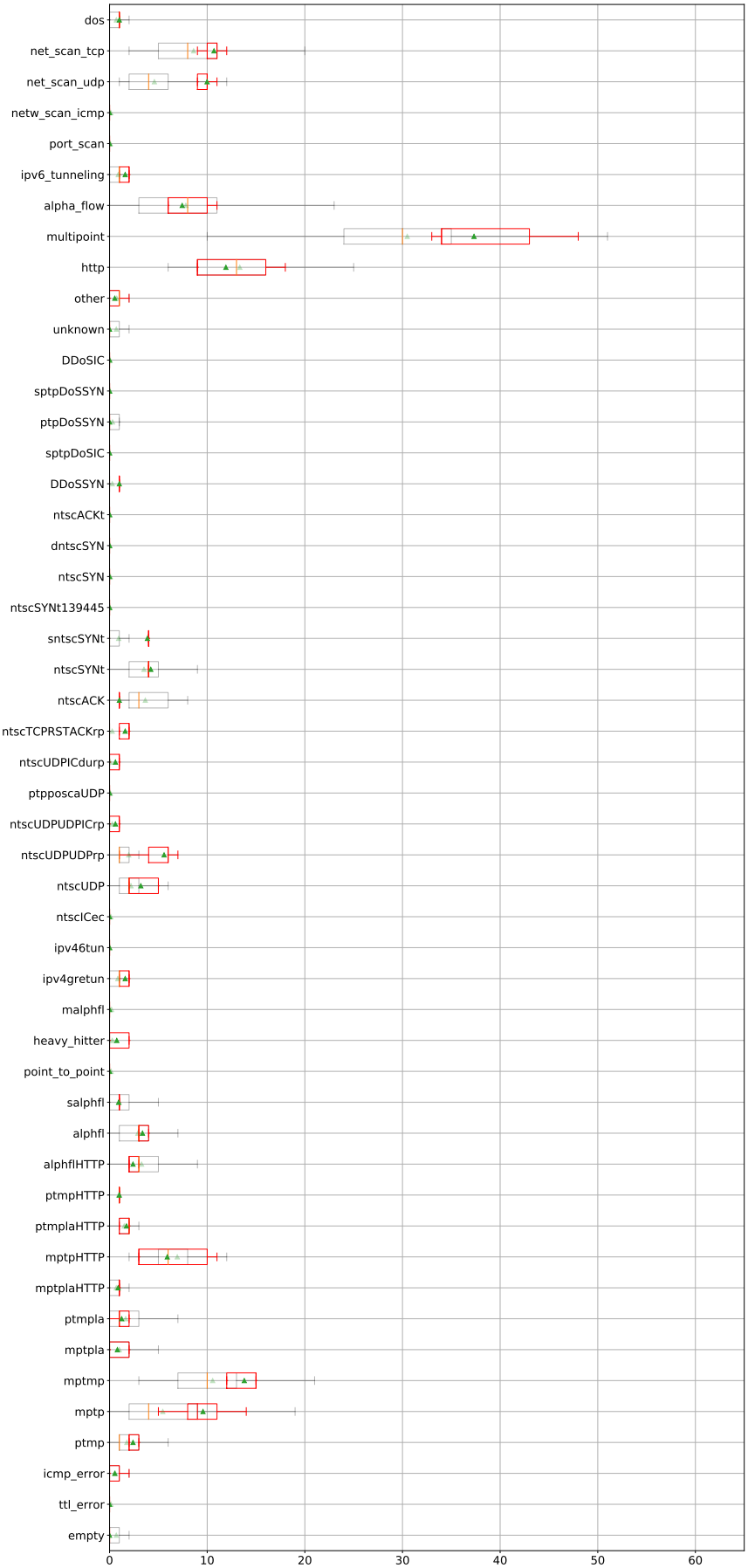


Figure E.6: *MAWI DBSCAN cluster 1 anomaly interpretation*

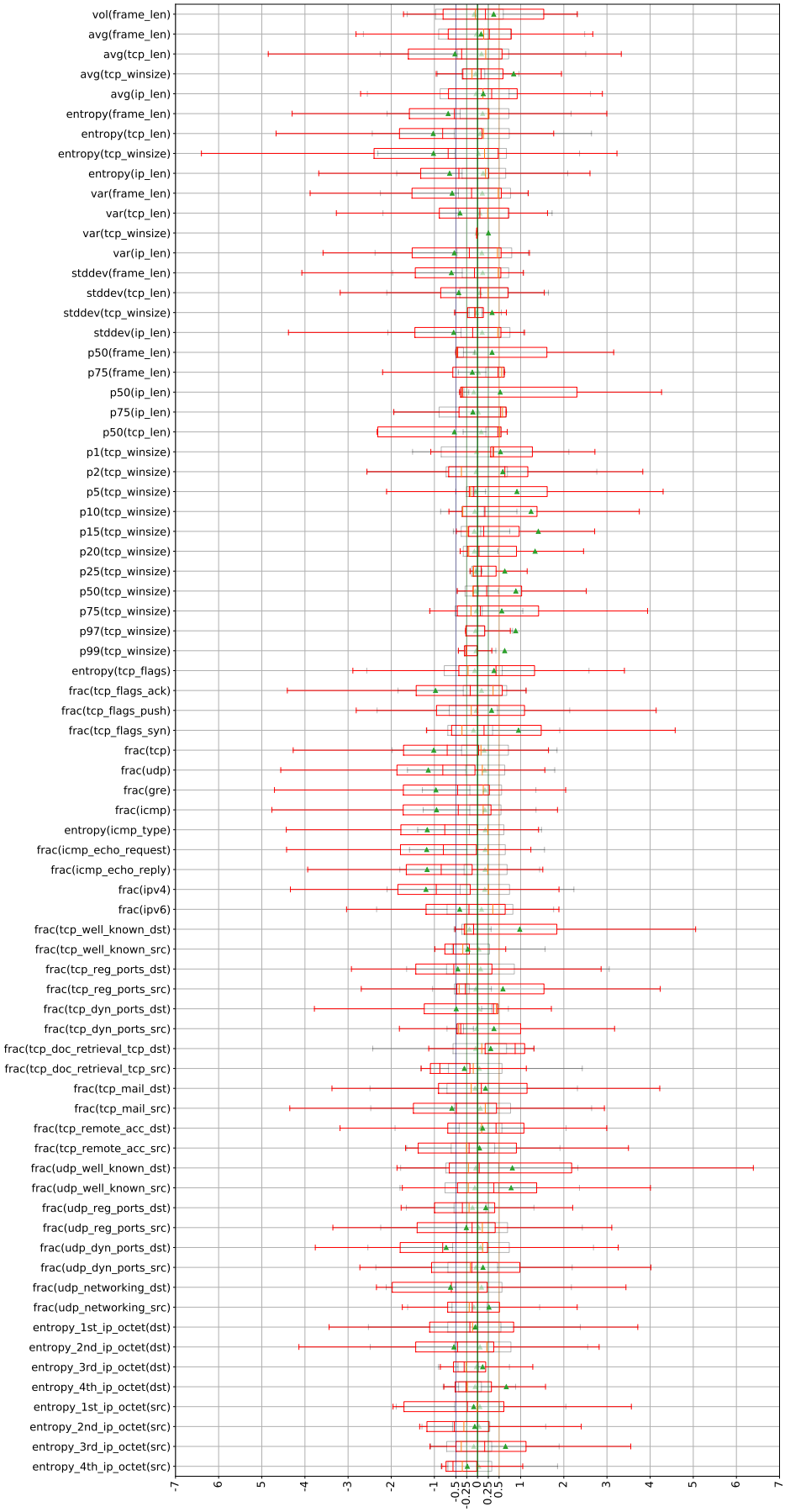


Figure E.7: *MAWI HDBSCAN cluster -1 feature interpretation*

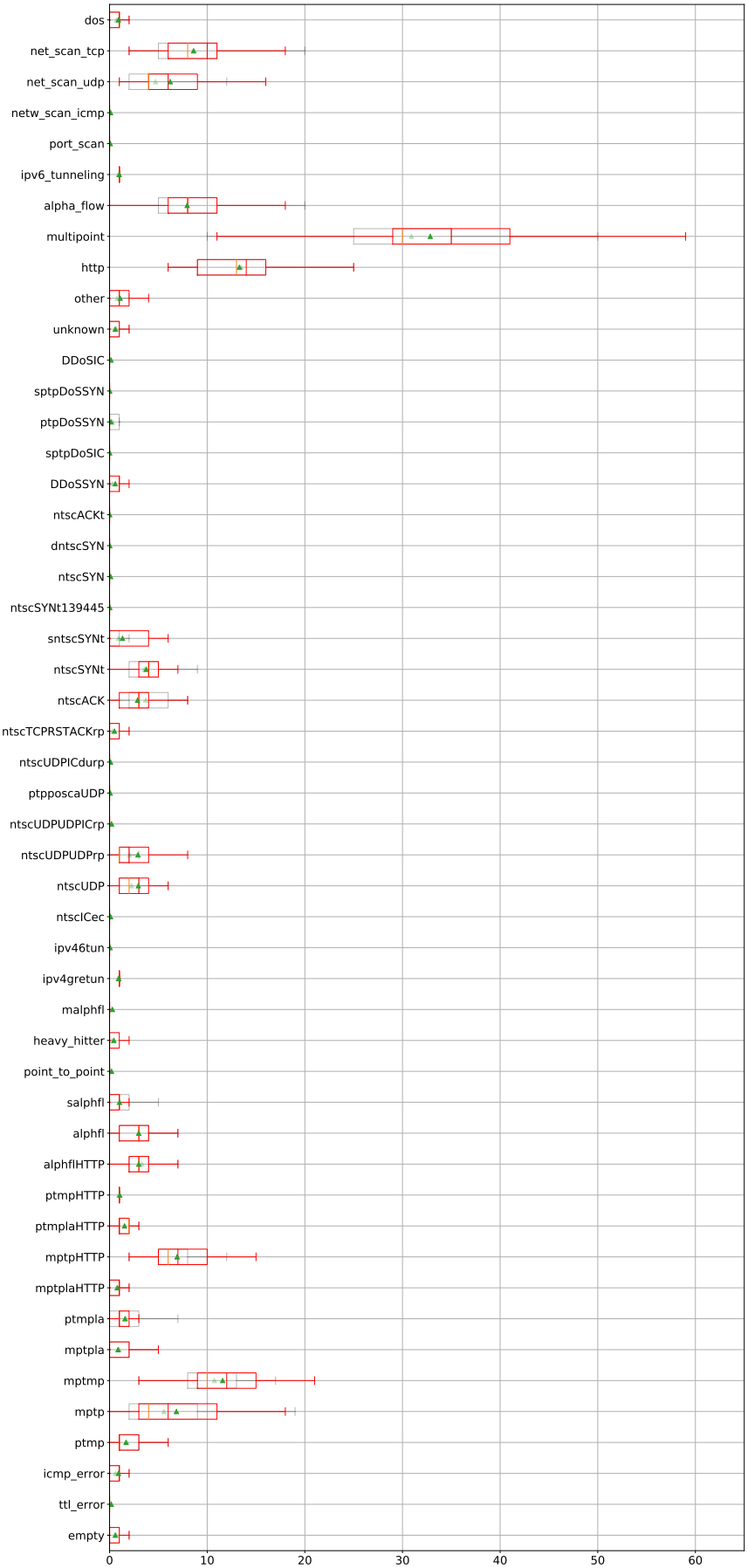


Figure E.8: *MAWI HDBSCAN cluster -1 anomaly interpretation*

Figure E.9: *MAWI HDBSCAN cluster 0 feature interpretation*

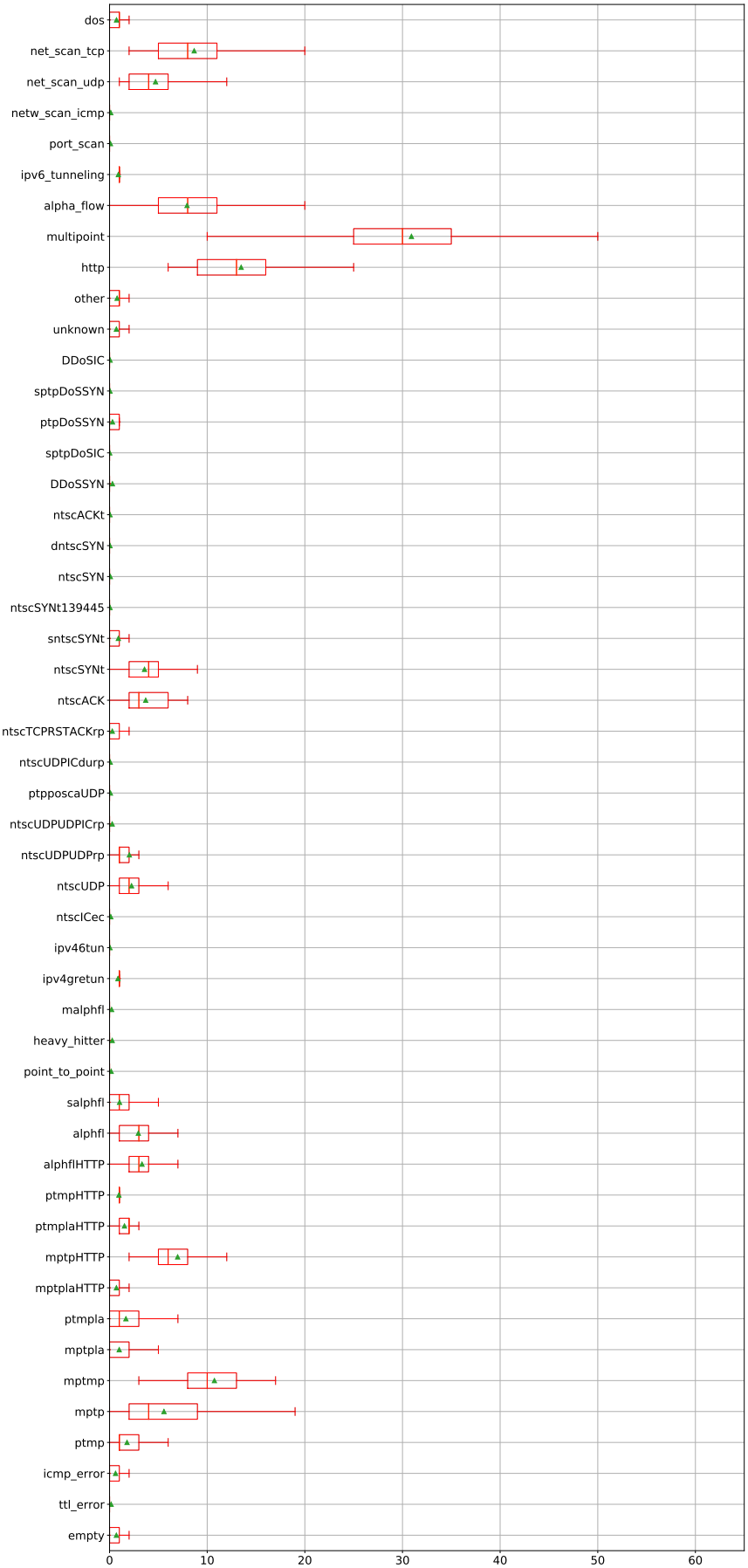


Figure E.10: *MAWI HDBSCAN cluster 0 anomaly interpretation*

Figure E.11: *MAWI HDBSCAN cluster 1 feature interpretation*

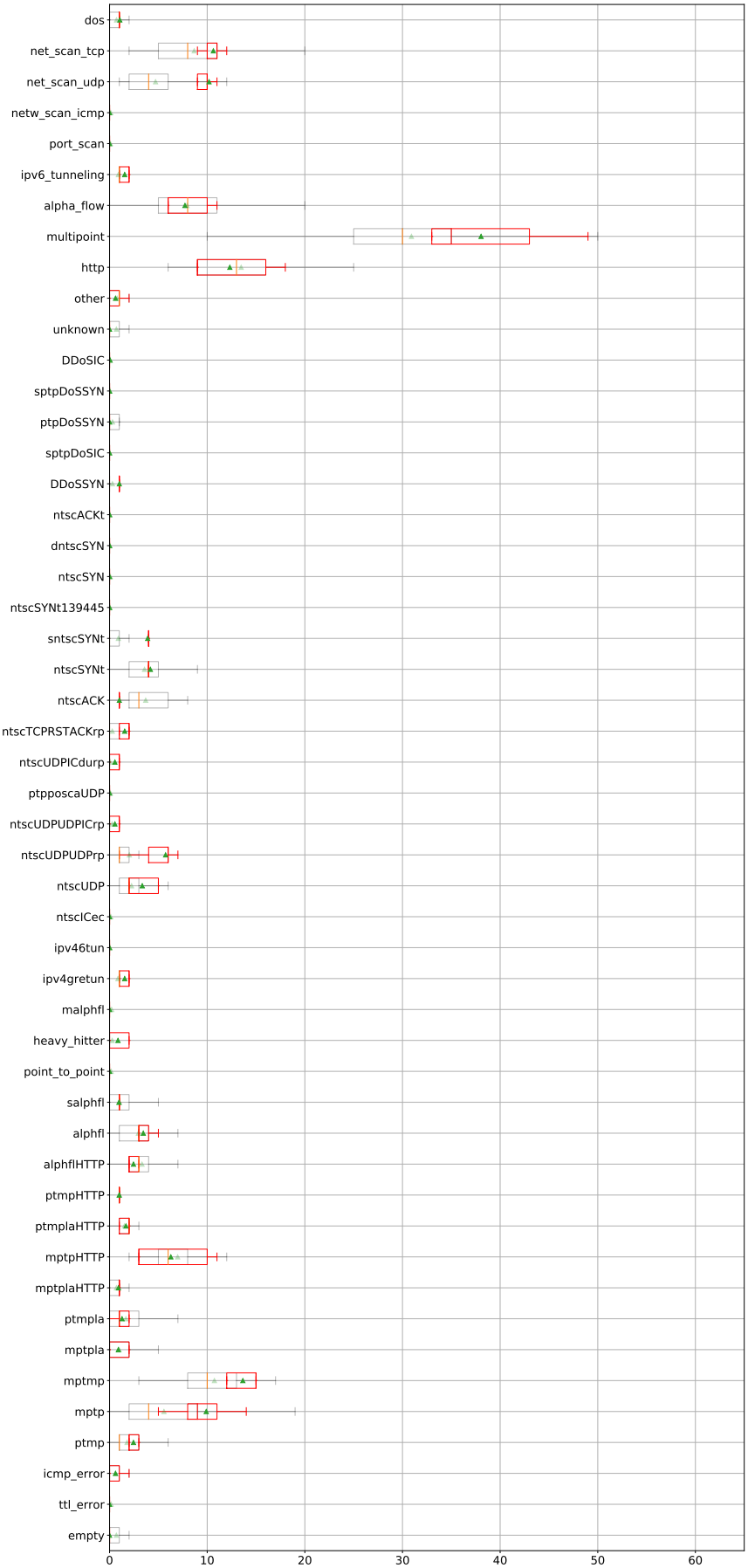


Figure E.12: *MAWI HDBSCAN cluster 1 anomaly interpretation*

Figure E.13: *MAWI OPTICS cluster -1 feature interpretation*

