Understandable Log-Based Anomaly Localisation through Inter-Host Distances

Mahira Ali

Delft University of Technology

TU Delft

Understandable Log-Based Anomaly Localisation through Inter-Host Distances

by

Mahira Ali

to obtain the degree of Master of Science in Computer Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 27, 2024 at 11:00 AM

Student Number:	4940857	
Project Duration:	November 21, 2023 – August 27, 2024	
Supervisor:	Dr. ir. S. E. Verwer	
Thesis Committee:	Dr. ir. E. F. Aivaloglou	EEMCS, TU Delft

An electronic version of this thesis is available at https://repository.tudelft.nl/

TU Delft

Disclaimer

In the preparation of this document, various artificial intelligence tools, including but not limited to ChatGPT and Bing AI, have been utilised to enhance the clarity, coherence, and quality of the textual content. These tools assisted in the refinement of language and expression. However, they were not used to generate any research, data, or results presented herein. All research, analysis, and findings are original and the sole responsibility of the author. The final responsibility for the accuracy, validity, and scholarly integrity of the material remains with the author.

Abstract

While artificial intelligence (AI) has undeniably ushered numerous solutions across various fields, the growing belief that AI can solve all problems overshadows their lack of transparency that comes along. Understanding how decisions are made and what has led to the output is crucial in critical systems to ensure accountability and trust.

This research proposes a complementary method leveraging inter-host distances that localise the outlying hosts, logs and the time frames, which require more advanced analysis. By relying on a variant of a prominent statistical method in the field of authorship attribution - Burrows Delta - the approach enhances transparency in identifying deviating hosts, logs and time frames. Hence, the proposed solution offers an understandable complementary method that preserves integrability by being a log-based method while enabling understandable pinpointing of the specific hosts, logs and time frames that warrant further advanced analysis. By providing insights into the behaviour of the hosts over time, a temporal summarisation for security analysts is provided, relaxing their need to go through all the log files to understand the hosts' behaviour.

The results show that a complementary method based on the textual content of the metadata of the logs provides alternative insight into the activities of the hosts than the attributes. Moreover, the behaviour defined by the proposed method requires less extensive lookup than the behaviour defined by attributes. The inter-host distances based on the textual content allow understandable localisation of the host behaviour over time. Hence, this research provides an understandable method that will summarise the behaviour of the hosts over time, which enables the localisation of the logs requiring more advanced, in-depth analysis, and thereby reducing the amount of logs security analists need to consider during a compromise.

Preface

In 2018, I embarked on my Bachelor's journey in Computer Science and Engineering at TU Delft. My cohort, the first to experience a fully international program, began with over 800 students - an unprecedented number for the university in this field. We were told that by the start of the second year, this number would likely be halved. Entering the program with no prior programming experience and facing Java as one of my first courses, I was overwhelmed with self-doubt and emotional struggles. The initial two quarters were particularly challenging. I witnessed fellow students, some with programming backgrounds, leave the program after struggling to pass the courses. Yet, six years later, not only have I successfully completed my Bachelor's degree, but I am also on track to complete my Master's in Computer Science.

The key lesson I want to share is one I have been reminded of throughout my upbringing: anything is possible with hard work and determination. Do not be discouraged by the skills and experiences others may have that you lack; instead, focus on working diligently to acquire them.

However, without doubt, hard work alone is only fruitful if Allah wills. I am deeply grateful to Allah for granting me the strength and opportunity to reach this significant milestone in my life. I thank Him for blessing me with a family that has supported me unconditionally and never doubted my ability to succeed.

I am fortunate to have reached this milestone under the guidance of Sicco Verwer. His invaluable advice and the freedom he allowed me to explore my own path have been instrumental in my journey. The weekly meetings with him were a highlight of my experience, made even more enjoyable by the presence of Otte and Mirijam. Thank you, Sicco, for believing in me. Otte, your off-topic stories and unfiltered thoughts always brought a unique energy to our discussions, and Mirijam, your support during our shared moments of stress and mental breakdowns meant more than words can express.

My deepest gratitude goes to my parents, who have showered me with love, time, and comfort, enabling me to focus on my studies. Without their support and especially their prayers, I would not be the person I am today. InshaAllah, I will do my utmost to repay them for all the good they have given me in life.

Lastly, but certainly not least, I am profoundly thankful to my sister Kinza. She has always been my partner in crime, my best friend, and my emotional rock. Without her guidance and unwavering support, I would not be where I am today.

Mahira Ali
Delft, August 2024

Contents

1

Introduction

In the past decade, artificial intelligence (AI) has emerged as a go-to solution across various domains. While the foundational concepts of AI date back to 1955, it was during the 1990s and early 2000s that AI, particularly machine learning, began to be extensively applied to address diverse challenges in academia and industry [1]. A pivotal moment came in 2012 with a breakthrough in deep learning, a subset of machine learning focused on algorithms capable of automatically learning significant features from raw data [2]. This breakthrough captured widespread attention due to its remarkable performance in tasks such as image classification.

Since 2012, deep learning has increasingly been leveraged to address (complex) problems across a wide range of research domains, from healthcare to finance and from autonomous driving to natural language processing [3]. The integration of AI into everyday life and business operations has witnessed substantial growth [4]. A 2023 survey by IBM revealed that 42% of enterprise-scale businesses had integrated AI into their operations, with an additional 40% actively considering its implementation [5].

While it is undeniable that AI has ushered in numerous promising opportunities across various domains, it has also given rise to a phenomenon known as "AI solutionism. [6]" This refers to the pervasive mindset that AI can solve all conceivable problems when equipped with sufficient data. Such a perspective, however, is increasingly seen as a hindrance to the field's advancement, as it often overlooks critical ethical considerations and fosters unrealistic expectations regarding AI's capabilities [7]. The unbridled optimism surrounding AI solutionism may lead to neglecting essential nuances in problem-solving, including the socio-technical implications of AI deployment.