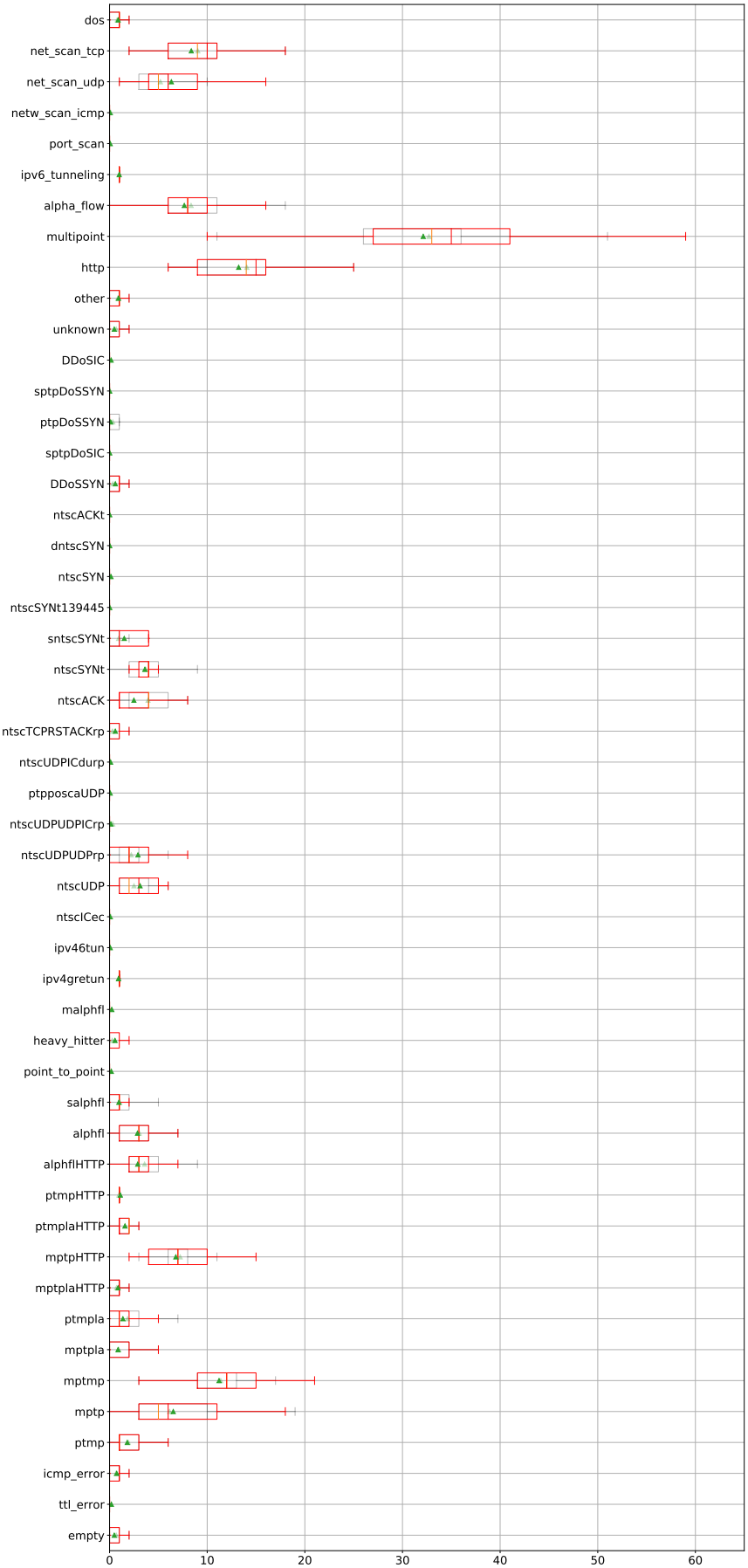


Figure E.14: *MAWI OPTICS cluster -1 anomaly interpretation*

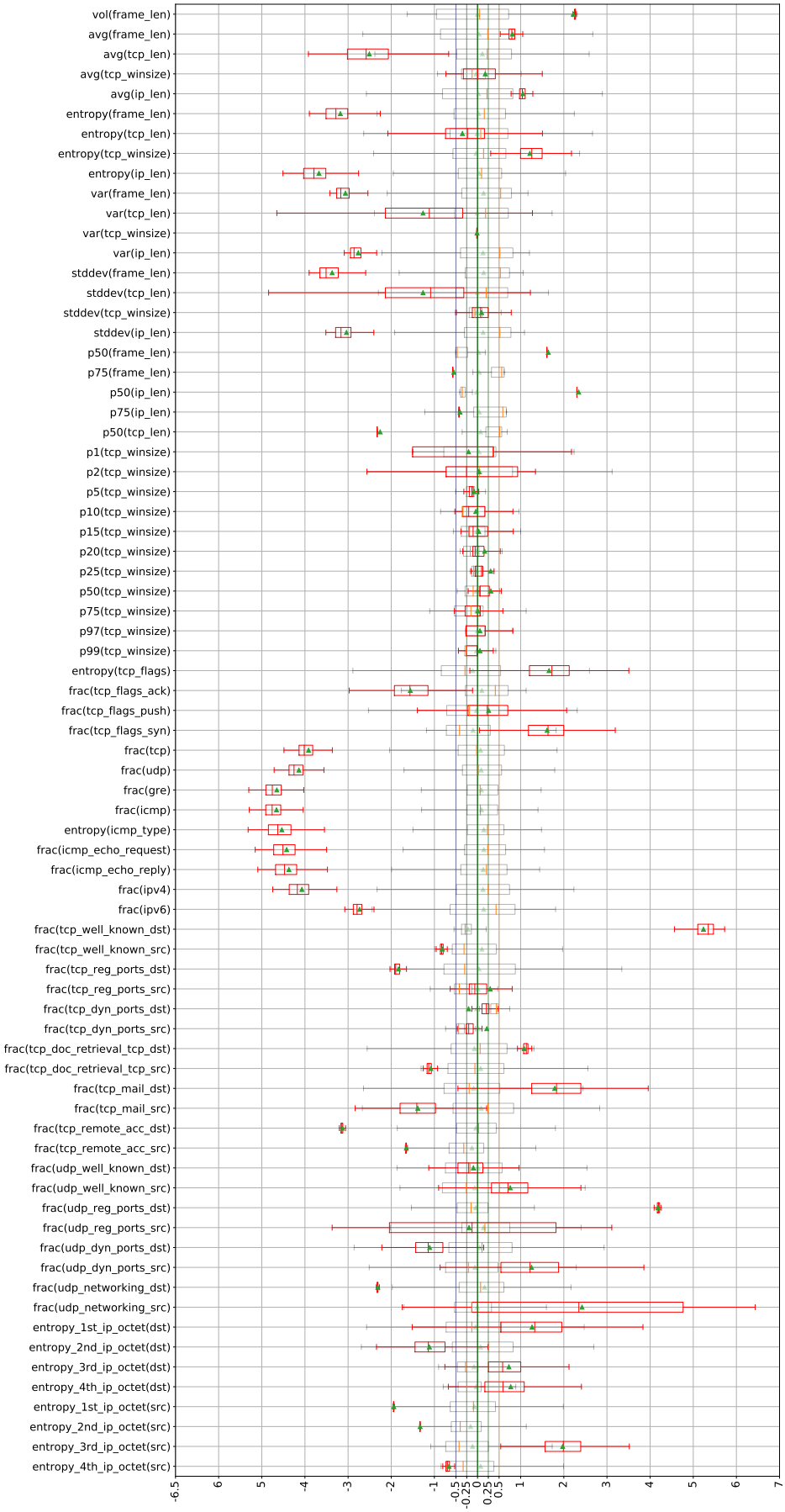


Figure E.15: *MAWI OPTICS cluster 0 feature interpretation*

Figure E.16: *MAWI OPTICS cluster 0 anomaly interpretation*

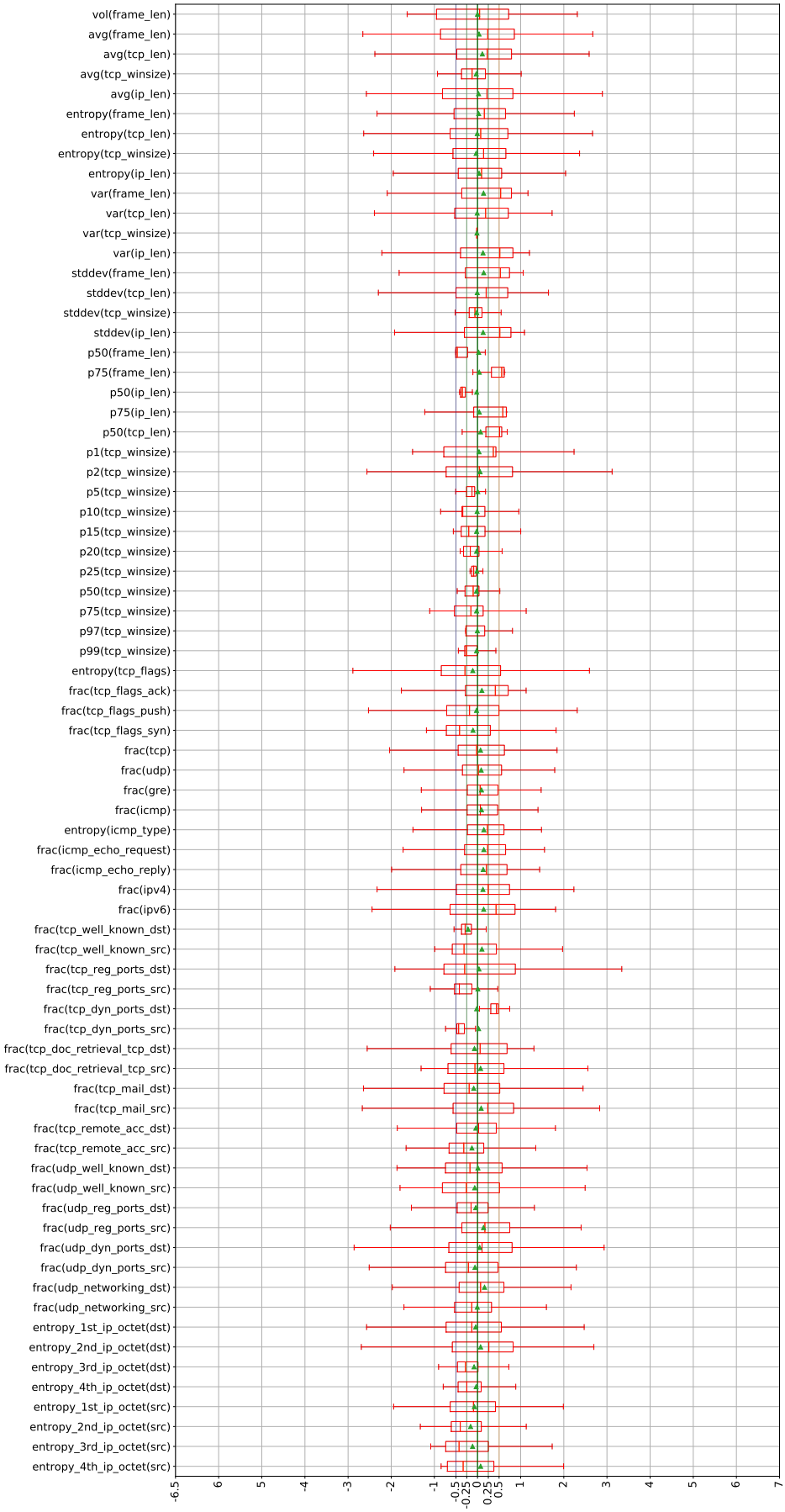


Figure E.17: *MAWI OPTICS cluster 1 feature interpretation*

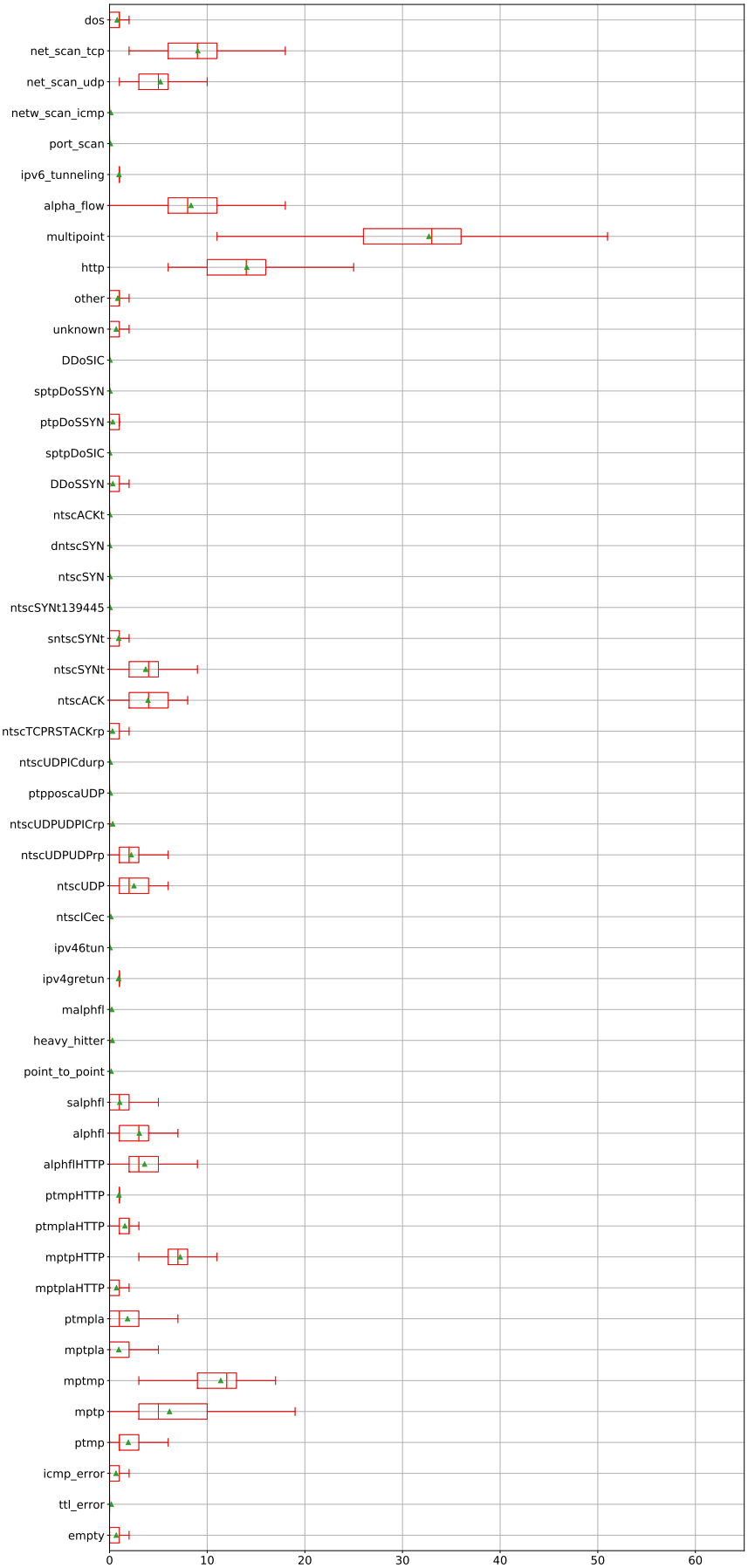


Figure E.18: *MAWI OPTICS cluster 1 anomaly interpretation*

Figure E.19: *MAWI OPTICS cluster 2 feature interpretation*

Figure E.20: *MAWI OPTICS cluster 2 anomaly interpretation*

Figure E.21: *MAWI OPTICS cluster 3 feature interpretation*

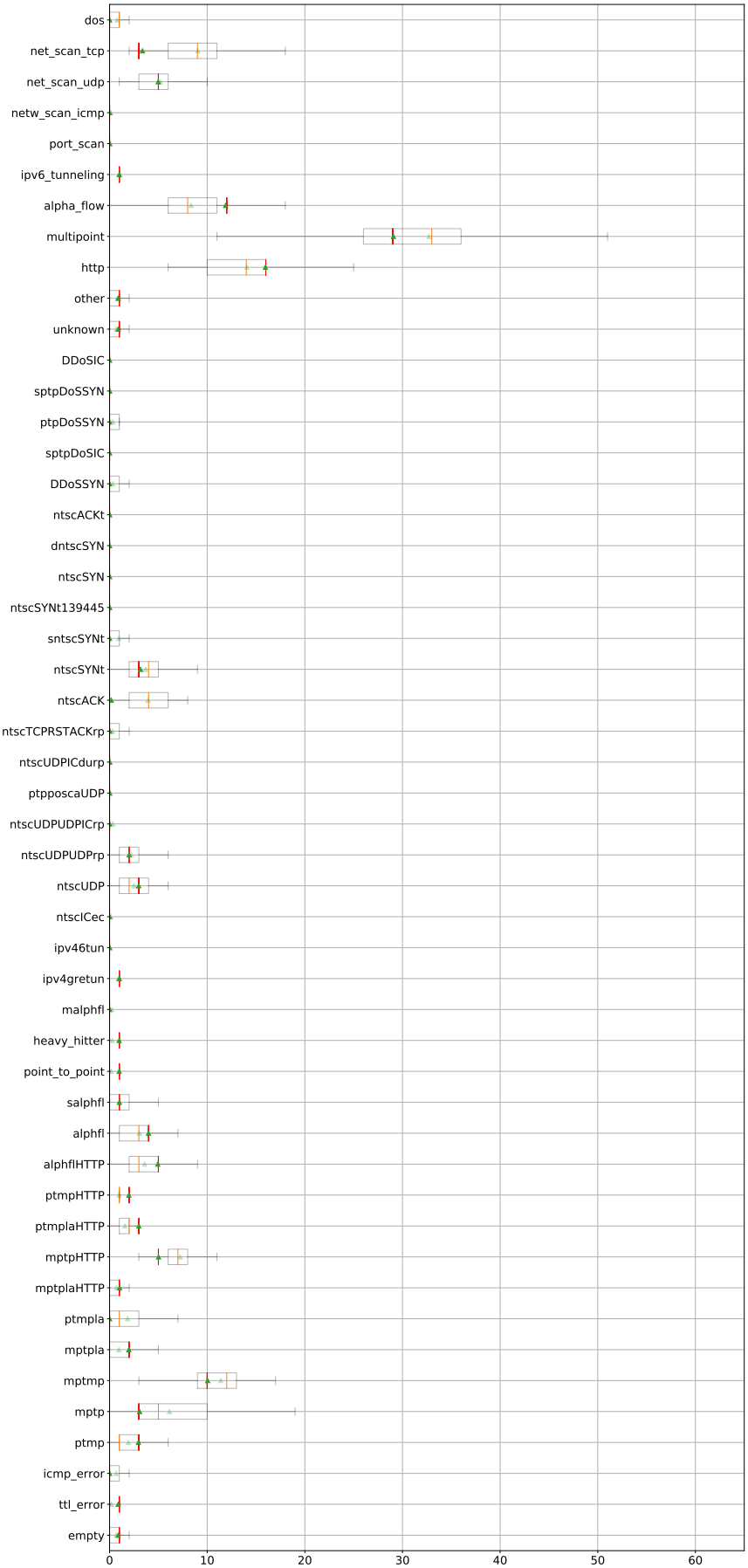


Figure E.22: *MAWI OPTICS cluster 3 anomaly interpretation*

Figure E.23: *MAWI OPTICS cluster 4 feature interpretation*

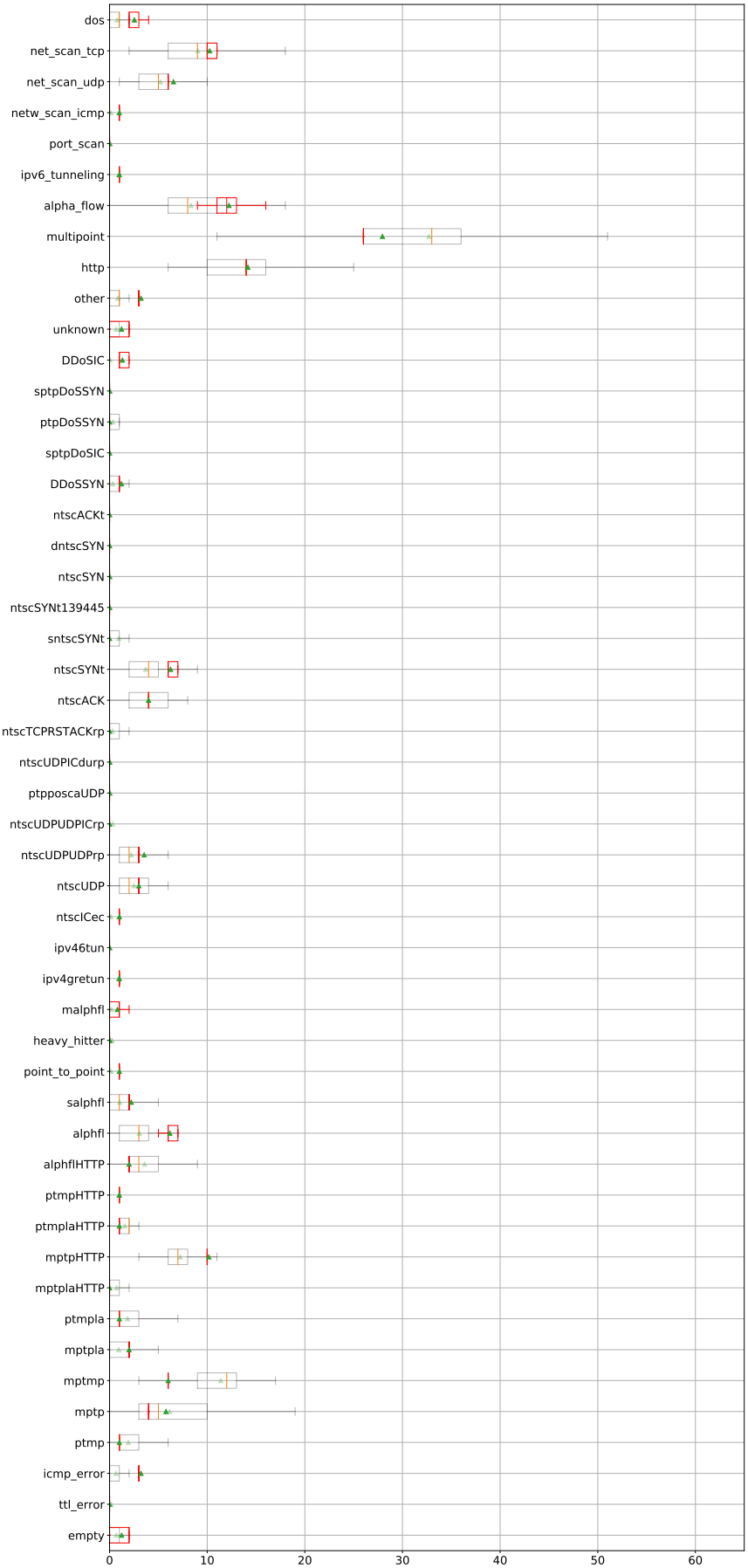


Figure E.24: *MAWI OPTICS cluster 4 anomaly interpretation*

Figure E.25: *MAWI OPTICS cluster 5 feature interpretation*

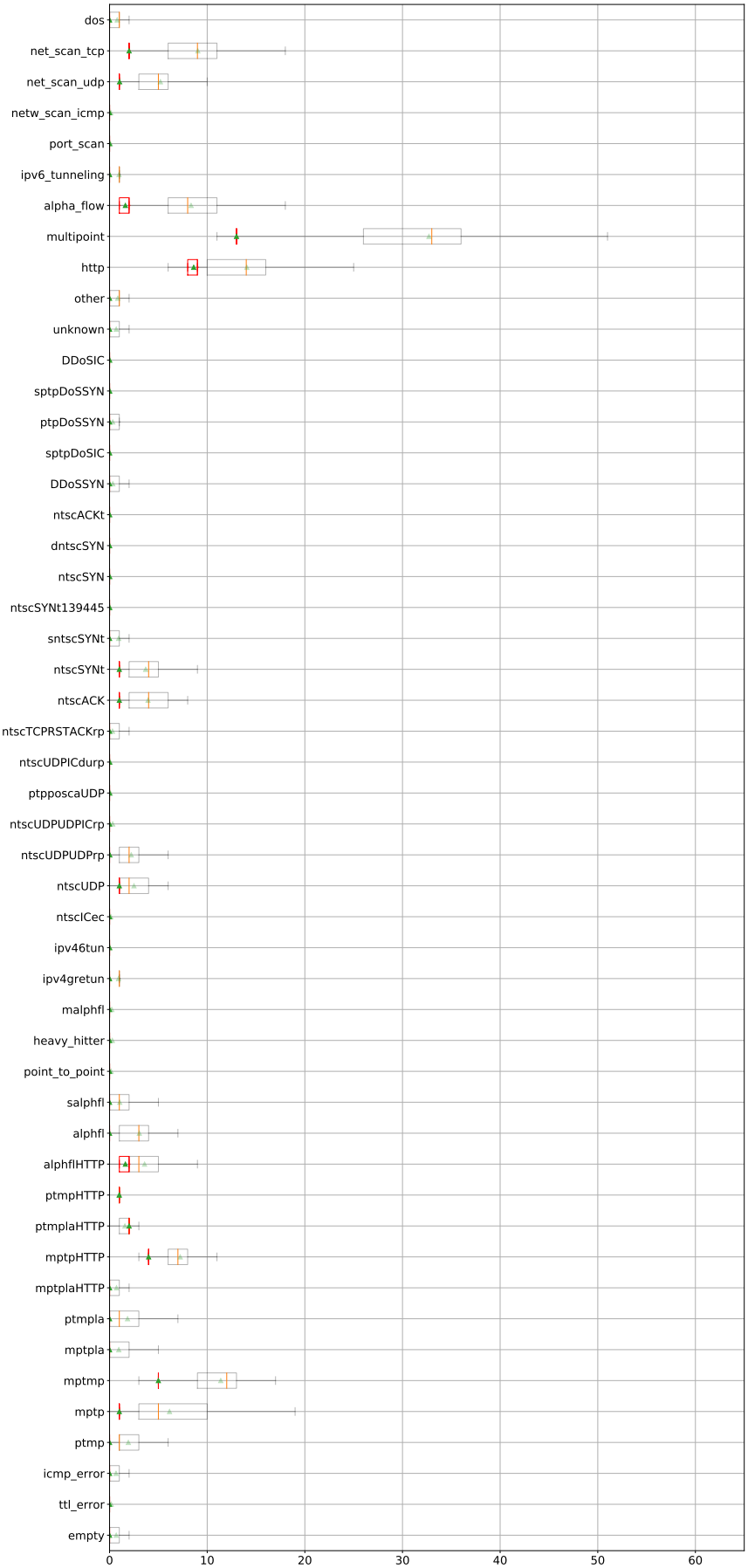


Figure E.26: *MAWI OPTICS cluster 5 anomaly interpretation*

Figure E.27: *MAWI OPTICS cluster 6 feature interpretation*

Figure E.28: *MAWI OPTICS cluster 6 anomaly interpretation*

Figure E.29: *MAWI OPTICS cluster 7 feature interpretation*

Figure E.30: *MAWI OPTICS cluster 7 anomaly interpretation*

Figure E.31: *MAWI OPTICS cluster 8 feature interpretation*