1.1. Motivation

An important application of AI lies in anomaly detection, which entails identifying rare items, events, or observations that deviate substantially from the majority of the data [8]. This capability is indispensable across various sectors, including cybersecurity, finance, and healthcare. Traditional methods of anomaly detection, such as rule-based systems and statistical analyses, often face substantial challenges when addressing high-dimensional data and complex patterns [9]. In this context, AI, particularly deep learning techniques, has transformed the field by providing sophisticated tools to accurately identify anomalies. These advancements capitalise on AI's capacity to process vast amounts of data while simultaneously accounting for the intricacies inherent in complex datasets.

However, the success of AI in this domain has led to a growing reliance on AI-based methods to the extent that traditional techniques are increasingly being disregarded [10]. This shift has also stifled the development of alternative approaches that might complement or enhance AI methods. The prevailing assumption that AI is the panacea for anomaly detection risks overlooking the limitations and potential pitfalls associated with such solutions.

AI solutions require large volumes of high-quality data for training [11]. In scenarios where such data is unavailable or biased, the model's performance can be significantly compromised. Even with sufficient data, training and deploying AI models for anomaly detection often require substantial computational resources, which can be cost-prohibitive and time-consuming. Additionally, scaling these solutions to different environments or types of data can be challenging and may require significant adaptation [12]. Furthermore, many AI solutions, particularly deep learning networks, operate as "black boxes," offering little transparency or interpretability in their decision-making processes [13]. This lack of explainability can be problematic, especially in high-stakes sectors where understanding the rationale behind anomaly detection is critical, especially in industries that require transparency and accountability in decision-making processes. This lack of transparency and interpretability also makes AI solutions a target for adversarial attacks, where slight perturbations in the input data are crafted to deceive the model into making incorrect detections.

1.2. Problem Statement

AI-based algorithms in anomaly detection face significant challenges due to their reliance on vast amounts of historical data, limited generalisability, and lack of transparency [14].
First, the dependence on large datasets necessitates considerable processing time, which is particularly problematic when real-time processing is required. The sheer volume of data can overwhelm available processing resources, leading to delays within the decision making.
Second, the limited generalisability of these models poses further difficulties. Models developed and trained in a specific context often struggle to perform adequately when applied to different systems, environments, or data types. This lack of adaptability is particularly concerning in anomaly detection, where the characteristics of anomalies can vary significantly across different domains and applications.
Third, the lack of transparency in AI-based algorithms complicates understanding the specific factors or events that trigger the identification of an anomaly. This opacity is especially problematic in critical sectors, where understanding the rationale behind anomaly detection is crucial for effective decision-making and response. The inability to interpret these triggers undermines trust in the AI system and hampers the ability to take targeted corrective actions, potentially leaving underlying issues unresolved.

These inherent shortcomings in existing solutions highlight critical issues, particularly the lack of transparency in explaining based on what certain hosts are flagged as anomalous. This absence of interpretability complicates understanding the rationale behind detected anomalies, making it challenging to verify and address potential issues effectively. Therefore, this research aims to address the following problem statement:

> *Developing a complementary method for anomaly detection that provides understanding into the behaviour of the hosts within the network, and thereby allowing to reduce the amount of logs requiring in-depth analysis.*

1.3. Outline

Chapter 2 reviews the relevant literature, which was essential for identifying the research gap this study addresses. Chapter 3 introduces the necessary background on the Burrows Delta method and its variants, ensuring that the reader is familiar with the method's mathematical principles and its application in this research. Then, chapter 4 defines the research objectives. Upon which, chapter 5 outlines an overview of the used methodology. In chapter 6, the analysis of attributes - namely, size (section 6.1), log types (section 6.2), and event IDs (section 6.3) - is presented. This chapter examines how these attributes cluster, assesses their effectiveness in identifying attacks and defines host behaviour. Chapter 7 evaluates the performance of the Burrows Delta method in identifying outlying hosts, detecting attacks, and defining host behaviour. It begins with section 7.1, which compares the clusters generated by different Delta methods and n-gram sizes, shortlisting the most effective methods for deeper analysis. Section 7.2 then assesses these shortlisted methods and sizes in terms of their ability to differentiate based on host behaviour. Section 7.3 applies the optimal Delta method and

n-gram size to define host behaviour, followed by section 7.4, which focuses on attack identification. Chapter 8 divides the dataset into time frames and conducts a behavioural analysis on the outlying host for each time frame. In chapter 9, factors that may have influenced the research are discussed and suggests future work. Finally, chapter 10 concludes the research by summarising the key results and formally answering the research questions.

2

Literature Review

This chapter provides an overview of the latest anomaly detection methods in the field of network and computer security. These methods are broadly categorised into two primary types: large-scale methods, which are explored in section 2.1, and log-based methods, detailed in section 2.2. Then, section 2.3 discusses methods for explainable AI, which represent key advancements in model interpretability. The chapter concludes with section 2.4, formally identifying the research gap this study aims to address, laying the foundation for the subsequent investigation.

2.1. Large-Scale Anomaly Detection Techniques

In July 2020, Thudumu *et al.* [16] employed advanced algorithms and tools to manage large volumes of data characterised by high velocity and variety. Their approach involved dimensionality reduction, feature extraction, and the application of machine learning algorithms to identify anomalies within the data. The novelty of this method lies in its capacity to effectively manage the complexities of high-dimensional big data while maintaining high accuracy and performance, even with large and diverse datasets. This approach overcomes the limitations of traditional methods, which struggle with the volume, velocity, and variety inherent in big data. However, dimensionality reduction techniques obscure the original features, making the final results not easy interpretable [17]. Moreover, the application of machine learning algorithms further contribute to the overall opacity of this solution as these generally lack transparency [18].