Figure E.32: *MAWI OPTICS cluster 8 anomaly interpretation*

Figure E.33: *MAWI OPTICS cluster 9 feature interpretation*

Figure E.34: *MAWI OPTICS cluster 9 anomaly interpretation*

Figure E.35: *MAWI OPTICS cluster 10 feature interpretation*

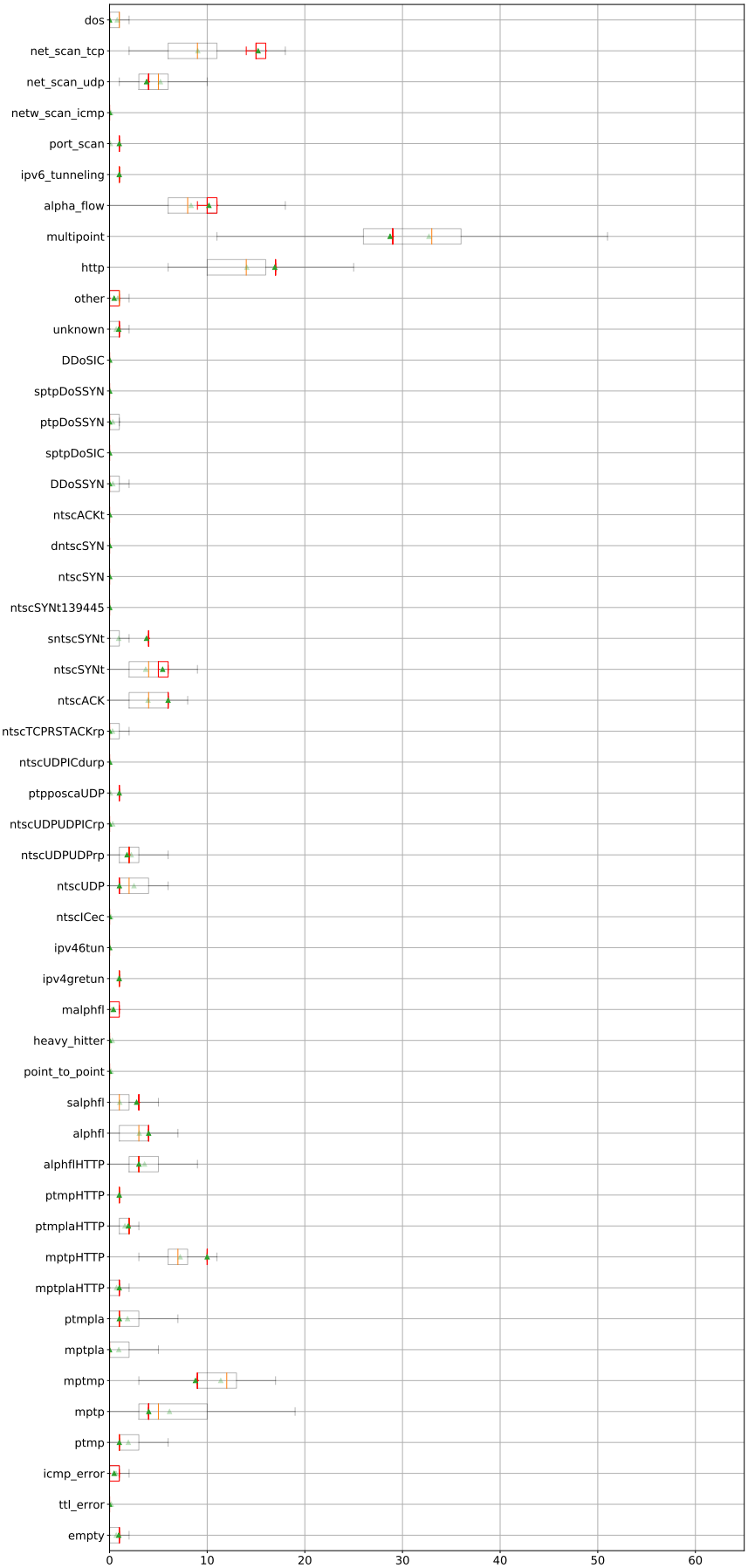


Figure E.36: *MAWI OPTICS cluster 10 anomaly interpretation*

Figure E.37: *MAWI OPTICS cluster 11 feature interpretation*

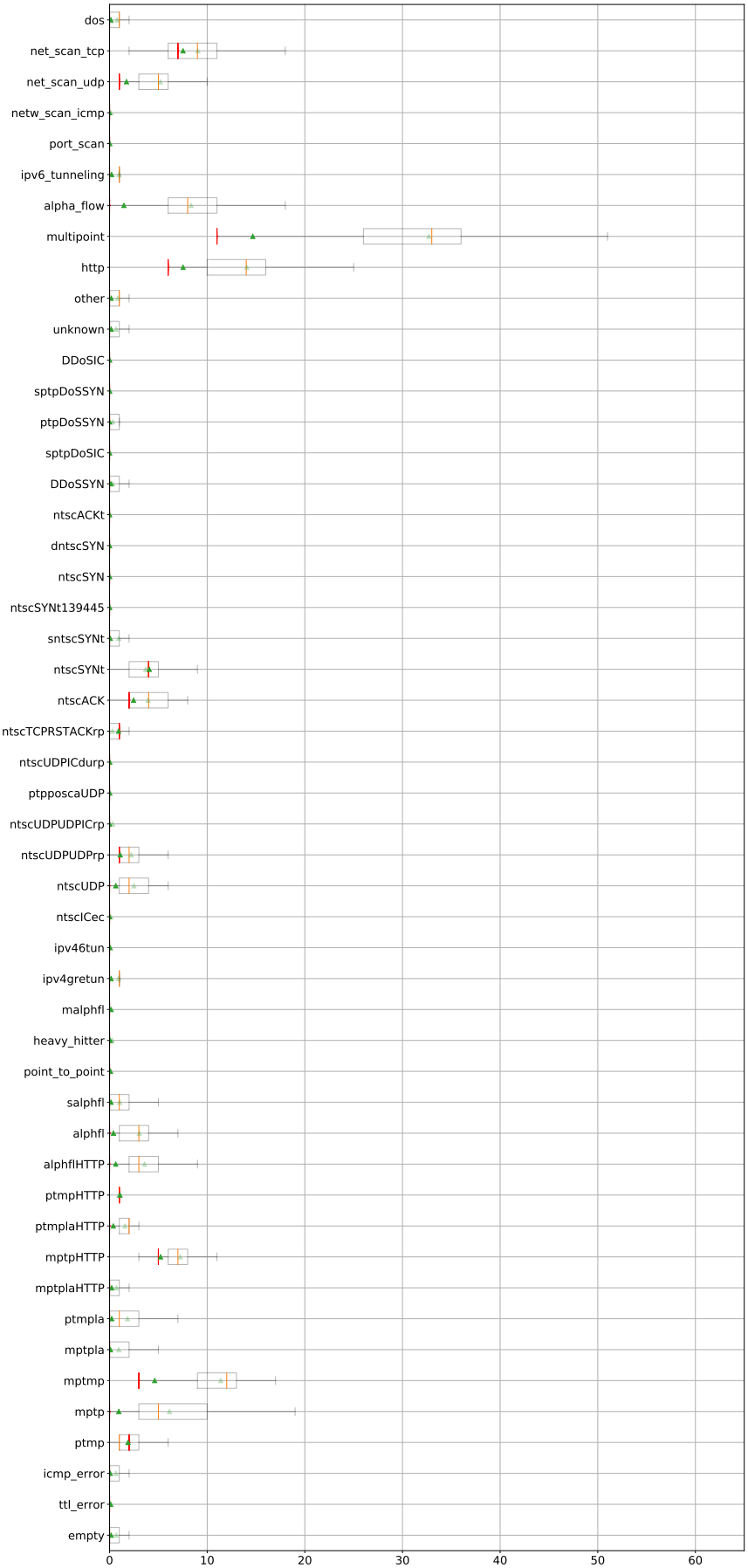


Figure E.38: *MAWI OPTICS cluster 11 anomaly interpretation*

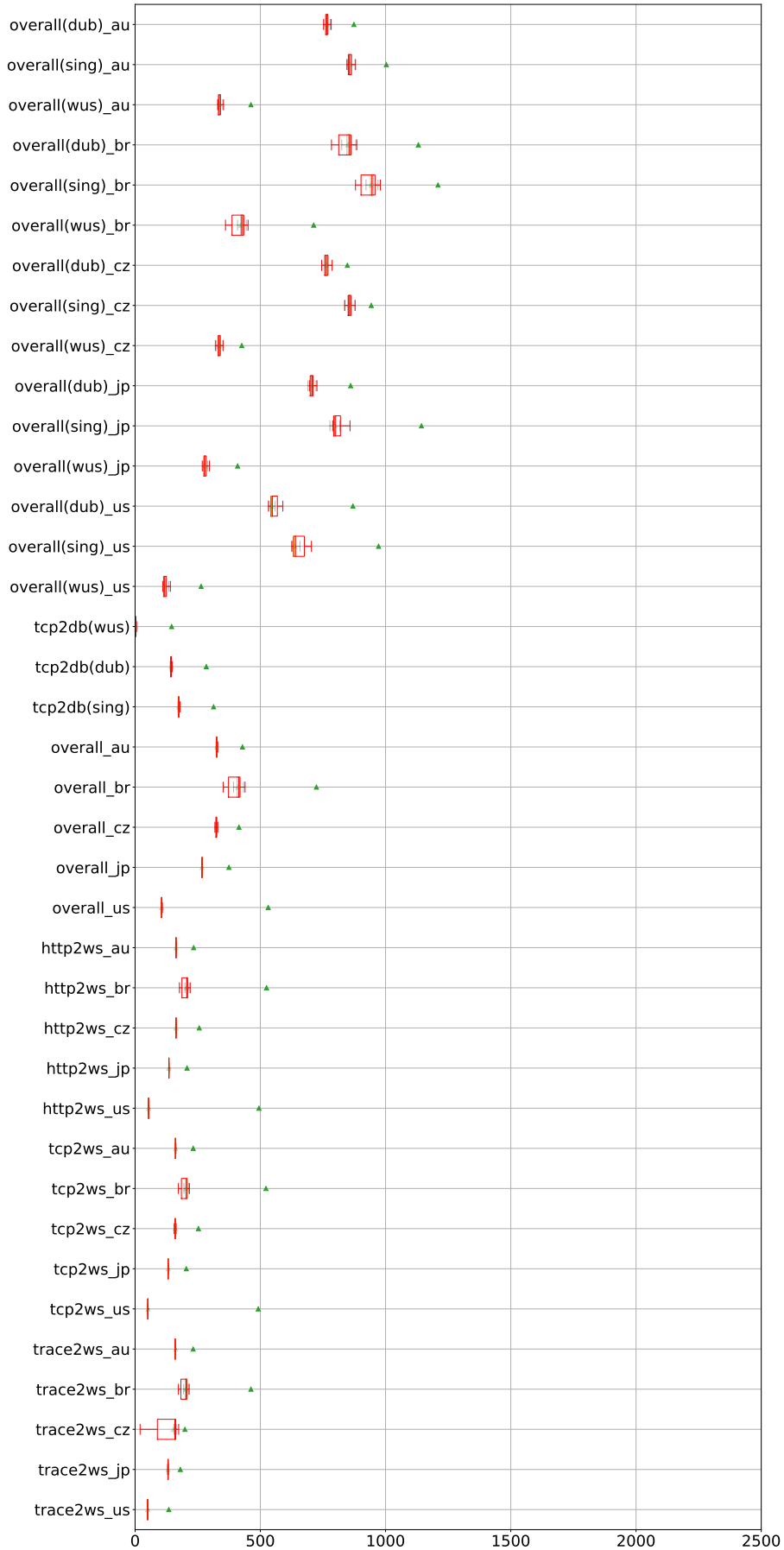


Figure E.39: *CLAudit DBSCAN cluster -1 feature interpretation*

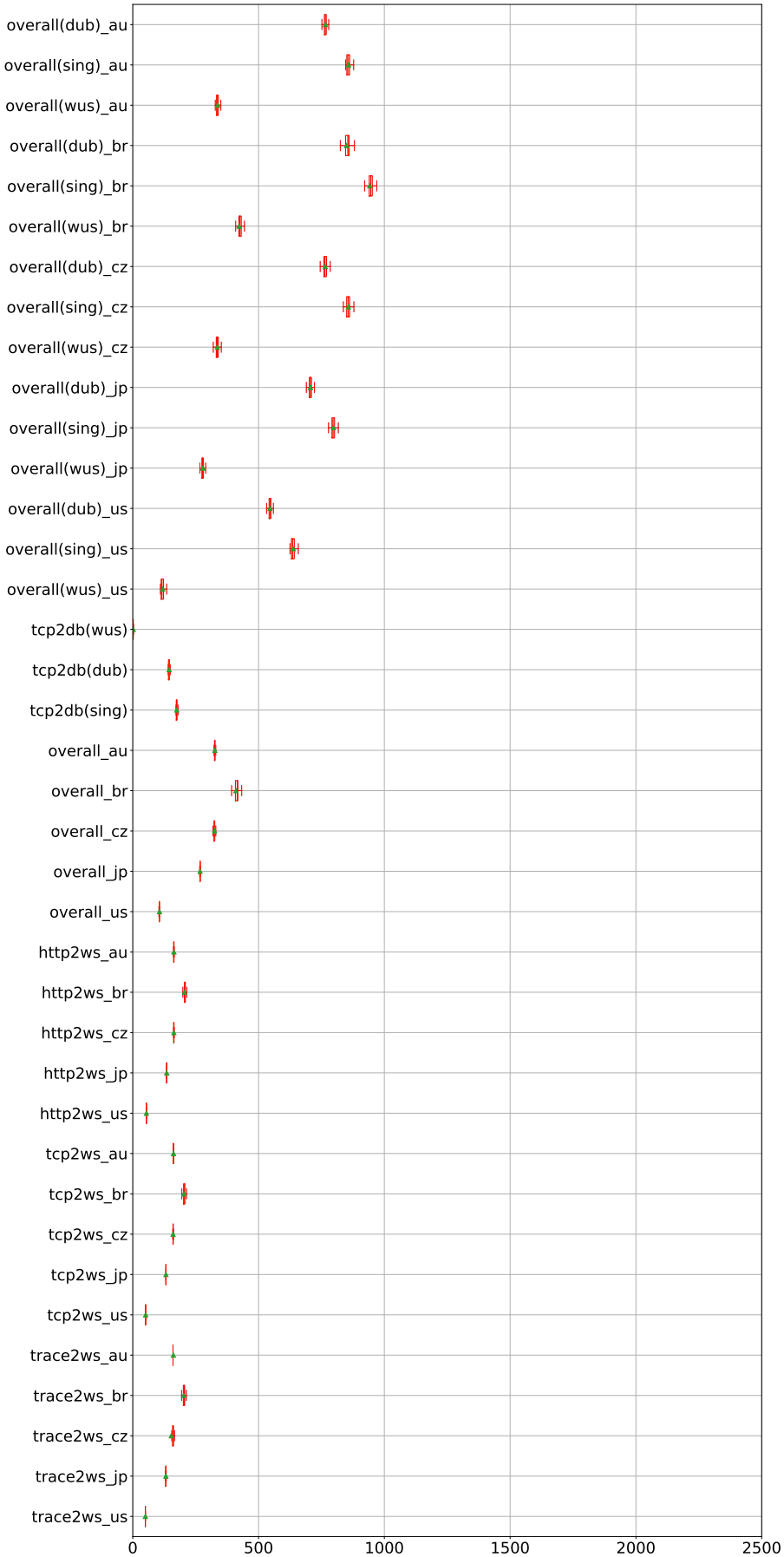


Figure E.40: *CLAudit DBSCAN cluster 0 feature interpretation*