In June 2023, Liu and Wang [19] applied convolutional neural networks (CNNs) to the real-time detection of anomalies in network traffic. This method emphasised the extraction of statistical features from network traffic data, utilising the powerful feature extraction capabilities of CNNs to identify patterns indicative of anomalies. The primary strength of this approach lies in its ability to automatically learn and extract relevant features from raw network traffic data, significantly enhancing detection performance compared to traditional methods reliant on manual feature engineering. However, the internal workings of CNNs are complex to interpret due to their multiple layers [20]. Moreover, the non-linear transformations applied at each layer obscure the relation between the input and output, making it challenging to trace how decisions are made.

In June 2024, Jin *et al.* [21] introduced the use of large language models (LLMs) for detecting anomalies within computational workflows. By harnessing the advanced capabilities of LLMs, this approach sought to identify deviations from expected patterns in complex execution environments. The innovation of this method lies in its ability to leverage the extensive pre-trained knowledge embedded within LLMs to detect anomalies without extensive data preprocessing. This significantly reduced the need for domain-specific feature engineering, a common bottleneck in traditional anomaly detection techniques. However, Liao and Vaughan [22] highlight the functioning of LLMs as black boxes, making it difficult to understand how decisions are derived. As many LMMs are proprietary, there is minimal disclosure of the models' architecture. This opacity prevents understanding of models' behaviour and decisions.

2.2. Log-Based Anomaly Detection Techniques

In May 2021, Gu *et al.* [23] proposed a log-based anomaly detection system that combines a Bidirectional Slicing Gate Recurrent Unit (Bi-SSGRU) with an attention mechanism. Initially, logs are parsed into structured sequences using log keys, with distinct importance assigned to each segment through weighted attention. These weighted feature sequences are then used to train the Bi-SSGRU-Attention model, which facilitates parallelisation and reduces training time through SSGRU. However, Chefer *et al.* [24] highlight the inherent challenges of Bi-SSGRU-Attention models in achieving transparency. The combination of bidirectional GRUs and attention mechanisms results in a highly complex model architecture, making it difficult to trace the decision-making process. Moreover, while attention mechanisms emphasise important parts of the input data, they do not provide clear explanations for why certain parts are deemed important, adding to the opacity.

In November 2021, Liu *et al.* [25] introduced LogNADS, a network-log-based anomaly detection system. The logs are first transformed into templates after discarding irrelevant words. Semantic features are then extracted from these templates by selecting theme words, which are concatenated into low-dimensional vectors. For each theme word, a low-dimensional embedding is created, which reduces computational time costs. However, the transformation into templates, extraction of semantic features and creation of low-dimensional embeddings fail to preserve the structure of the original data [26]. This loss of information conceals the rationale behind anomaly detection decisions.

In September 2021, Lv *et al.* [27] presented ConAnomaly, a method that uniquely utilises both the semantic and sequential relationships within logs. ConAnomaly begins with log parsing, after which the parsed words are vectorised using the log sequence encoder log2vec. Following vectorisation, part-of-speech tagging is applied to filter out invalid words. The resulting vector representations are then aggregated into a sequence vector through a weighted average method. This approach enables ConAnomaly to capture semantic information within the logs and leverage their sequential relationships. This capability distinguishes it from many existing log-based anomaly detection methods that typically focus solely on the latter. However, the vectorisation and aggregation abstract the data into low-dimensional representations. Additionally, the multi-step approach of ConAnomaly to transform the data makes it difficult to trace how anomalies are detected.

In November 2021, Le and Zhang [15] introduced NeuralLog, an anomaly detection method designed to address the challenges posed by Out-Of-Vocabulary (OOV) words and semantic misunderstandings, which have been shown to cause significant detection errors. NeuralLog does not require log parsing; instead, it extracts the semantic meaning of log events and represents them as semantic vectors. These vectors are then used to identify anomalies via a transformer-based classifier that captures the contextual information of log sequences. However, extracting these semantic vectors requires sophisticated natural language processing techniques, which abstract the data, making it difficult to trace the raw log data back [28]. Moreover, the transformer-based classifiers are inherently complex due to their multi-head attention mechnisms, making it challenging to understand how they capture and utilise contextual information [29].

2.3. XAI Techniques

Ribeiro *et al.* [30] proposed LIME (Local Interpretable Model-agnostic Explanations) in 2016. This technique was designed to explain the predictions of any machine learning classifier by learning an interpretable model locally around the prediction. LIME works by perturbing the input data, where it makes slight modifications to create a set of new samples. These samples are then feeded into the original complex model to observe prediction changes. LIME then fits a simple, interpretable model, such as a linear model, to these perturbed samples to approximate the behaviour of the complex model. This surrogate model highlights which features contribute the most to the prediction. However, while LIME can identify important features, it does not provide detailed insights into specific feature values that led to the anomaly.

Shrikumar *et al.* [31] presented DeepLIFT (Deep Learning Important FeaTures) in 2017. This method decomposes the output prediction of a neural network on a specific input by backpropagating the contributions of all neurons in the network to every feature of the input. DeepLIFT initiates by defining a reference activation, which is typically the activation of neurons for a baseline input. For a given input, DeepLIFT compares the activation of each neuron to its reference activation. Based on the differences in activations, it assigns contribution scores to each input feature. These scores indicate how much each feature contributed to the model's output. These contributions are backpropagated through the network to attribute the output to the input features. Hence, while DeepLIFT can highlight which features were important in the model's decision, similar to LIME, it does not narrate the specific feature values that led to the anomaly.

Han *et al.* [32] proposed DeepAID in 2021. DeepAID is a framework designed to interpret and improve deep learning-based anomaly detection systems, particularly in security applications. Similar to LIME, DeepAID perturbs the input data to generate a range of samples. Then, DeepAID creates a reference data point, which is used to compare against the anomalous data. DeepAID uses gradient information to analyse the differences between the anomalous data and the reference. This helps in identifying which features or aspects of the input data are most responsible for the anomaly. By optimising the loss function, it ensures that its interpretations are accurate, stable, and concise. Unlike LIME and DeepLIFT, DeepAID not only highlights which features are important in the model's decision, but also, specifies the specific values that contributed to the anomaly, hence, offering a more detailed explanation. Yet, domain expertise is required to analyse and respond to the findings.