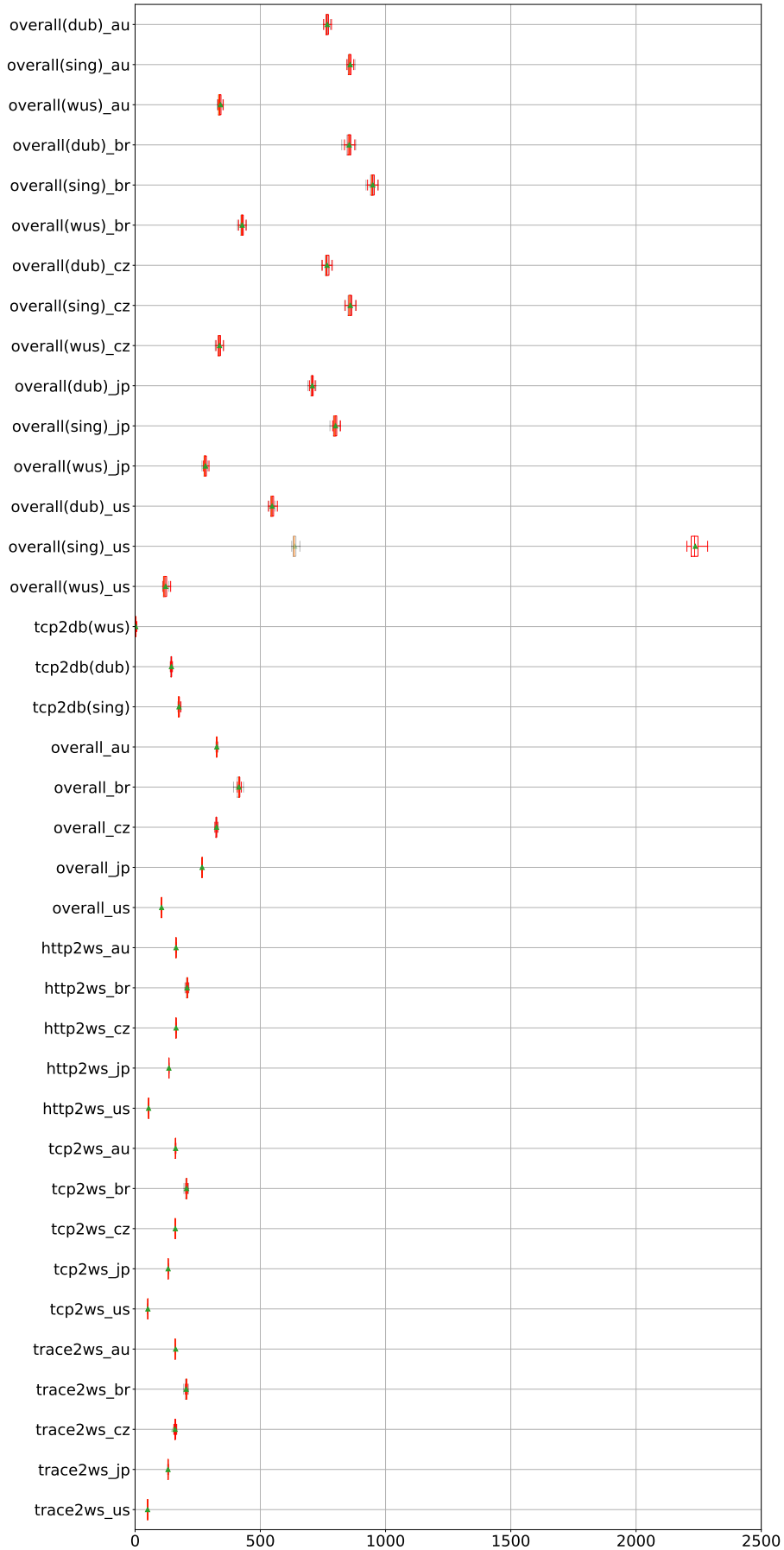


Figure E.41: *CLAudit DBSCAN cluster 1 feature interpretation*

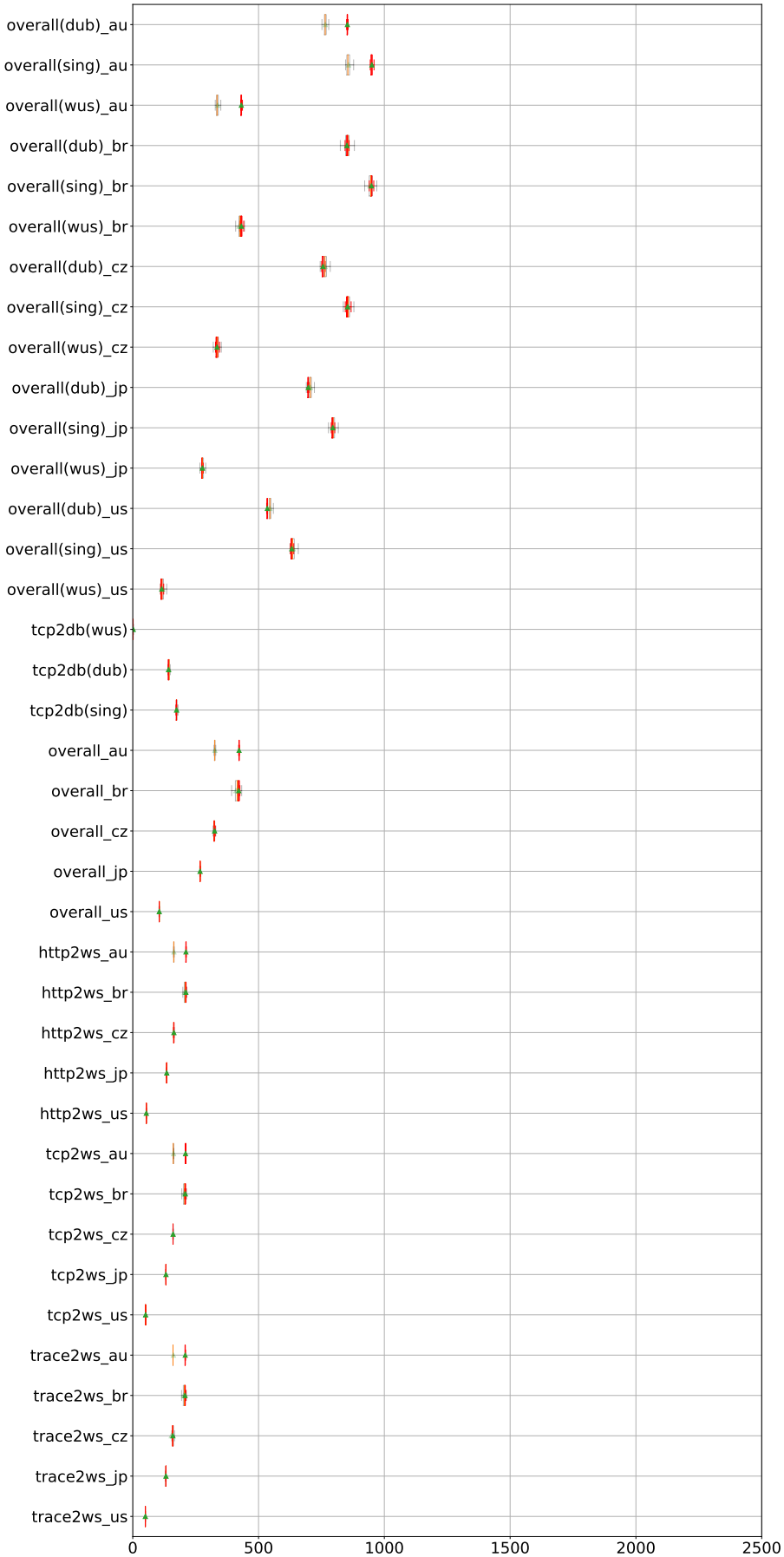


Figure E.42: *CLAudit DBSCAN cluster 2 feature interpretation*

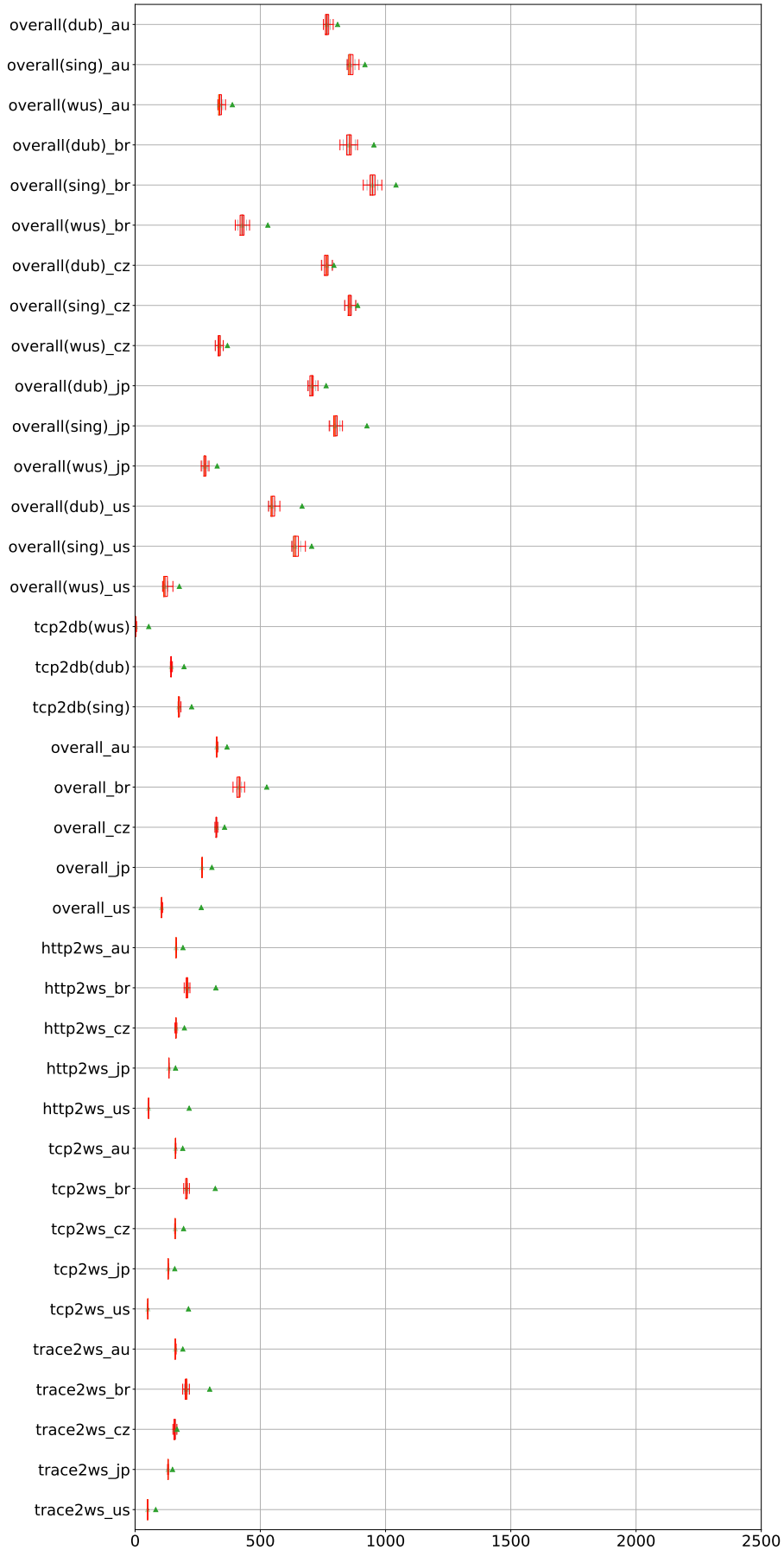


Figure E.43: *CLAudit HDBSCAN cluster -1 feature interpretation*

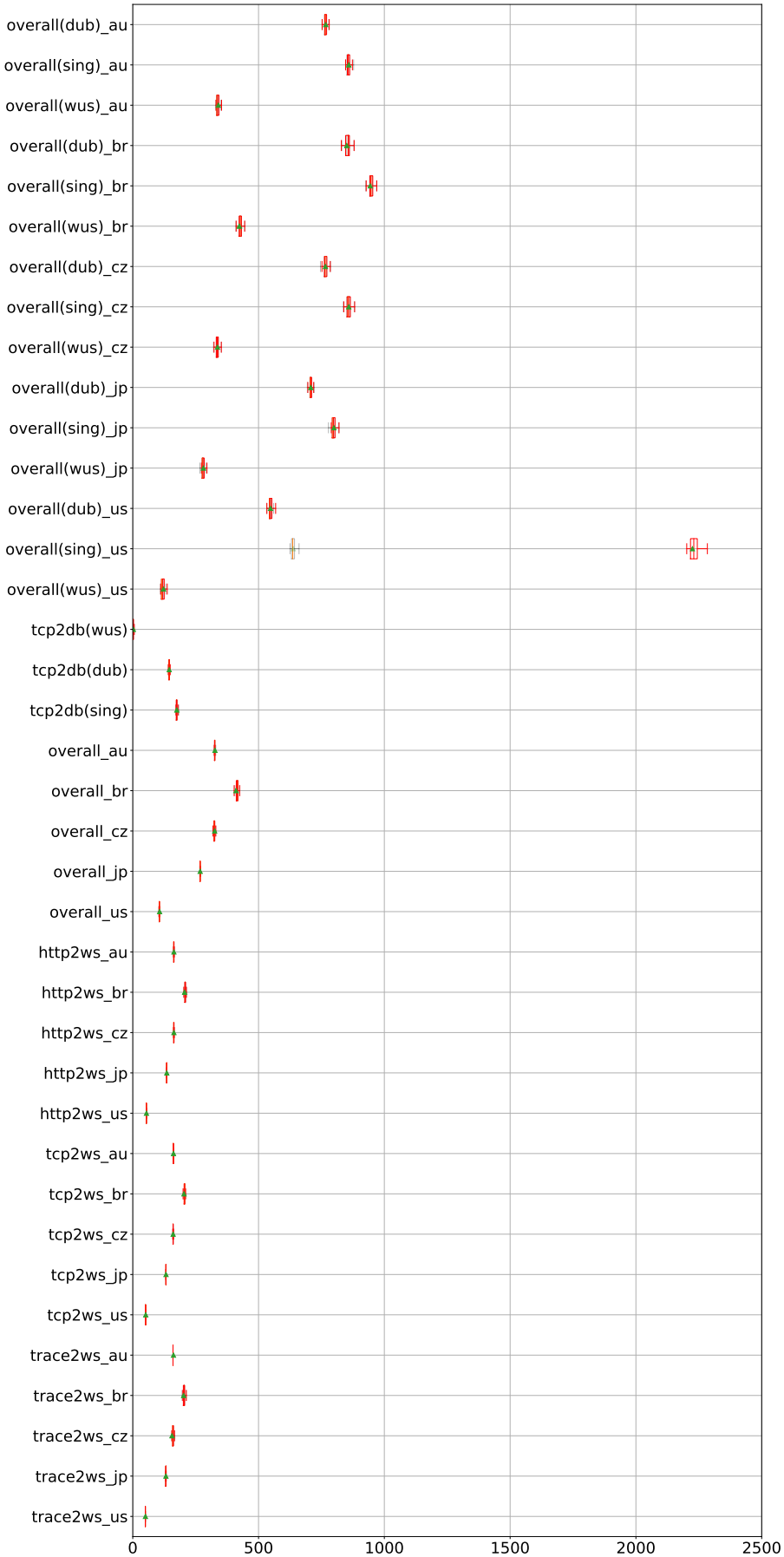


Figure E.44: *CLAudit HDBSCAN cluster 0 feature interpretation*

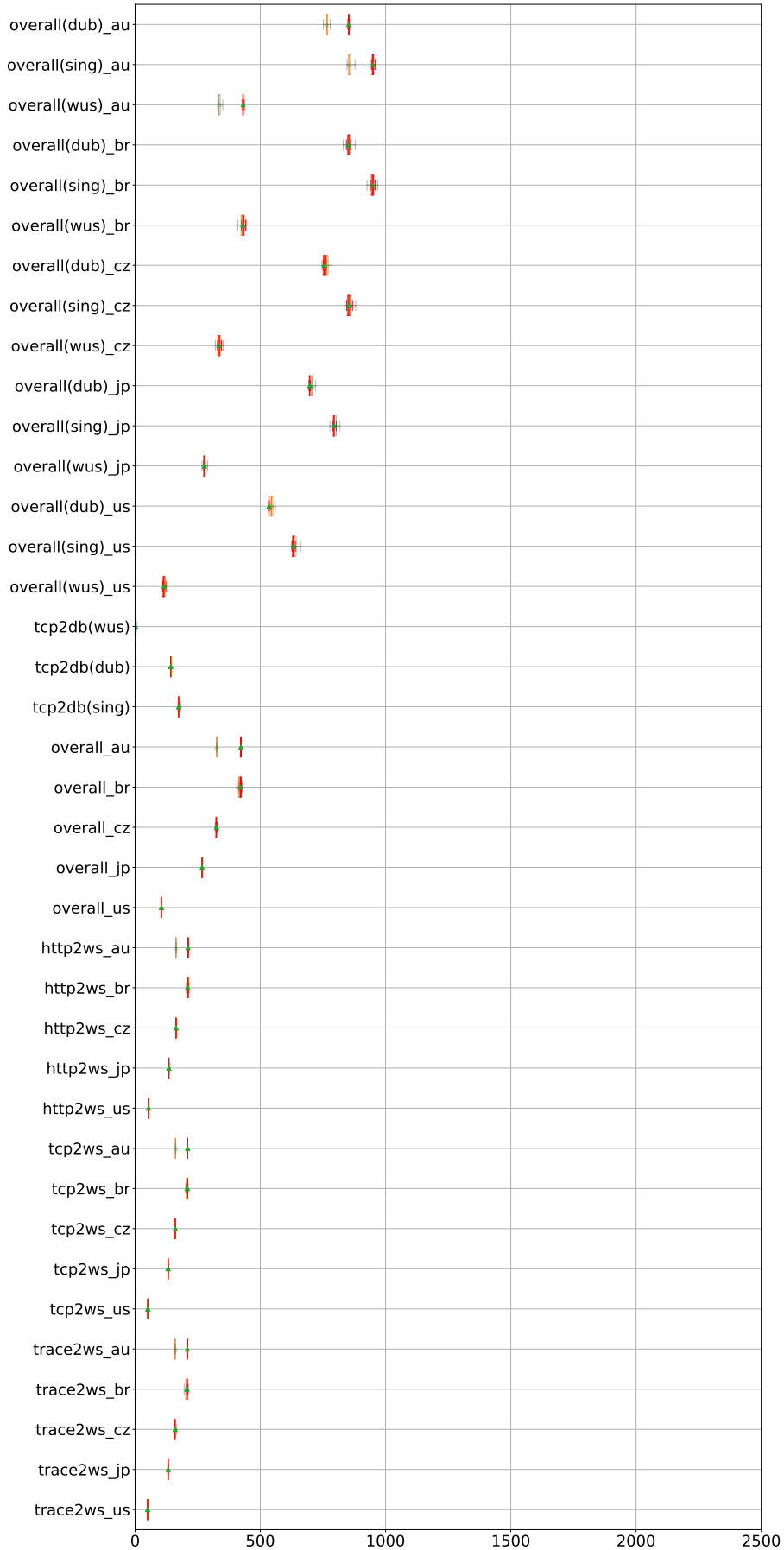


Figure E.45: *CLAudit HDBSCAN* cluster 1 feature interpretation*

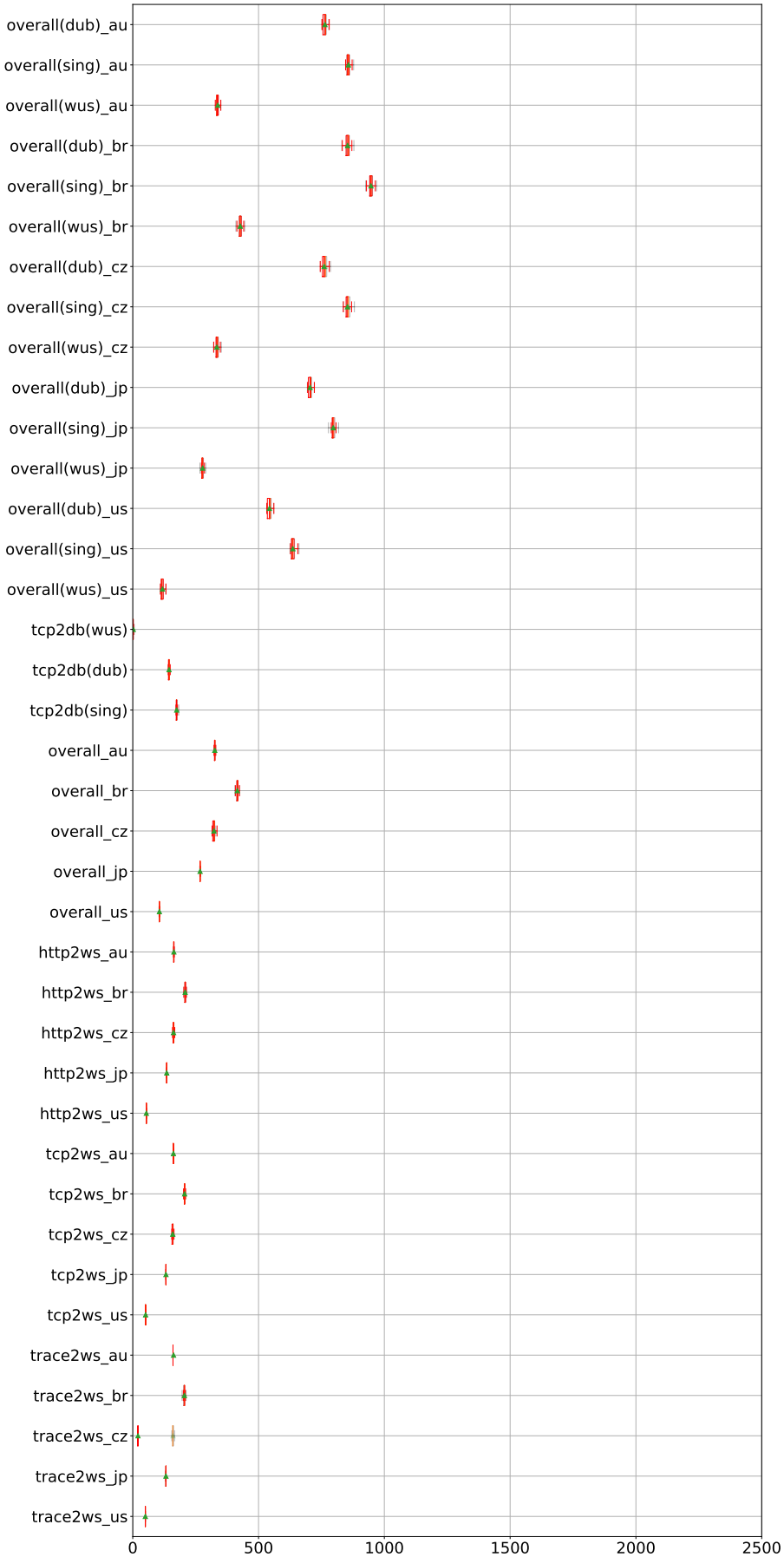


Figure E.46: *CLAudit HDBSCAN* cluster 2 feature interpretation*

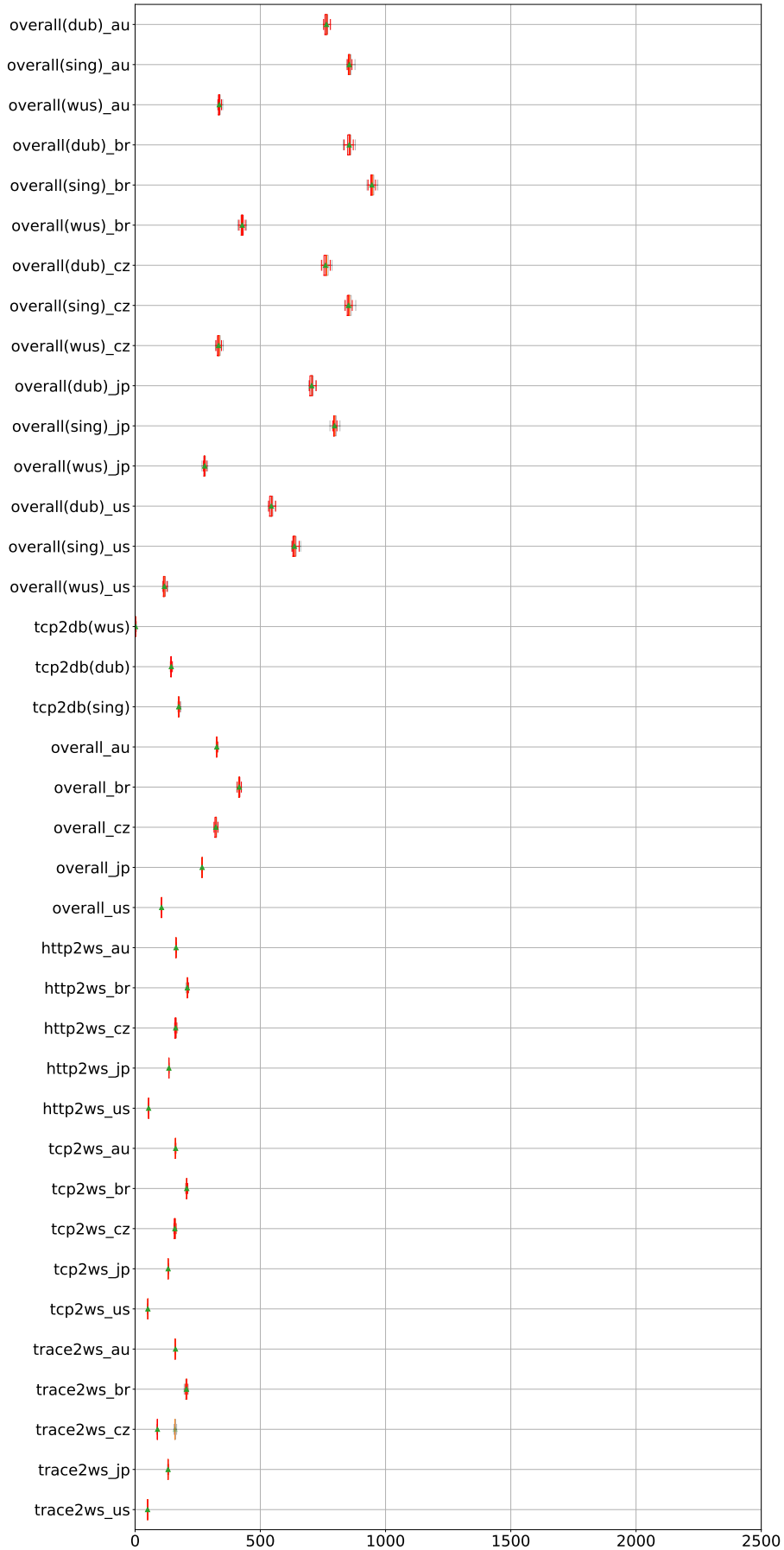