2.4. Research Gap

The analysis of the system-based and log-based anomaly detection methods show that the lack of transparency is a consistent drawback. On top of that, there is a growing adoption of machine learning, primarily deep learning, in anomaly detection methods, driven by the need to manage vast and complex datasets efficiently [16]. The techniques have emerged as a prominent solution in anomaly detection due to its precision in identifying anomalies while offering robust and scalable solutions. Traditional methods, which often depended on manual feature engineering, were constrained in their capacity to handle high-dimensional data. In contrast, machine learning models can autonomously extract relevant features and learn intricate patterns from data, making them well-suited for vast amount of data [33]. As data continues to increase in volume and complexity, the reliance on these techniques is expected to expand. However, AI techniques suffer inherently from opacity [34]. Hence, the rising usage of AI techniques within anomaly detection exacerbates the existing problem of transparency, further complicating the ability to interpret and understand the underlying decisions.

Existing explainable AI methods, such as LIME, DeepLIFT, and DeepAID, provide valuable insights into the features that contribute to the identification of anomalies. DeepAID, in particular, offers detailed insights into the specific values of these features associated with anomalous events. Nevertheless, its findings necessitates substantial domain-specific expertise [32]. This requirement can pose challenges in the practical implementation of its insights, particularly in interdisciplinary contexts where users may not possess deep familiarity with the domain from which the data originates. Consequently, while these explainable AI tools significantly advance the interpretability of AI systems, their effectiveness is constrained by the need for expert interpretation, underscoring the importance of developing methods that are not only interpretable but also lower the need for domain knowledge.

3

Stylometry

Stylometry, a branch of computational linguistics, is dedicated to quantitatively analysing linguistic features in natural language texts [35]. It provides a suite of methodologies applicable to various natural language processing (NLP) tasks. These tasks include, but are not limited to, authorship attribution, authorship verification, detection of style changes, and classification of written texts. Authorship attribution, a key area of research in quantitative text analysis, involves deducing the distinctive traits of an author from the characteristics of the texts attributed to them [36]. Contrary to analysing text content, the primary objective of authorship attribution is to estimate the likelihood that a specific author wrote a text based on stylistic traits [37]. Authorship attribution operates on the fundamental assumption that each individual possesses unique language usage habits, leading to stylistic similarities in texts authored by the same person [38]. Stylometry enables the detection of these idiosyncrasies by quantifying various linguistic features, such as the relative frequencies of function words or parts of speech, the richness of vocabulary, and more [39]. Sari, Stevenson, and Vlachos [40] deduced that frequency-based features have emerged as the predominant feature for quantifying the stylistic distance between texts. These features effectively encapsulate an author's preferences in terms of topics and writing style. The preeminent method for measuring stylistic distance, leveraging these frequency-based features, is Burrows' Delta [37].
Section 3.1 provides an in-depth exploration of the Burrows Delta method, beginning with an outline of its underlying mathematical principles in subsection 3.1.1. Following this, section 3.2 examines the main variants of the Burrows Delta, highlighting the key aspects in which they differ. Finally, section 3.3 addresses the relevance of these methods to the overarching research objectives, offering a discussion of their strengths and limitations.

3.1. Burrows' Delta

In 2001, John F. Burrows received the Roberto Busa Prize [1] for his groundbreaking contributions to stylometry [38]. During his acceptance speech, Burrows introduced a new metric, Delta, to this established field. Notably, Delta relied heavily on multivariate statistics, aiming to surpass the prevailing practice of comparing very small groups of candidates, as presented by the works of Bailey [41] and Binongo and Smith [42]. At the time, the existing state of the art primarily involved comparing two likely candidates. Burrows, however, perceived the need to surpass this limitation, expressing the primary objective of his proposed methodology as the imperative "to shake off these constraints [43]." Burrows recognised that prevailing techniques in authorship attribution were applicable primarily within a "closed game," where the suspected author was chosen from a limited list, often consisting of only one or two authors. Identifying a critical gap, he envisioned a methodology that could extend beyond this confined scope, allowing comparing an unattributed text with a more extensive pool of authors to identify a potential author or, at the very least, narrow down a short-list of authors [44]. To address this need, he introduced his Delta technique. The strategic reduction provided by this Delta method provided a sophisticated solution to streamline a large set of potential candidates to a smaller

[1] An award for outstanding, groundbreaking contributions in the application of information technology to the study of language, literature, and culture.

set such that more intricate, time-consuming, and computationally intensive multivariate statistics could be applied effectively. Since its introduction as a novel stylometric measure, Delta has emerged as a keystone in automated authorship attribution, establishing itself as one of the most recognized distance measures in this field [45]. As a similarity-based model, Delta operates on the fundamental concept of calculating pairwise similarity measures between an unseen text and all training texts [46]. By operating on the most frequent words in the training corpus as features, this methodology proves particularly valuable in clustering or classification tasks, attributing a text of unknown authorship to the most similar candidate within a typically closed set of authors [39]. The method is primarily evaluated on literary texts, encompassing English poems and novels [46]. Here, Delta has exhibited noteworthy efficacy. Notably effective for texts exceeding 1,500 words, Delta's attribution accuracy experiences a proportional decline with decreasing text length. Nevertheless, even in the case of relatively short texts - approximately 100 words -, the correct author tends to be prominently featured within the first five positions of the ranked authors, offering a pragmatic means to reduce the set of candidate authors [44]. This requirement underscores the method's reliance on an adequate length of text to extract meaningful stylistic features for accurate authorship assessments.

3.1.1. Mathematical Principles

This section provides a thorough understanding of the underlying mathematical principles inherent in Burrows' Delta method, as this is pivotal for a comprehensive comprehension of its efficacy in discerning textual deviations. Burrows' Delta measure is defined as:

> *the mean of the absolute differences between the z-scores for a set of word variables within a given text-group and the z-scores for the same set of word-variables in a target text* [43]

The initiation of the Delta method involves considering a collection, denoted as $\mathcal{D}$, comprising n_d number of text documents D. Each text document D within collection $\mathcal{D}$ serves as a representative of the writing style of an author and is represented by a profile of relative frequencies, denoted as $f_i(D)$, corresponding to n_w most frequent words (MFW) $w_1, w_2, ..., w_{n_w}$. The selection of these words and the determination of n_w, serve as influential parameters. The careful configuration of these parameters significantly influences the method's efficacy, emphasising the necessity for thoughtful consideration and fine-tuning of these aspects to optimise the performance of Burrows' Delta. Parameter n_w conferres former key findings related to optimising the selected words and n_w.
The complete profile of D is represented by the feature vector $f(D) = (f_1(D), ..f_{n_w}(D))$. Subsequently, these feature vectors undergo re-scaling, often through a linear transformation [39]. The conventional approach in Burrows' Delta involves re-scaling through a z-transformation. The application of the z-transformation is a widely adopted practice in stylometry [47]. It serves the purpose of standardising features, ensuring that each selected feature - specifically, words in this context - holds equal importance [48]. Nevertheless, as the standardisation method alters the scale of the data and its effectiveness is dependent upon the intrinsic properties of the data, the performance of the Delta method is affected by the chosen standardisation technique [49]. Therefore, it is essential to employ a transformation that aligns most effectively with the particular characteristics of the data concerned. Standardisation delves into the z-transformation, and outlines its assumptions.
After acquiring the z-scores for each w_i within every D, the dissimilarity between documents is calculated through the application of a distance metric. The selection of an appropriate distance metric holds paramount significance in quantitative authorship attribution, serving to measure the extent of similarity between two documents [38]. Distance Metric examines the distance metric employed in the original Burrows' Delta.

Parameter n_w

Originally, the parameter n_w was set to 150, with the initial 50 to 100 most common words proving effective candidates for reliably distinguishing between authors [46]. Burrows [43] explicitly articulates a fundamental insight in the field, emphasising the greater reliability of supporting conclusions with numerous 'weak discriminators' instead of relying on a few robust discriminators. This insight is corroborated by Evert, Proisl, Jannidis, *et al.* [38], who demonstrated that information essential for identifying the author of a text resides in the profile of deviation across the MFW rather than in the magnitude of the deviation itself. Notably, the profile, encompassing the entire spectrum of MFW

frequencies, proves to hold pertinent information, with specific words contributing less discernible value.
Furthermore, Smith and Aldridge [44] revealed in their systematic analysis of Burrows' Delta method that no discernible enhancement in author short-listing occurs within the range of 200 to 500 words. However, the capacity to accurately identify the correct author gradually improves within this span. Beyond 500 words, a descent is observed in both short-listing and author identification efficacy. Notably, the utilisation of a substantial word vector significantly amplifies the likelihood of comparisons based on hapaxes within the comparison vector. Consequently, they assert that extending the word vector beyond 200–300 words is likely to be counterproductive. It is noteworthy that the selection of the word vector for the Delta method relies solely on the frequency of occurrence in the corpus, lacking an objectively determined criterion for the optimal number of words. Given the curse of dimensionality, the mere addition of words to the word frequency vector should not be assumed to yield improved results inherently. Significantly, Smith and Aldridge [44] findings underscore that text length is a far more crucial factor than word vector length for authorship attribution accuracy. As text length increases, the accuracy of authorship identification rises, with more promising results observed for author short-listing. Their experiments reveal the phenomenon where the word frequency vector's performance deteriorates when the text length under consideration contains fewer words than the number of word dimensions in the word frequency vector, suggesting potential overfitting for larger word frequency vectors on shorter texts. Additionally, their experiments highlight the critical role of the word selection process. It emphasises that commencing the selection with the MFW or MFW-F/C - the top n words taken with the corpus ordered based on the most frequent word while ensuring that all function words precede content words -, significantly contributes to the success of employing Burrows' Delta. Optimal performance is achieved by MFW and MFW-F/C, offering consistent efficacy at word vector sizes up to 150 and beyond 300 words.

Standardisation

In the realm of Burrows' Delta and a majority of its variants, a conventional practice for re-scaling features involves the application of the z-transformation, also known as the z-score normalisation [38]. The z-transformation is a statistical technique employed to standardise data within a dataset to a standard normal distribution such that the relative standing of a raw score within its original distribution can be measured. This enables the comparison of data points from different distributions by expressing them in terms of standard deviations from the mean. Let μ_i be the mean of the distribution of f_i in document $D \in \mathcal{D}$; σ_i the standard deviation of the distribution of f_i in document $D \in \mathcal{D}$;and $f_i(D)$ the frequency of word i in document $D \in \mathcal{D}$. The z-transformation aims to standardise the frequency of the word to have a mean of 0 and a standard deviation of 1. The z-transformation of word i in document $D \in \mathcal{D}$, denoted as $z_i(D)$, is formally defined as:

$$z_i(D) = \frac{f_i(D) - \mu_i}{\sigma_i} \tag{3.1}$$

While the z-transformation serves as a valuable tool, its effectiveness is contingent upon the alignment of data characteristics with the underlying assumptions of the transformation. Three key assumptions concerning the dataset must be satisfied for the z-transformation to yield meaningful results:

1. **Normally Distributed:** The Z-transformation assumes that the underlying data adheres to a normal distribution [50]. This assumption is intrinsic to the interpretation of a Z-score, which is derived from the properties of a standard normal distribution - a distribution with a mean of 0 and a standard deviation of 1. When the data conforms to a normal distribution, the Z-score provides a meaningful measure of how many standard deviations a particular data point is from the mean. However, it is important to note that if the data does not exhibit a normal distribution, the Z-score may lose its interpretative value [51]. Deviations from normality can affect the accuracy of the Z-score in representing the relative position of a data point within its distribution.
2. **Large Dataset:** The reliability of Z-transformation is enhanced when applied to larger sample sizes [52]. This phenomenon is grounded in the principles of the Central Limit Theorem, a fundamental concept in statistics [53]. According to the Central Limit Theorem, as the sample size increases, the distribution of sample means tends to approximate a normal distribution, irrespective of the shape of the population distribution [54]. In practical terms, this means that for larger samples, the

Z-transformation is more likely to yield robust and dependable results. The convergence toward a normal distribution, as dictated by the Central Limit Theorem, contributes to the stability and accuracy of Z-scores, reinforcing their utility in statistical analyses.