Figure E.47: *CLAudit HDBSCAN* cluster 3 feature interpretation*

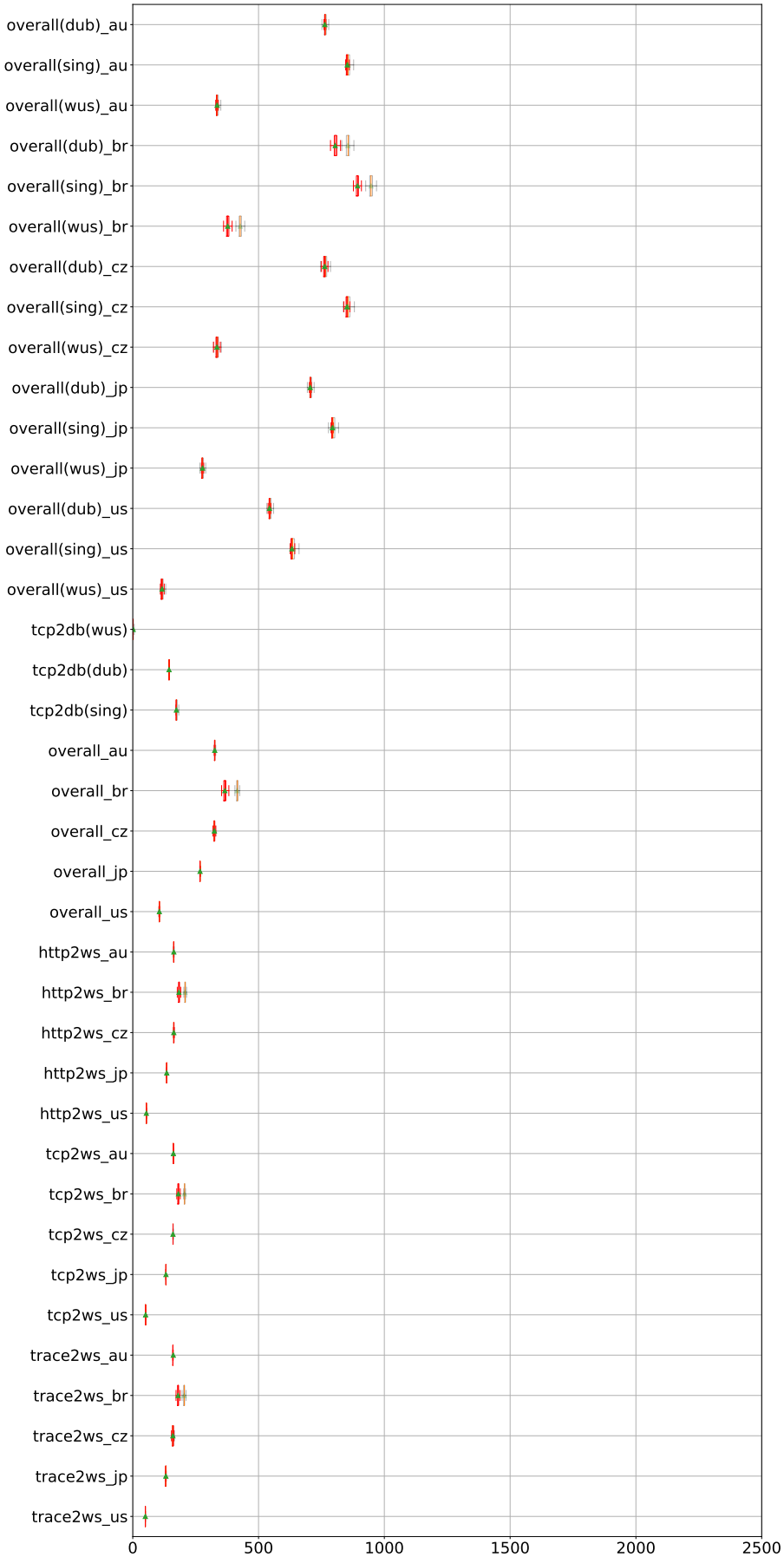


Figure E.48: *CLAudit HDBSCAN* cluster 4 feature interpretation*

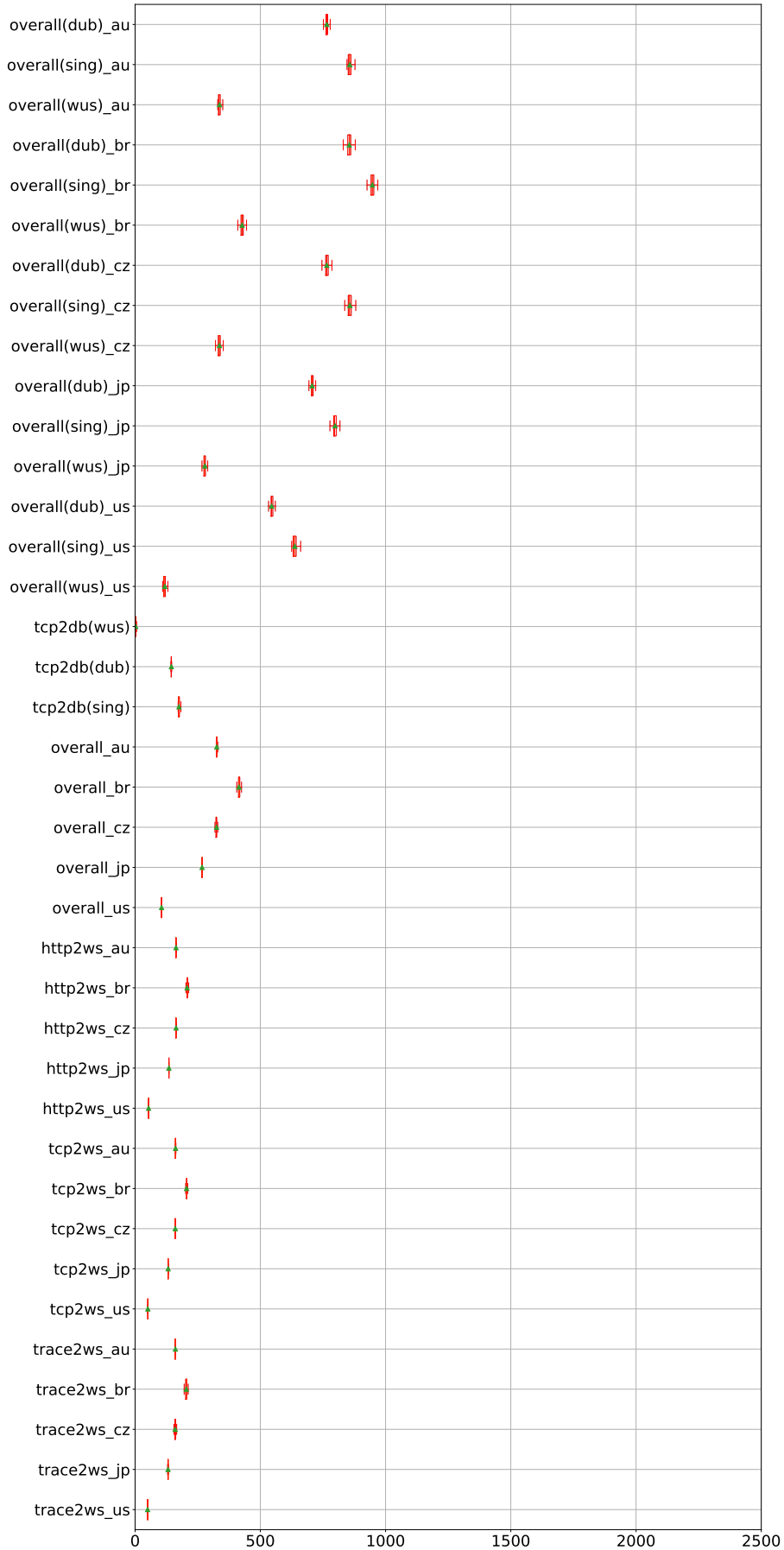


Figure E.49: *CLAudit HDBSCAN* cluster 5 feature interpretation*

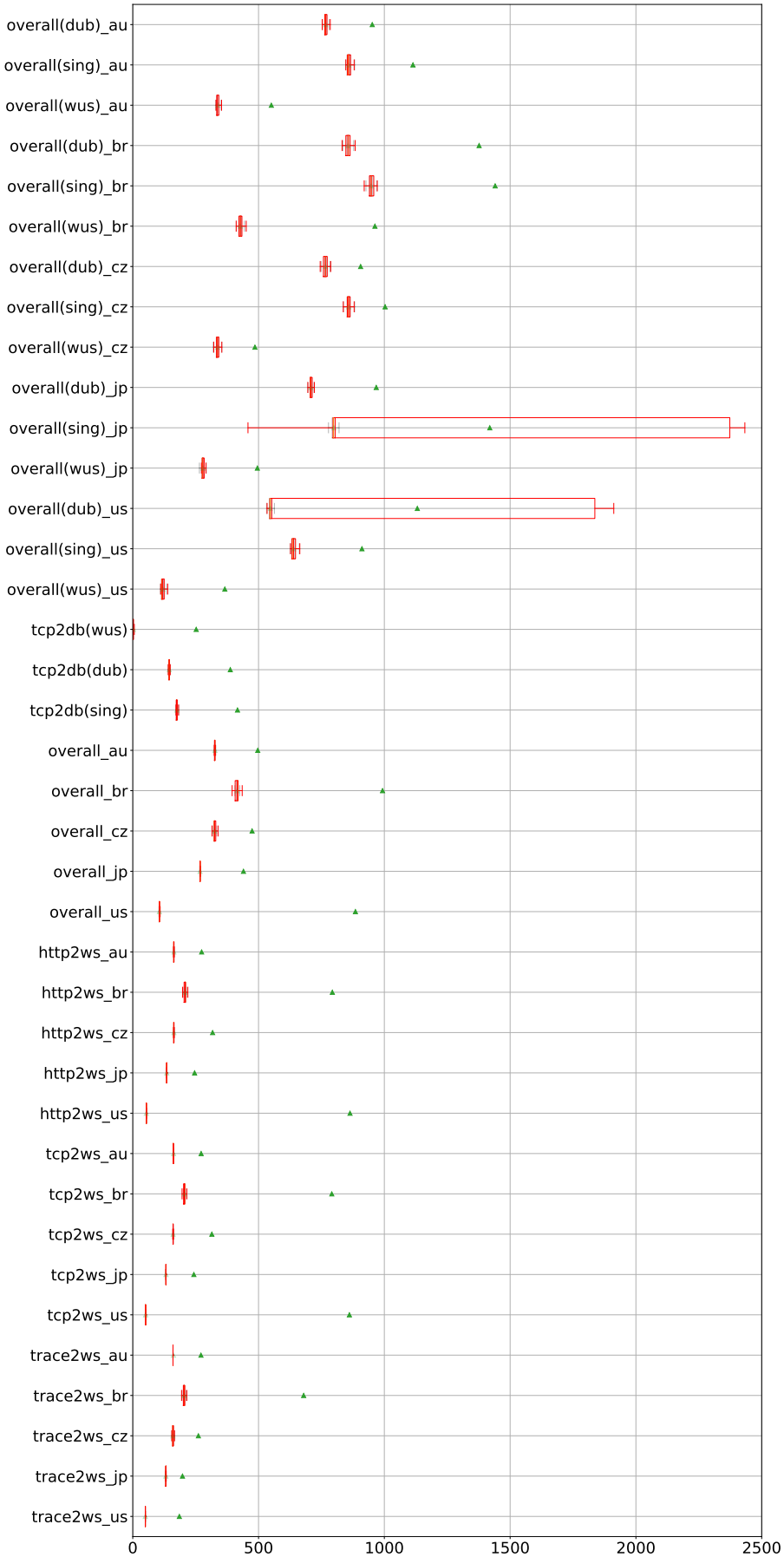


Figure E.50: *CLAudit OPTICS cluster -1 feature interpretation*

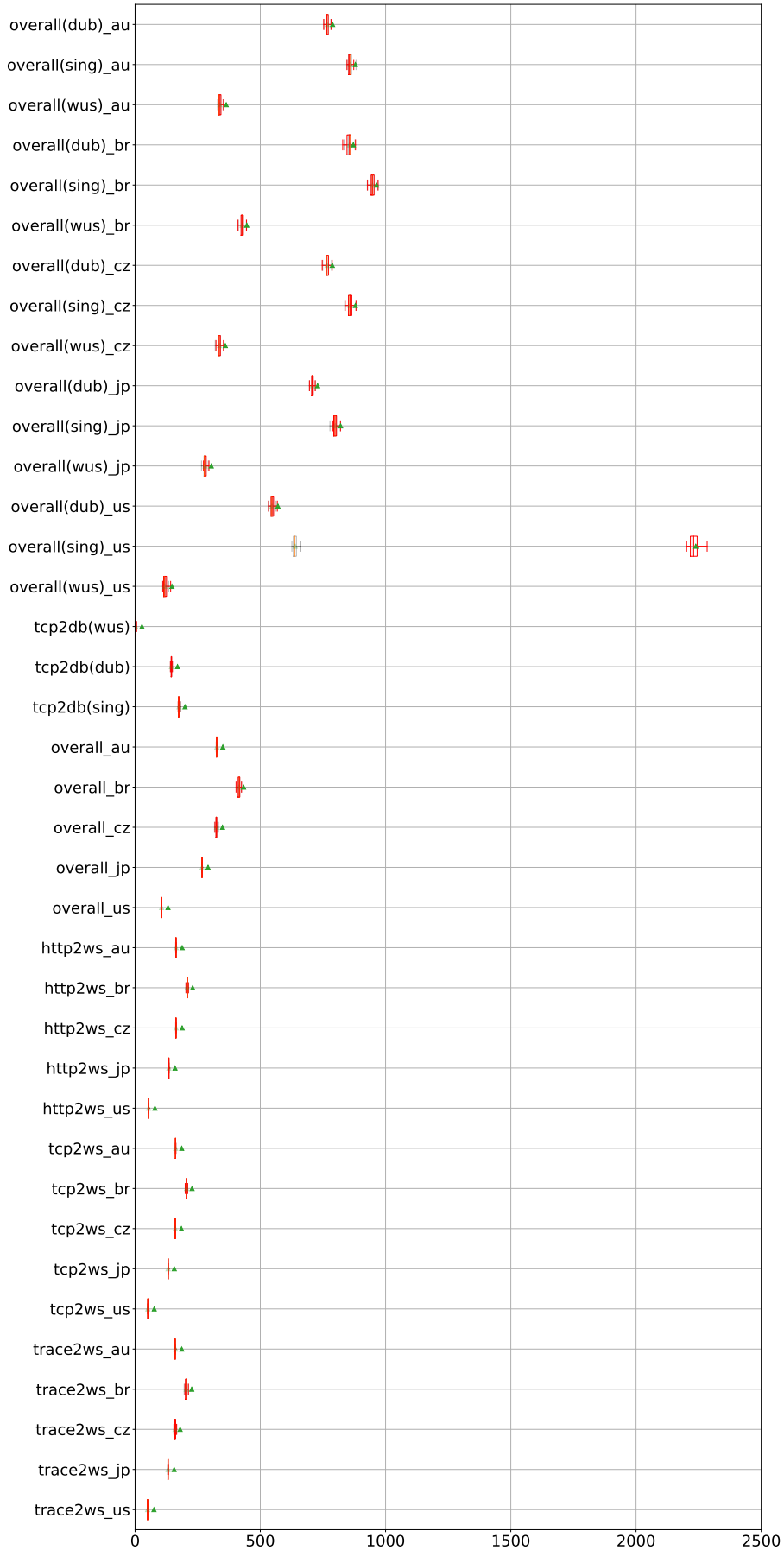


Figure E.51: *CLAudit OPTICS cluster 0 feature interpretation*

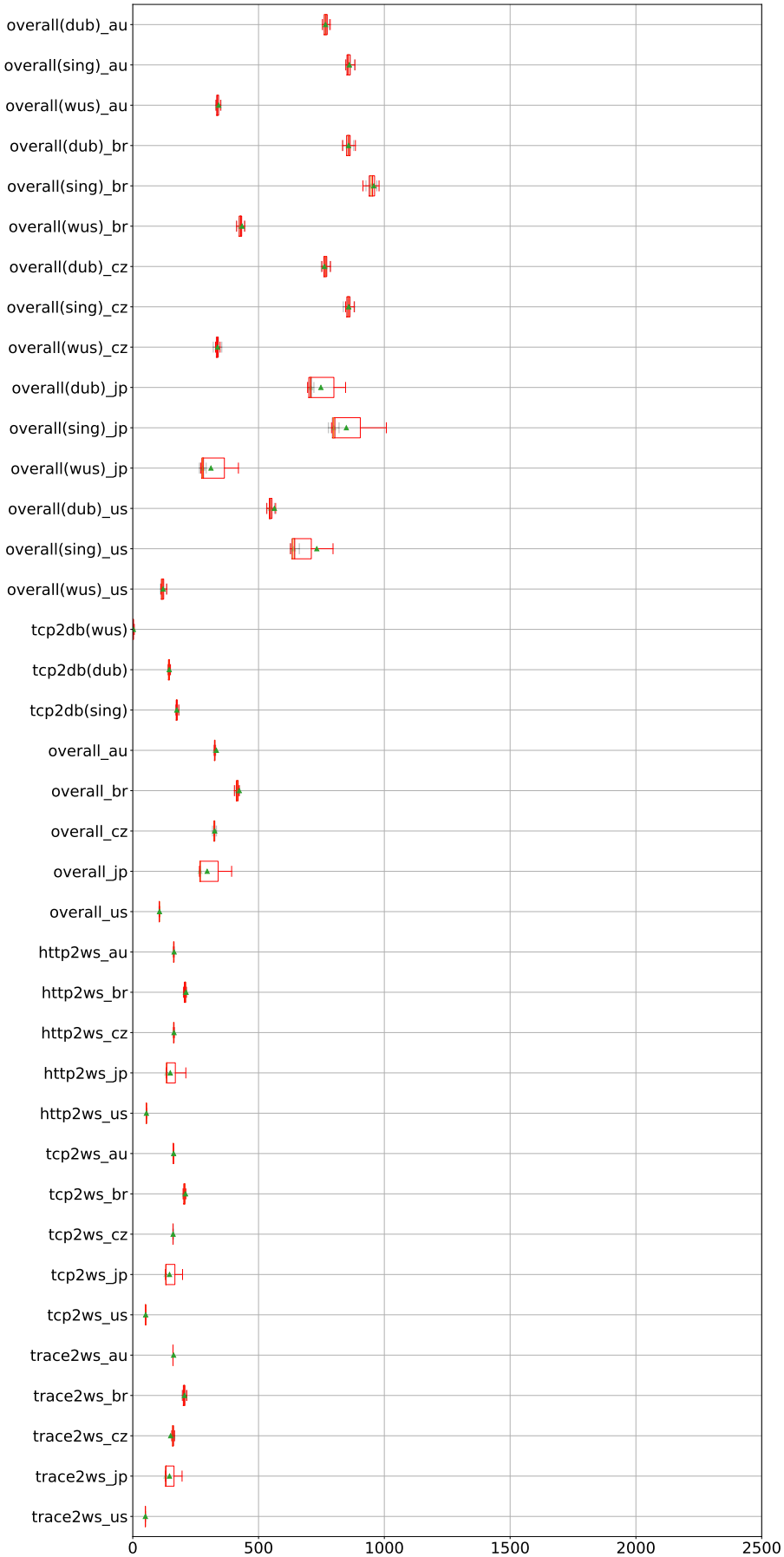


Figure E.52: *CLAudit OPTICS cluster 1 feature interpretation*

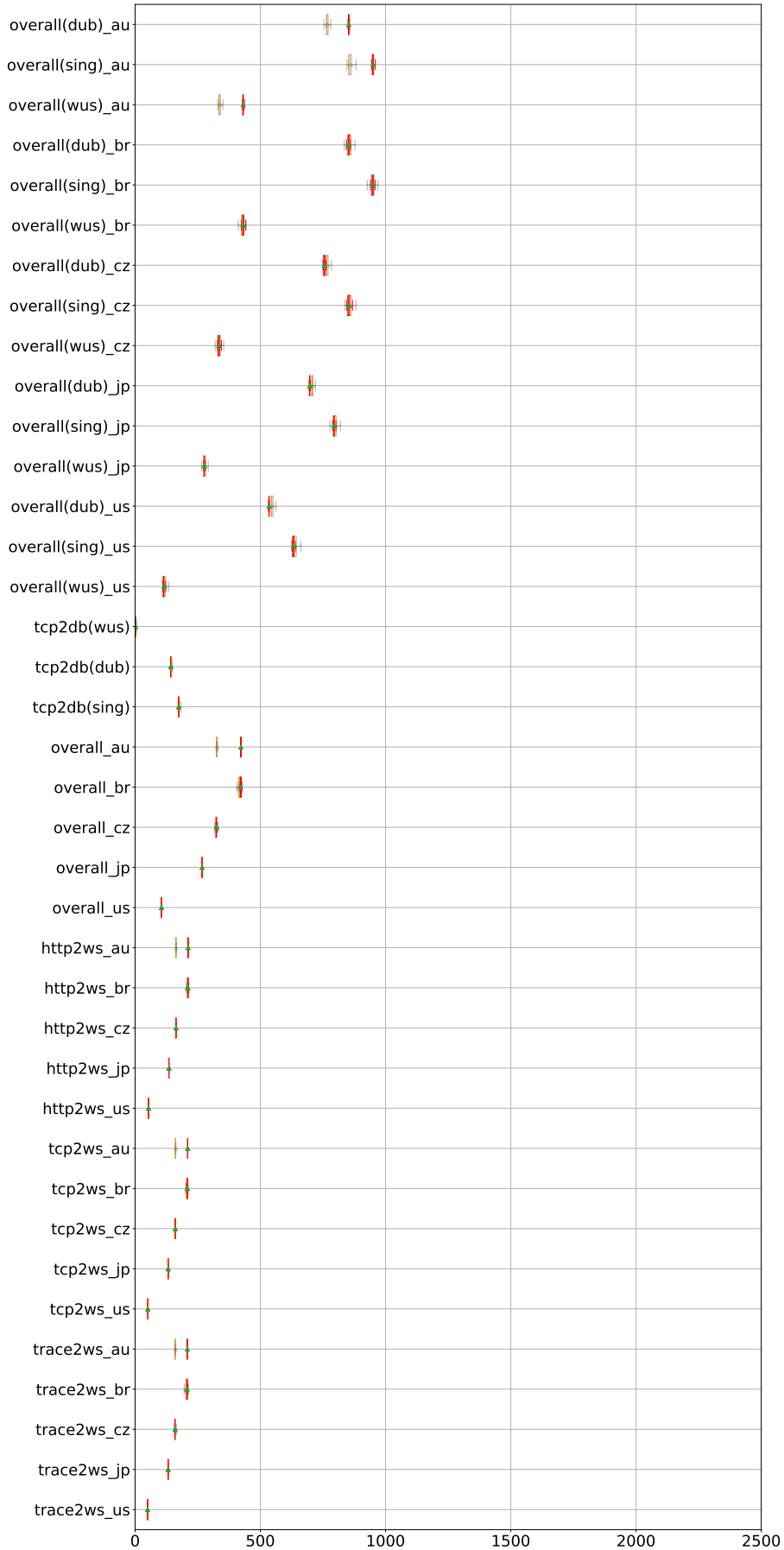


Figure E.53: *CLAudit OPTICS cluster 2 feature interpretation*