3. **No outliers:** The presence of extreme values or outliers in a dataset can exert a disproportionate influence on both the mean and standard deviation, subsequently impacting the accuracy and interpretability of Z-scores [55]. Extreme values can distort the normality assumptions inherent in the Z-transformation, potentially yielding misleading results.

The principal aim of the z-transformation in Burrows' Delta is to consider all words as potential markers of equal influence, thereby alleviating the impact of variations in raw frequency profiles [38]. Argamon [56] underscores the assumption of the z-transformation, emphasising that its meaningfulness relies on the relative frequencies approximating a Gaussian distribution across the texts in collection D. Building upon this premise, an alternative is proposed and detailed in subsection 3.1.3, motivated by additional distinctions in distance metrics.
However, Evert, Proisl, Jannidis, *et al.* [38] have demonstrated that, even when the data deviates from a normal distribution, words ranking above 100 in MFW counts contribute minimally to the overall frequency profiles without standardisation. While the outcomes exhibit resilience concerning the number of words, the few MFWs that do contribute meaningfully fall short of achieving satisfactory clustering quality. Consequently, the z-transformation emerges to diminish the influence of top-scoring words, particularly within the framework of Zipf's law, which characterises the distribution of word frequencies in a language [57]. According to Zipf's law, a small subset of words exhibits exceptionally high frequencies, while the majority of words occur infrequently. Without the z-transformation, these high-frequency words could disproportionately impact the distance metric.

Distance Metric

Burrows' Delta is founded upon the Manhattan distance between the z-scores of two documents, D and D', and is formally defined as follows:

$$\Delta_B(D, D') = ||z(D) - z(D')||_1 \tag{3.2}$$

$$= \sum_{i=1}^{n_w} |z_i(D) - z_i(D')| \tag{3.3}$$

Argamon [56] showed that this can be derived from the definition provided at the beginning of subsection 3.1.1, where the delta measure is quantified as the average absolute difference of the z-scores, expressed by the formulation:

$$\Delta(D, D') = \frac{1}{n} \sum_{i=1}^{n} \left| z(f_i(D)) - z(f_i(D')) \right| \tag{3.4}$$

Further algebraically simplified yields:

$$\Delta(D, D') = \frac{1}{n} \sum_{i=1}^{n} \left| z(f_i(D)) - z(f_i(D')) \right| \tag{3.5}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left| \frac{f_i(D) - \mu_i}{\sigma_i} - \frac{f_i('D) - \mu_i}{\sigma_i} \right| \tag{3.6}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left| \frac{f_i(D) - f_i(D')}{\sigma_i} \right| \tag{3.7}$$

This algebraic simplification underscores that the Delta measure is independent of the mean frequencies and can be viewed as the normalisation difference between the frequencies.
Argamon [56] argued that the Burrows' Delta is equivalent to a probabilistic ranking. Interpreting Burrows' Delta as a means of ranking authorship candidates based on their probabilities is advocated by Stein and Argamon [58] as it offers a nuanced comprehension rooted in its geometric foundations. Hence, when employing Burrows' Delta as a ranking metric, the constant factor $\frac{1}{n}$ becomes irrelevant.

The constant factor $\frac{1}{n}$ - dependent on the total number of words - is introduced to obtain the average. Since this causes a linear transformation, the relative order of ranked items is preserved. Consequently, omitting the element $\frac{1}{n}$ renders the formula as equivalent to Burrows' Delta:

$$\Delta(D, D') = \sum_{i=1}^{n} \left| \frac{f_i(D) - f_i(D')}{\sigma_i} \right| \tag{3.8}$$

$$= \sum_{i=1}^{n} \frac{1}{\sigma_i} \left| f_i(D) - f_i(D') \right| \tag{3.9}$$

$$= \sum_{i=1}^{n_w} |z_i(D) - z_i(D')| \tag{3.10}$$

$$= \Delta_B(D, D') \tag{3.11}$$

Thus, Burrows' Delta defines the scaled distance between two documents, calculated as the sum of absolute differences across each dimension independently, a metric known as the Manhattan distance. This method estimates the likelihood of a document being authored by a particular candidate based on the similarity in frequencies of each word usage independently. This shows that Delta operates by ranking authorship candidates according to the distance of their document D from the test document D'. Each dimension of difference - corresponding to a word frequency - is scaled by a factor of $\frac{1}{\sigma_i}$. This scaling implies that smaller differences are accorded greater significance when the variance in word frequencies is less. Consequently, Burrows' Delta can be regarded as an axis-weighted variant of the 'nearest neighbour' classification method [59]. In this context, a test document is classified as being authored by the writer whose known work exhibits the smallest 'distance' concerning the Delta measure.

3.2. Variants of Burrows' Delta

Since the introduction of Burrows' Delta, researchers have sought to enhance its performance by proposing various variants. The first notable variant was presented by Hoover in 2004, known as Delta Prime, which distinguished between positive and negative scores [38]. This differentiation emphasises the presence of features absent in one of the documents, offering a more accurate measure of stylistic dissimilarity. Unfortunately, the details of Hoover's Delta Prime remain elusive, as Hoover's paper titled "*Delta Prime?*" is inaccessible, and other references to the Delta Prime method lack sufficient information about its inner workings. Nonetheless, the impact of this limitation is partially mitigated by a survey conducted by Stamatatos [46], which indicated that Hoover's Delta does not outperform the original Burrows Delta. However, Hoover acknowledged a lack of compelling theoretical justification for his proposed method, a shared concern with Burrows' Delta [56]. In response, in 2008, Argamon [56] explored the geometric interpretation of Burrows' original Delta measure to address this gap. This exploration resulted in several Delta measure extensions, namely the Quadratic, Linear and Rotated Delta, intended as alternatives to Burrows' Delta. Subsequently, in 2011 Smith and Aldridge [44] introduced the Cosine Delta. Hereafter, in 2015, Eder proposed Eder's Delta and Simple Delta [38].

3.2.1. Quadratic, Linear and Rotated Delta

Argamon [56] claimed that the selection of an appropriate distribution for modelling word frequencies is an empirical decision. The Gaussian distribution, allowing for a more midrange spread around the mean, contrasts with the Laplace distribution, which permits a higher likelihood of outliers as outliers strongly influence the mean. The stability of the Laplace distribution, especially in scenarios where outliers are prevalent, makes it a preferred choice. Consequently, the decision between these two distributions depends on the expected characteristics of the author's texts. A Gaussian distribution may be more suitable for scenarios expecting similar frequencies for common words with a moderate spread and minimal outliers. On the other hand, if tightly clustered frequencies with a higher likelihood of a few highly atypical word frequencies are expected, the Laplace distribution might be a more appropriate choice.

Argamon identified a misalignment in Burrows' Delta concerning the standardisation method and the distribution model, upon which he proposed the Quadratic and Linear Delta methods. Argamon demonstrated that ranking by the Euclidean distance aligns with ranking by the highest probability in a multivariate Gaussian distribution. In contrast, ranking by the Manhattan distance corresponds to a Laplace distribution. This insight revealed a misalignment in Burrows' Delta, which employs the Manhattan distance but normalises by the mean and standard deviation, a practice consistent only with a Gaussian distribution. To rectify this methodological incongruity, Argamon proposed two variants: the Quadratic and Linear Delta methods.

Quadratic Delta

The Quadratic Delta method employs the z-transformation, similar to Burrows' Delta, but is based on the Euclidean distance. Accordingly, Quadratic Delta is defined as the sum of squared deviations of standardised word frequencies and is defined as follows:

$$\Delta_Q(D, D') = ||z(D) - z(D')||_2^2 \tag{3.12}$$

$$= \sum_{i=1}^{n_w} (z_i(D) - z_i(D'))^2 \tag{3.13}$$

$$= \sum_{i=1}^{n_w} \frac{1}{\sigma_1^2}(f_i(D) - f_i(D'))^2 \tag{3.14}$$

As Delta serves as a ranking principle, only the relative values are pertinent. Therefore, in Equation 3.13, the square root has been omitted. These adjustments resolve the methodological mismatch by applying standardisation to a ranking with an appropriate underlying probability distribution, specifically the Gaussian distribution. Argamon demonstrated that the Quadratic Delta method is equivalent to maximising a probability following the Gaussian distribution. This equivalence implies that employing the Quadratic Delta for authorship candidate selection corresponds to choosing candidates with the highest probability.

Linear Delta

Alternatively, the Linear Delta method was presented to address this methodological mismatch. The Linear Delta method retains the Manhattan distance, akin to Burrows Delta, but standardises the relative frequencies using the parameters of the Laplace distribution, namely 'median' and 'spread'. Let a be the median, representing the value such that half of the set of numbers is higher and the other half is lower; and let b be the spread in the distribution. Then, a and b can be estimated for each word from the word frequencies in the document collection $\mathcal{D} = D_1, ..., D_m$ as follows:

$$a_i = \text{median}(\langle f_i(D_1), f_i(D_2), ..., f_i(D_m)\rangle) \tag{3.15}$$

$$b_i = \frac{1}{n} \sum_{j=1}^{m} \left| f_i(D_j) - a_i \right| \tag{3.16}$$

This adaption preserves the structure of Burrows' Delta while more firmly establishing it as a probabilistic ranking principle. Accordingly, Linear Delta is defined as the average absolute deviation of word frequencies from the median word frequency and can be formally defined as:

$$\Delta_L(D, D') = \sum_{i=1}^{n} \frac{1}{b_i} \left| f_i(D) - f_i(D') \right| \tag{3.17}$$

Rotated Delta

Similar to Burrows' Delta, both variants of Argamon - the Quadratic and Linear Delta - operate under the assumption that the frequencies of individual indicator words are statistically independent of each other. However, this assumption is inherently inaccurate in most cases, posing a significant theoretical challenge for the general application of the Delta method. In response to this limitation, Argamon

introduced the Rotated Delta, aiming to relax the stringent independence assumption.
When frequencies of different indicator words are not statistically independent, it implies a non-zero covariance. Since complete access to all potential documents is not feasible, the covariance must be estimated from the collected document set $\mathcal{D}$. The estimated covariance σ_{ij} between the frequencies of words w_i and w_j is defined as:

$$\sigma_{ij} = \frac{1}{|C|} \sum_{D \in \mathcal{D}} (f_i(D) - \mu_i)(f_j(D) - \mu_j) \tag{3.18}$$

These covariances are organised into a covariance matrix S, with its inverse denoted as S^{-1}, defined as:

$$S = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{bmatrix} \tag{3.19}$$

In instances where all variables are independent, elements of the matrix other than those on the diagonal are zero. Correspondingly, the relevant word frequencies in a given document are represented by a vector of all frequencies:

$$\vec{f}(D) = \begin{bmatrix} \vec{f}_1(D) \\ \vec{f}_2(D) \\ \vdots \\ \vdots \\ \vec{f}_n(D) \end{bmatrix} \tag{3.20}$$

Hence, the Rotated Delta, which rotates the frequency differences into a space where they are maximally independent, can be formally defined as:

$$\Delta_R(D, D') = (\vec{f}(D) - \vec{f}(D'))^T S^{-1} (\vec{f}(D) - \vec{f}(D')) \tag{3.21}$$

$$= \sum_i \sum_j (f_j(D) - f_j(D'))(S^{-1})_{ij} \times (f_i(D) - f_i(D')) \tag{3.22}$$