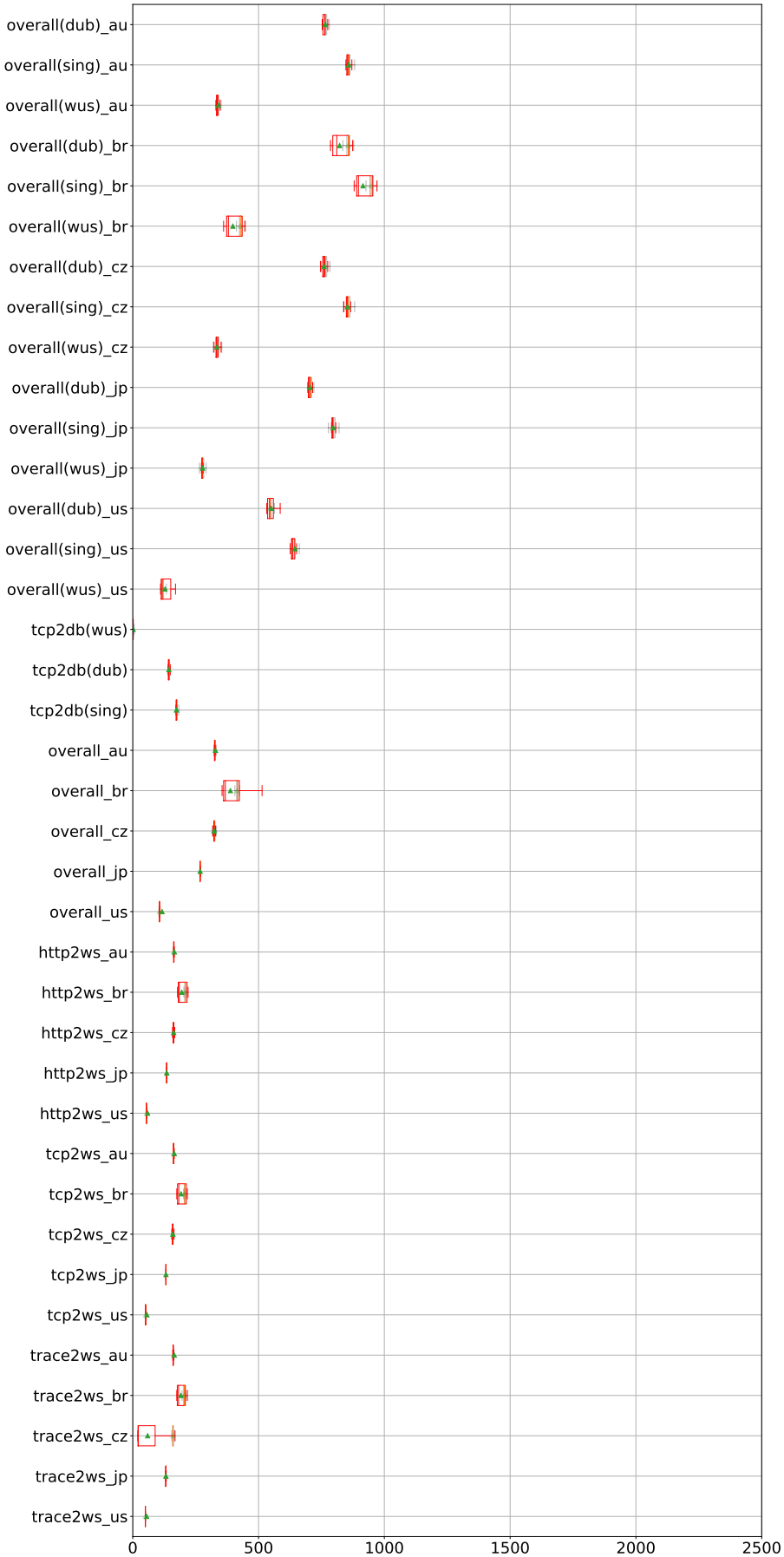


Figure E.54: *CLAudit OPTICS cluster 3 feature interpretation*

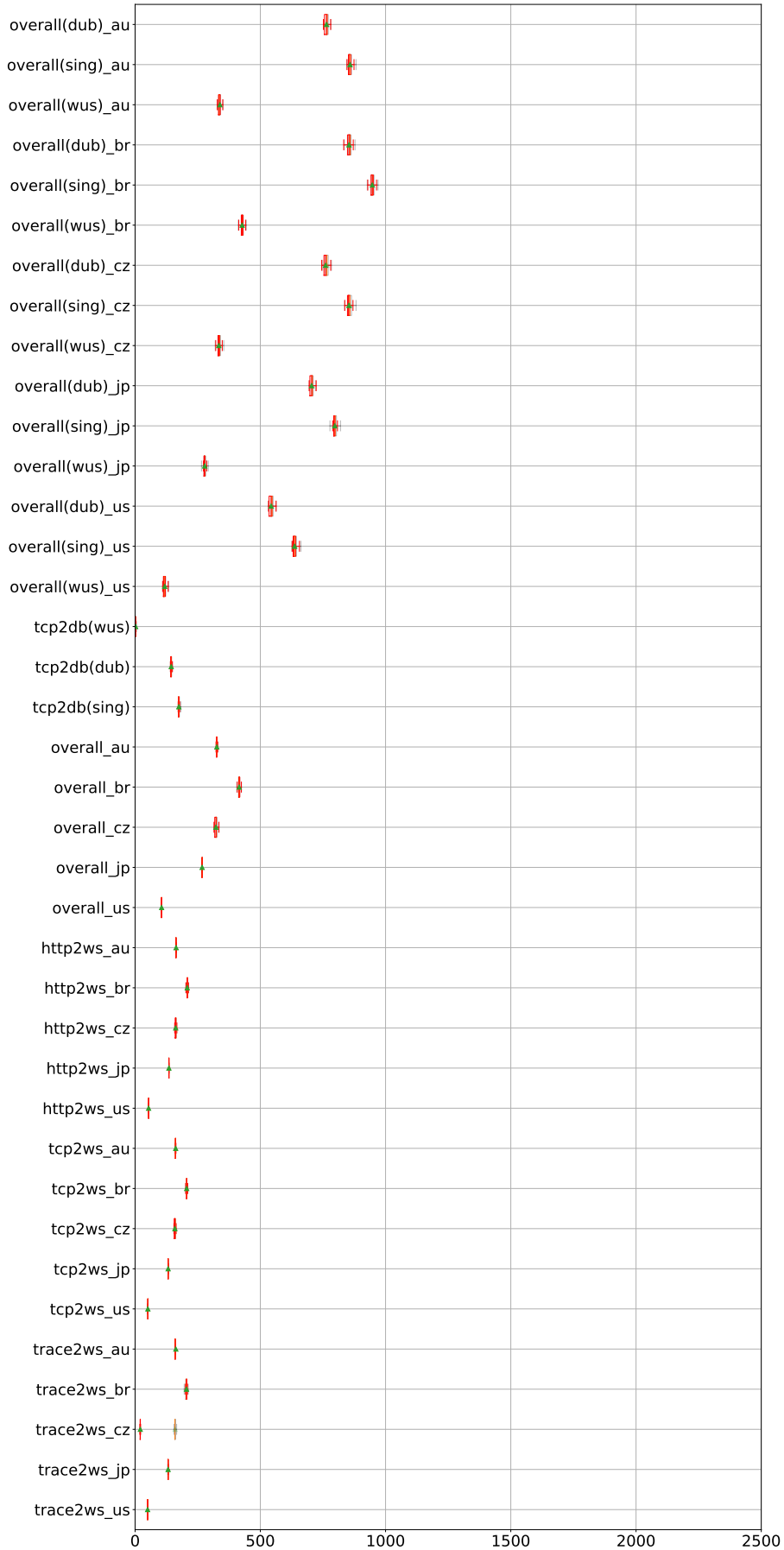


Figure E.55: *CLAudit OPTICS cluster 4 feature interpretation*

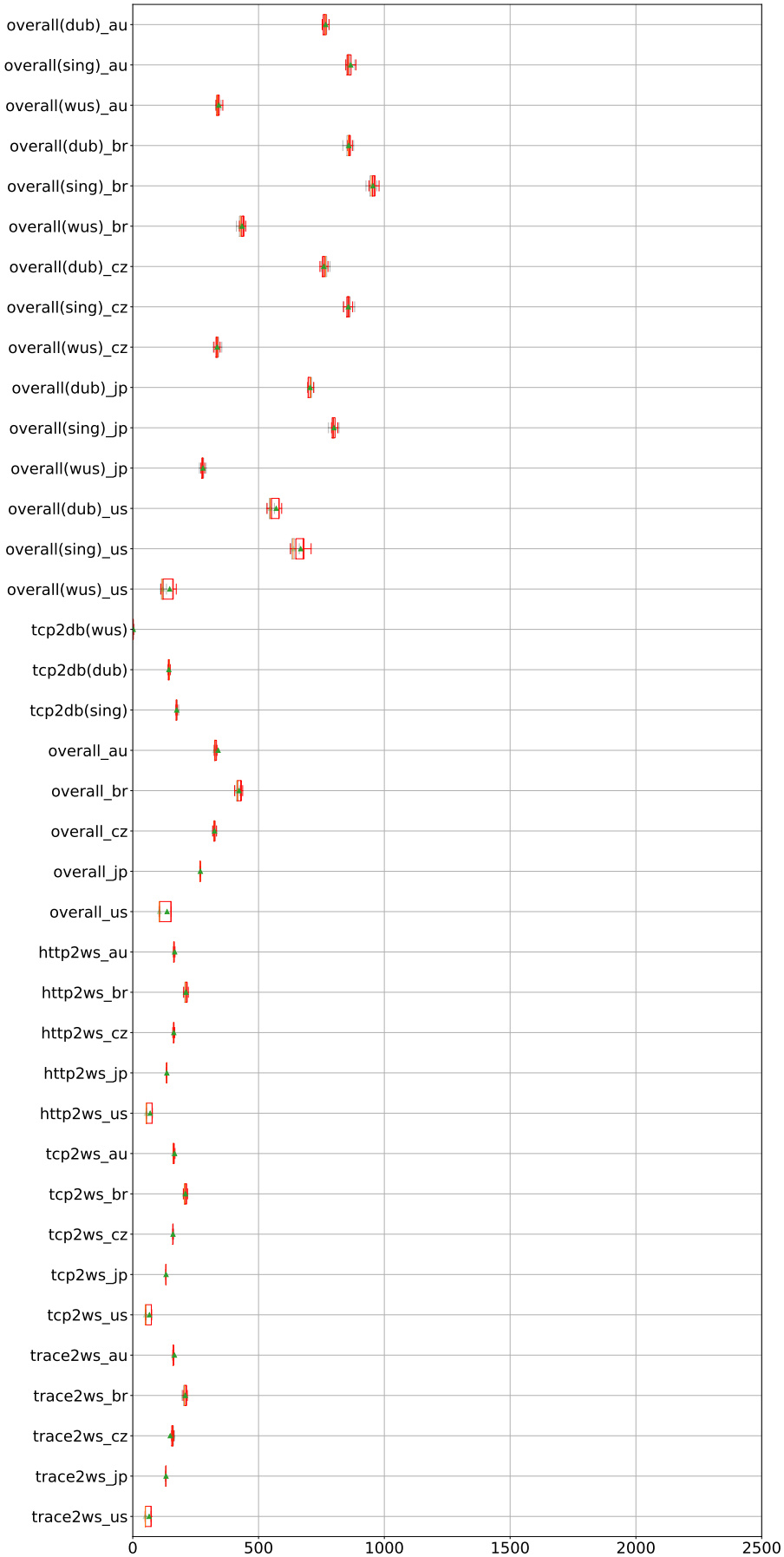


Figure E.56: *CLAudit OPTICS cluster 5 feature interpretation*

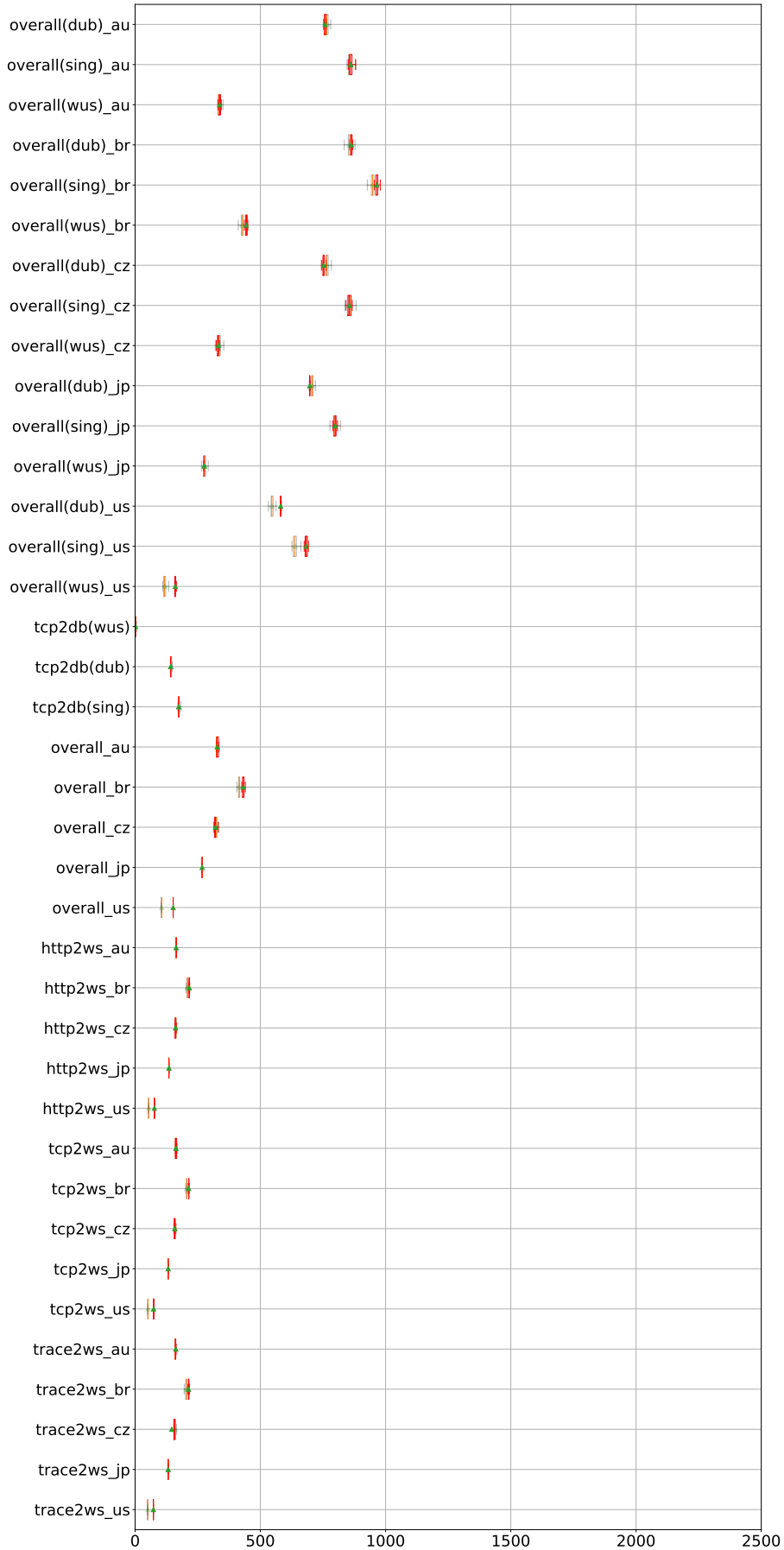


Figure E.57: *CLAudit OPTICS cluster 6 feature interpretation*

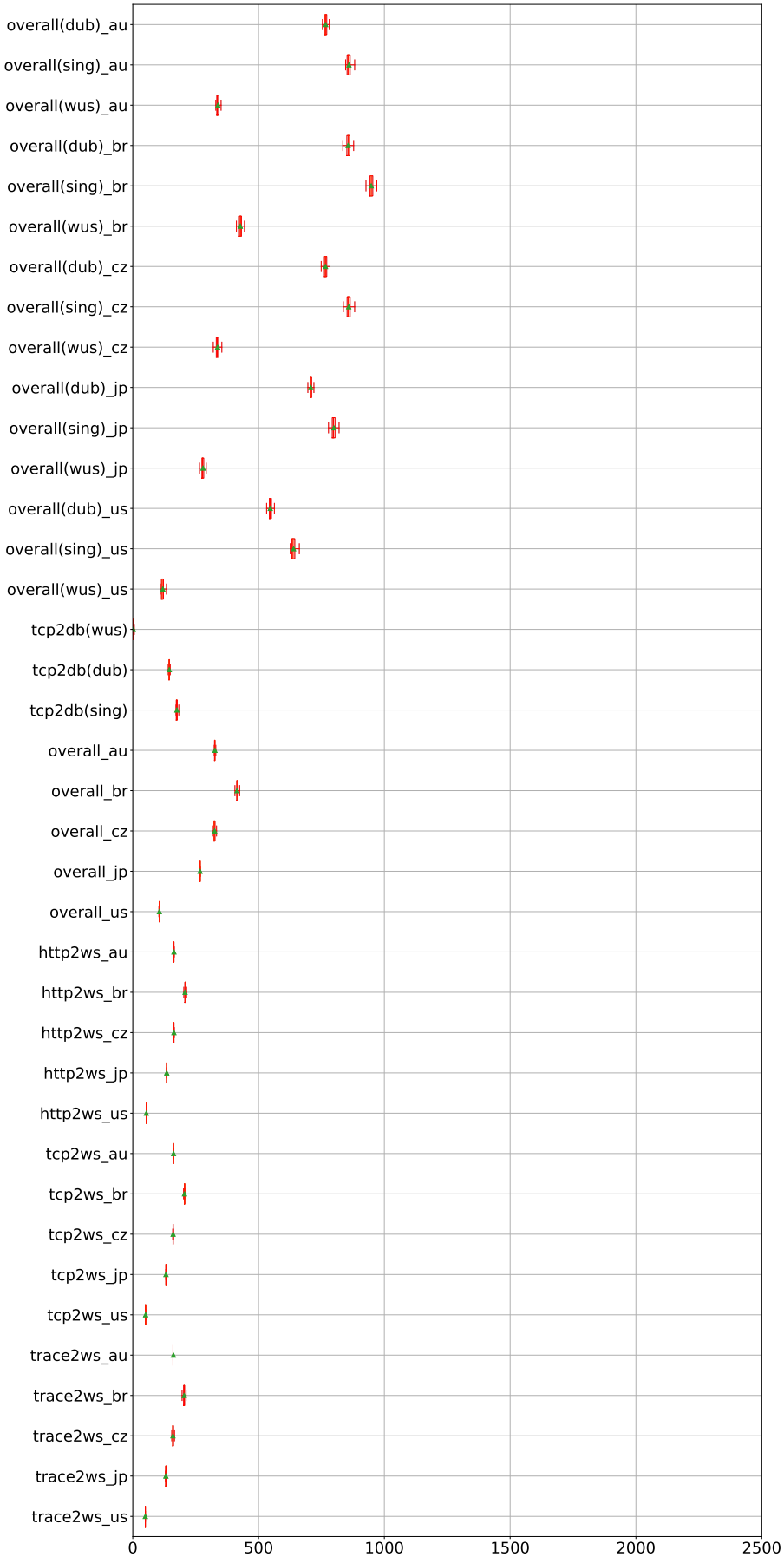


Figure E.58: *CLAudit OPTICS cluster 7 feature interpretation*

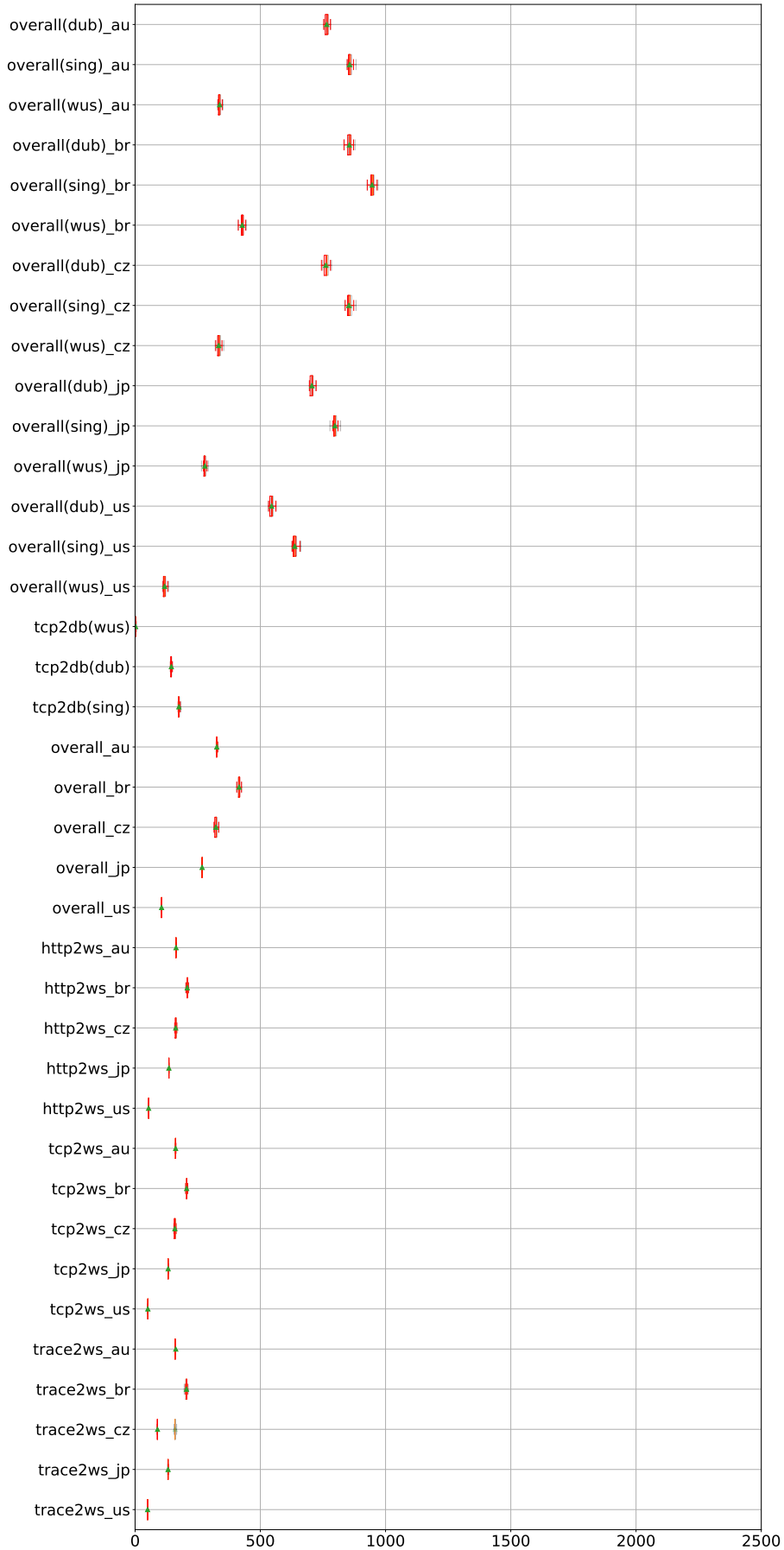


Figure E.59: *CLAudit OPTICS cluster 8 feature interpretation*

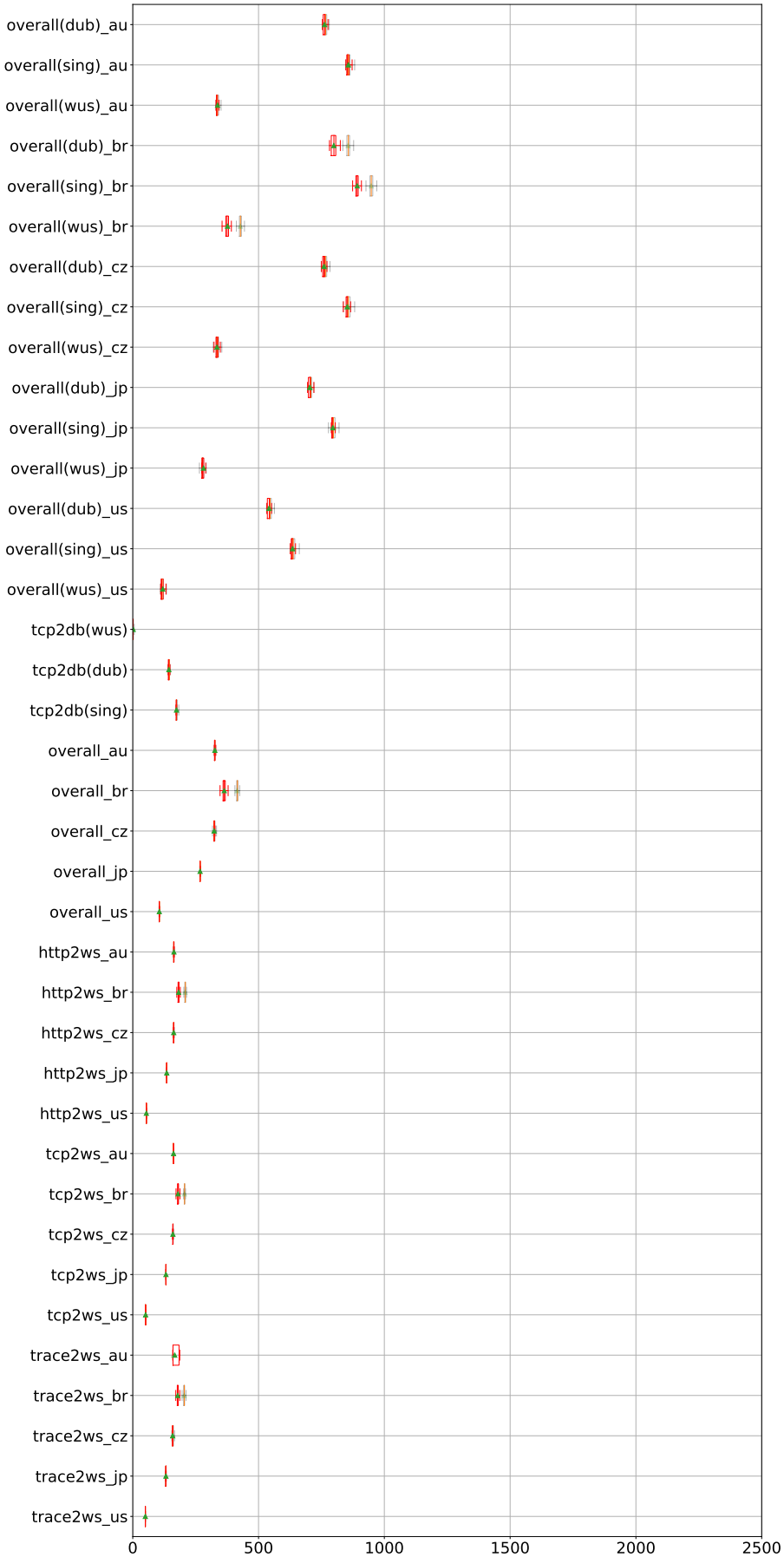


Figure E.60: *CLAudit OPTICS cluster 9 feature interpretation*

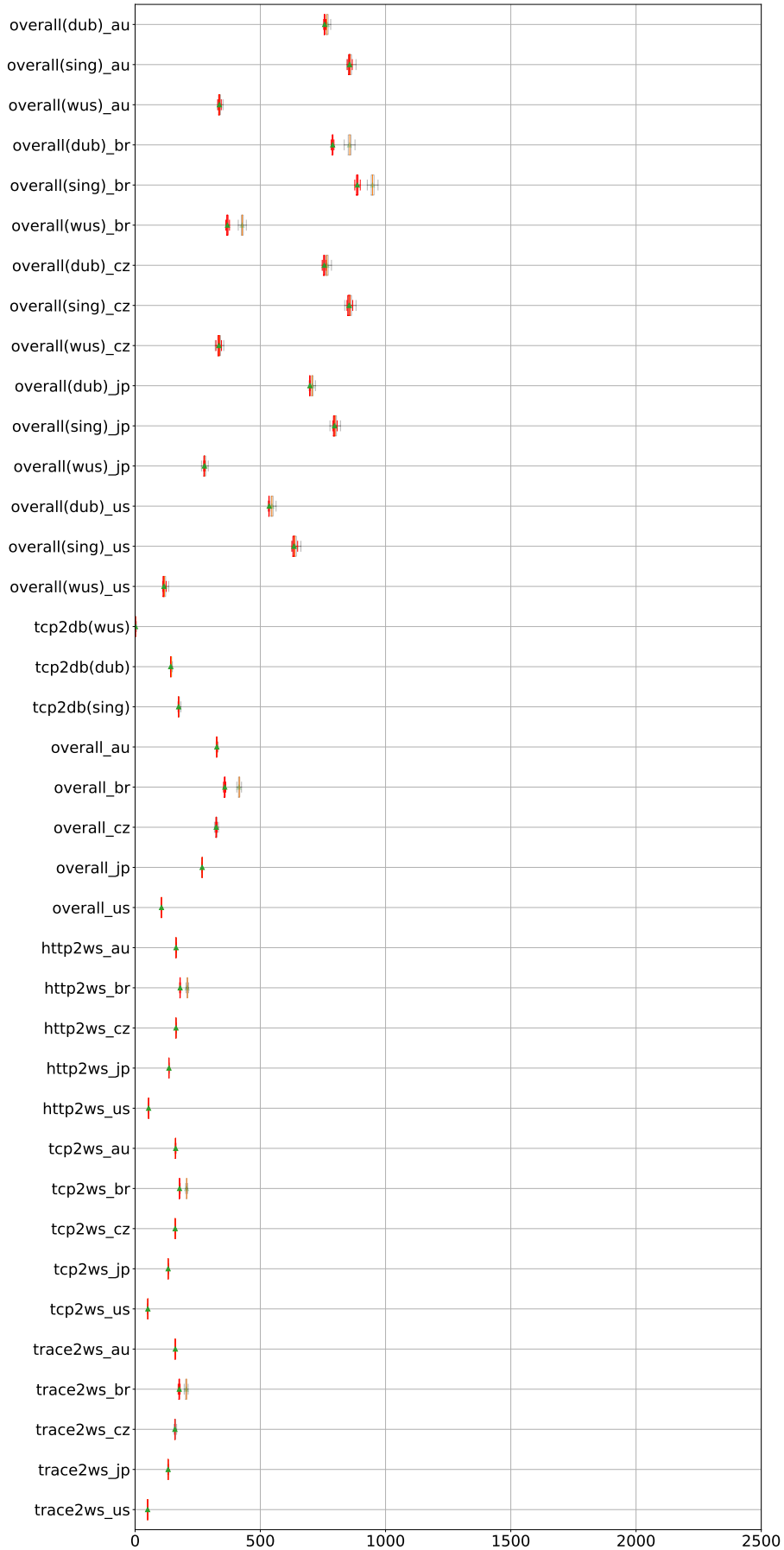


Figure E.61: *CLAudit OPTICS cluster 10 feature interpretation*

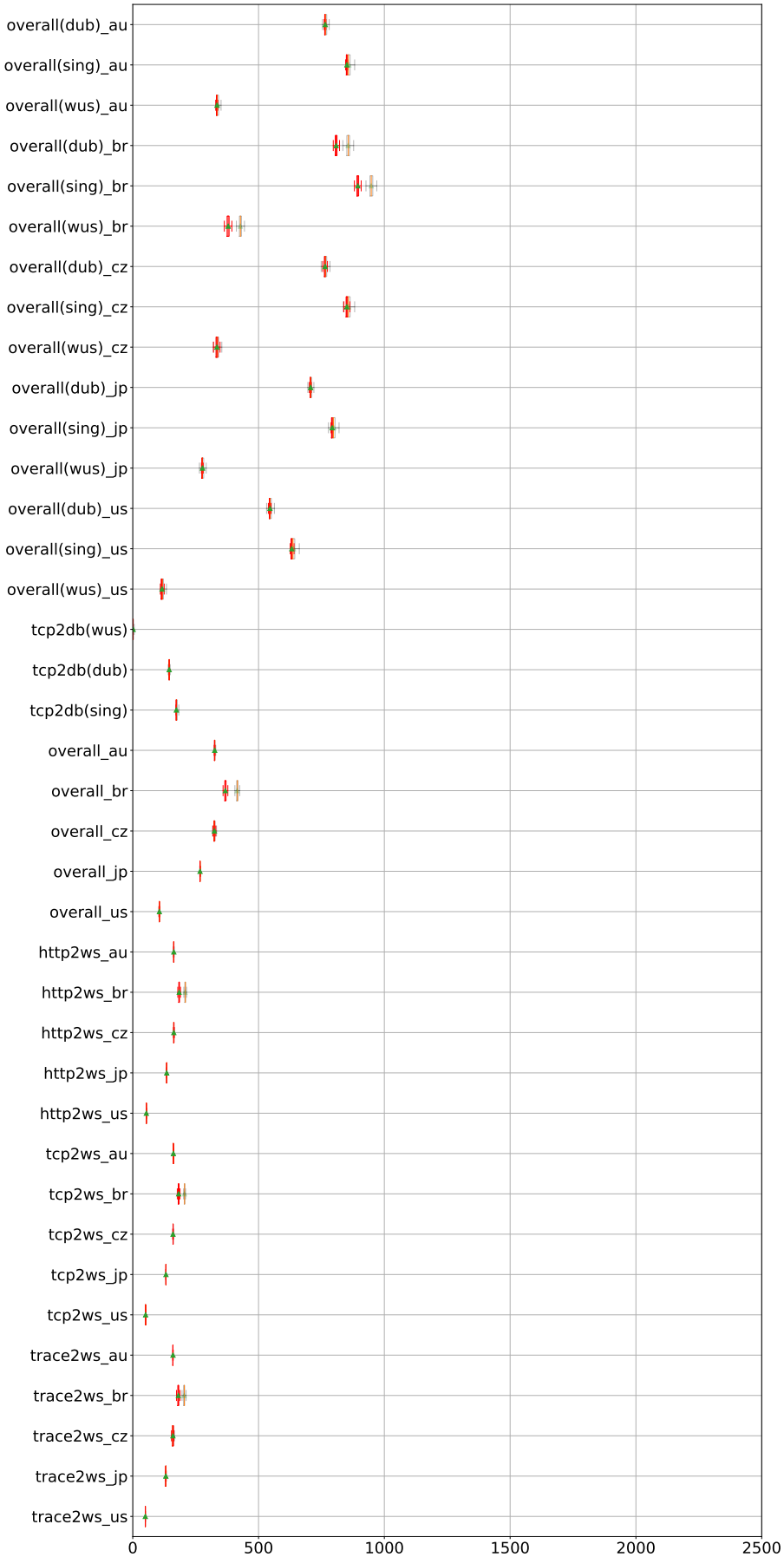


Figure E.62: *CLAudit OPTICS cluster 11 feature interpretation*

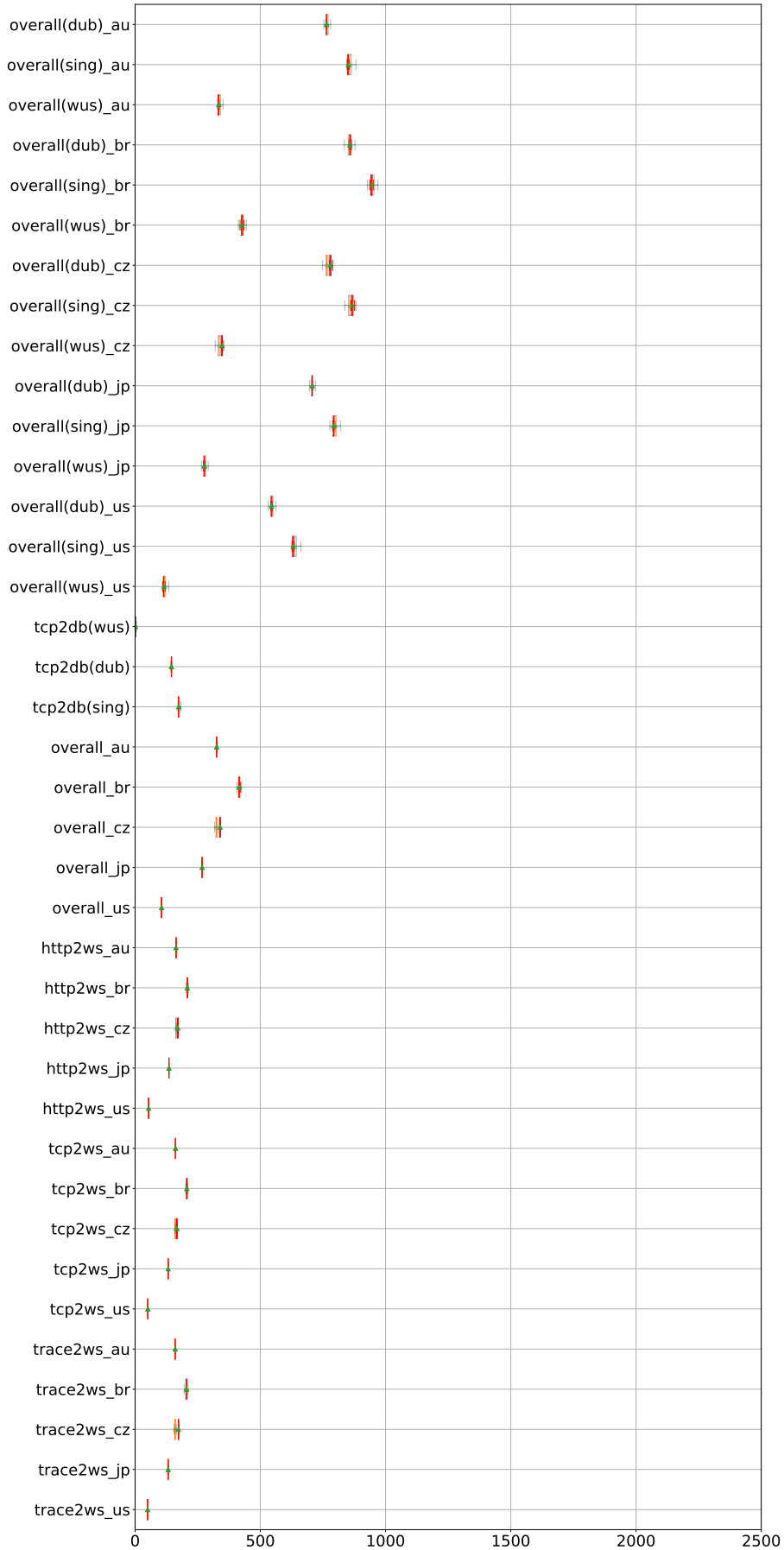


Figure E.63: *CLAudit OPTICS cluster 12 feature interpretation*

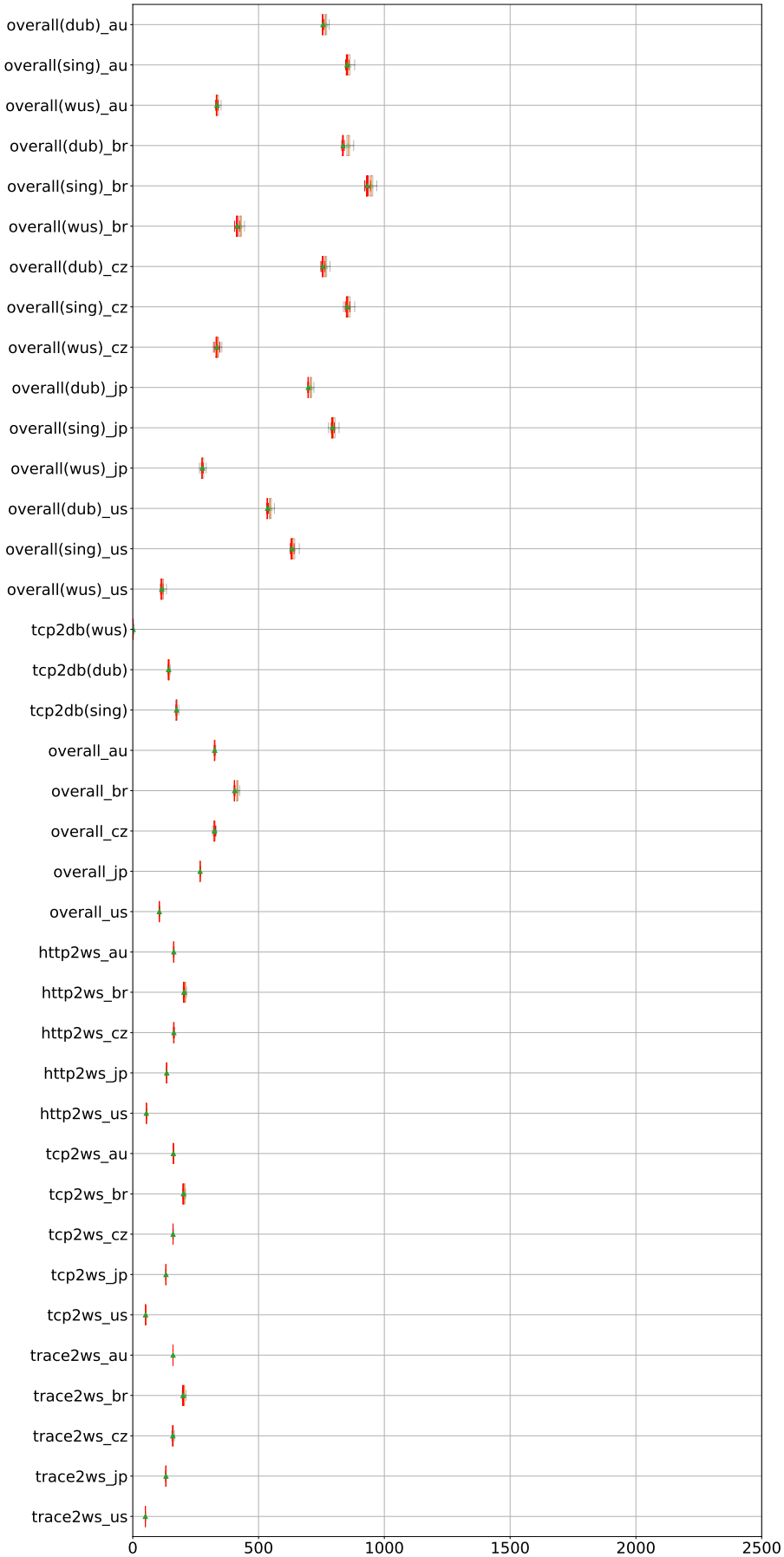


Figure E.64: *CLAudit OPTICS cluster 13 feature interpretation*

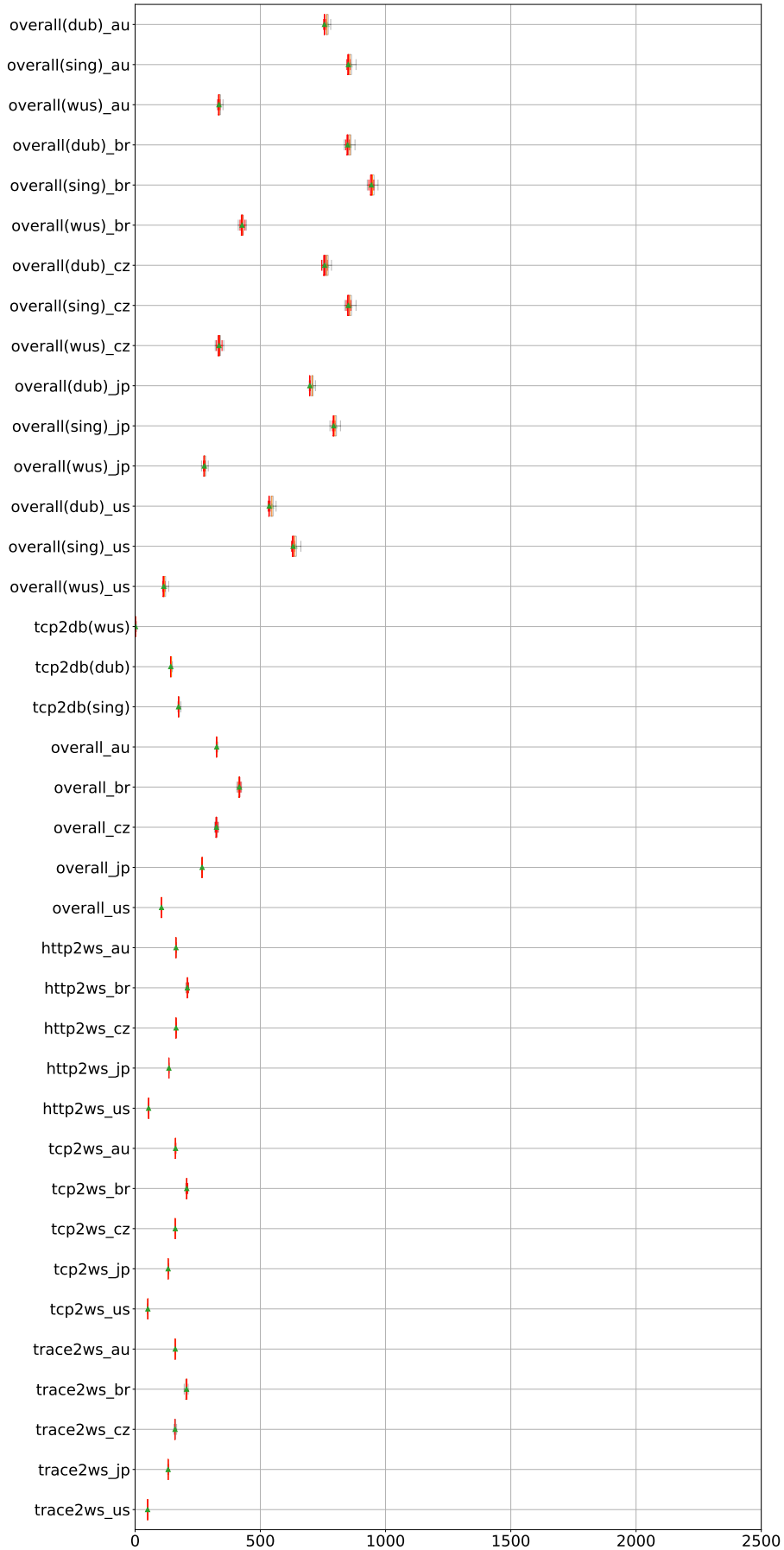


Figure E.65: *CLAudit OPTICS cluster 14 feature interpretation*

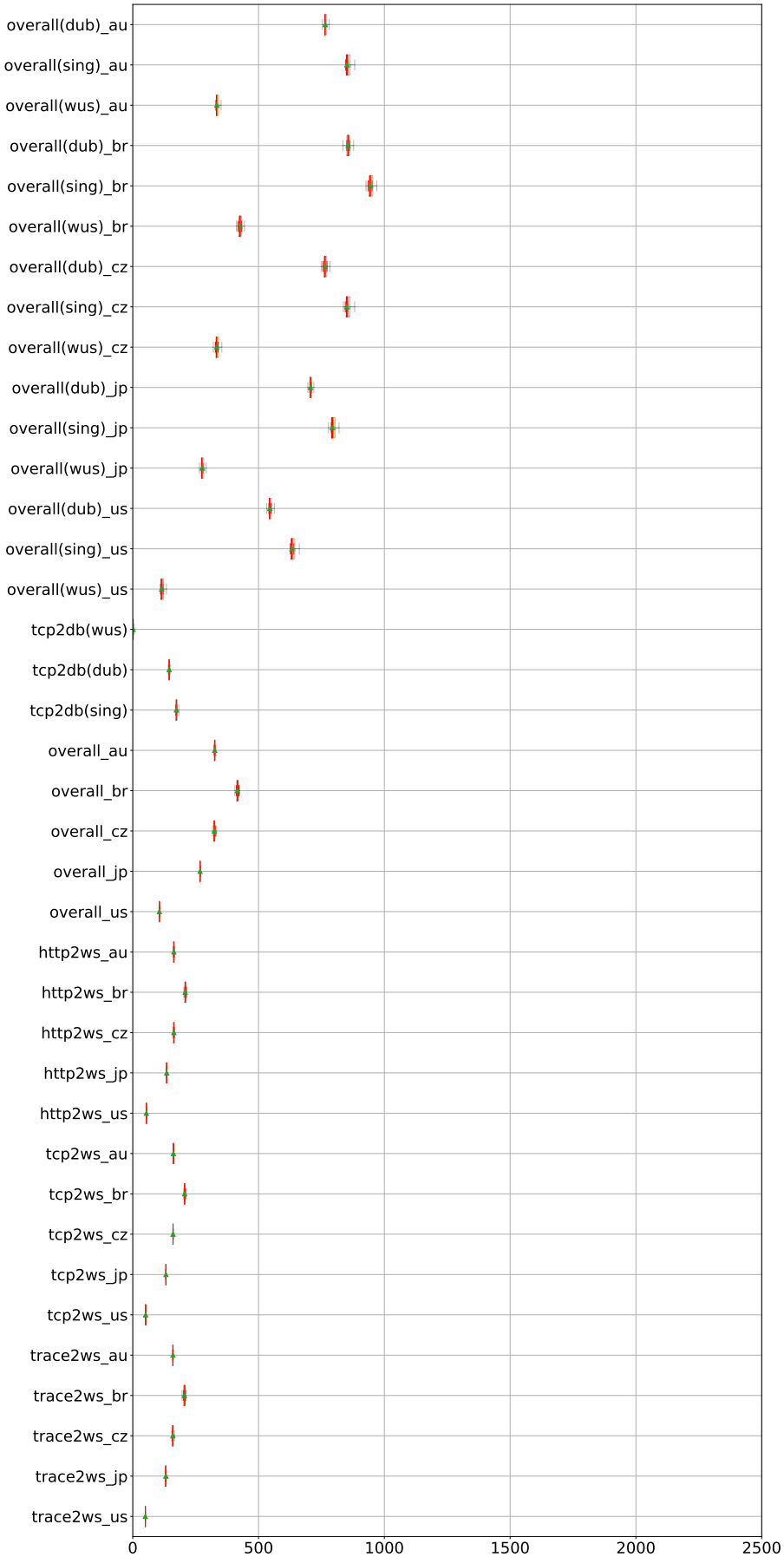


Figure E.66: *CLAudit OPTICS cluster 15 feature interpretation*