The Rotated Delta assumes a Gaussian distribution, providing a firmer theoretical foundation that facilitates result justification. However, it's crucial to note that S does not always have an inverse. Specifically, if the number of texts in collection $\mathcal{D}$ is fewer than the number of MFWs considered, S is guaranteed to lack an inverse. Nevertheless, in low-dimensional cases, given that $S = EDE^T$, where D is the diagonal matrix, and E is the square eigenvector matrix, the eigenvalue decomposition of S can be employed to obtain E and D. Hence, given the decomposed matrices E_* and D_*, the rotated delta is formally defined as:

$$\Delta_R(D, D') = (\vec{f}(D) - \vec{f}(D'))^T E_* D_*^{-1} E_*^T (\vec{f}(D) - \vec{f}(D')) \tag{3.23}$$

Unfortunately, there is no mathematically and computationally tractable solution for multivariate Laplace distributions with correlated variables, as there is no universally accepted multivariate generalisation of the Laplace distribution. Consequently, these methods impose a challenge when word frequencies follow an underlying Laplace distribution and exhibit interdependence, as no Delta metric is theoretically justifiable.

3.2.2. Cosine Delta

Smith and Aldridge [44] introduced the Cosine Delta as a consequence of exploring angular similarity. Angular similarity assesses vectors based on their angular separation, disregarding the scaler element of the vector - unlike Euclidean distance, which expresses separation in terms of the scaler element. The motivation for experimenting with angular similarity stems from the well-known fact that large word

vectors exhibit greater reliability in text mining under angular similarity measures.
The Cosine Delta aligns with Burrows' Delta in employing the z-transformation on the frequencies of the most frequent word. However, instead of utilising the Manhattan distance measure, it uses the cosine similarity as an angular similarity measure. The cosine similarity measures the similarity between two n-dimensional vectors by finding the cosine of the angle between them. Let $x = z(D)$ and $y = z(D')$ be the z-transformed MFW-vectors of documents D and D' respectively; $x^T y = \sum_{i=1}^{n_w} x_i y_i$ be the dot product; and $||x||_2 = \sqrt{\sum_{i=1}^{n_w} x_i^2}$ be the length of the vector x according to the Euclidean norm. Consequently, the Cosine Delta is defined as follows:

$$\Delta_C(D, D') = \cos \frac{x^T y}{||x||_2 \cdot \left\|y\right\|_2} \tag{3.24}$$

Evert, Proisl, Vitt, *et al.* [39] demonstrated a close connection between the angular distance and the Euclidean distance due to the Euclidean norm. The Euclidean norm can be expressed as a dot product:

$$||x||_2^2 = x^T x \tag{3.25}$$

Therefore,

$$\begin{aligned} ||x||_2^2 &= (x - y)^T (x - y) && (3.26)\\ &= x^T x + y^T y - 2x^T y && (3.27)\\ &= ||x||_2^2 + \left\|y\right\|_2^2 - 2\,||x||_2 \left\|y\right\|_2 \cos\alpha && (3.28) \end{aligned}$$

Hence, if the frequency profiles are normalised concerning the Euclidean norm $||x||_2 = \left\|y\right\|_2 = 1$, the Euclidean distance becomes a monotonic function of the angle α.

$$||x||_2^2 = 2 - 2\cos\alpha \tag{3.29}$$

As a result, the Quadratic Delta and the Cosine Delta are equivalent for normalised profiles. Hence, the distinction between the Quadratic and Cosine Delta lies in the normalisation parameter; they are not based on genuinely different distance metrics. The Cosine Delta is an exclusive variant within the Delta family as it explicitly addresses profile normalisation. This normalisation enhances the method's robustness in comparing documents of varying lengths, focussing on stylistic dissimilarities rather than the overall frequency variations due to text length. Jannidis, Pielström, Schöch, *et al.* [45] and Evert, Proisl, Jannidis, *et al.* [38] demonstrated that the Cosine Delta outperformed all variants, attributing its success to the normalisation, which makes the measure more robust against the choice of MFW. In this measure, the difference in direction is considered the decisive factor for authorship attribution rather than the length of the vectors. Nonetheless, there is no clear understanding of why Cosine Delta and Burrows' Delta are robust and reliable.

3.2.3. Eder's Delta and Simple Delta

Like Hoover's Delta Prime, Eder's official publications proposing his alternatives are inaccessible. However, other references provide valuable information about Eder's Deltas. Jannidis, Pielström, Schöch, *et al.* [45] demonstrated that Eder's Delta outperformed Hoover's and Argamon's Delta. Additionally, Stanikūnas, Mandravickaitė, and Krilavičius [60] showed that Eder's Simple Delta performs exceptionally well on Lithuanian texts.
Eder's Delta and Simple Delta were proposed based on the observation that Burrows' Delta could perform better with highly inflected languages [60]. Inflected languages exhibit rich morphological inflexions, where words change form based on factors such as tense, gender, number, and other grammatical features [61]. Examples of inflected languages are Russian, Latin and Finnish. The existing methods in the Delta family struggle with sparsity due to the vast number of possible word forms. Collecting sufficient training data for all inflected forms can be impractical. Hence, Eder proposed Eder's Delta. This measure, like Burrows' Delta, employs the Manhattan distance and the z-transformation. However, it introduces the Eder Ranking Factor (ERF) alongside the z-transformation, defined as

follows:

$$ERF = \frac{n_w - n_i + 1}{n_w} \tag{3.30}$$

Hence, Eder's Delta can be defined as:

$$\Delta_E = \sum_{i=1}^{n_w} \left(|z(D) - z(D')| \cdot \frac{n_w - n_i + 1}{n_w} \right) \tag{3.31}$$

ERF is a relevance factor used to give more weight to terms that are both common and distinctive rather than just common [62]. It helps balance the importance of terms in a document based on their occurrence frequency, preventing the overemphasis on distinct inflected forms that could lead to suboptimal predictions.
Alternatively, Eder proposed Eder's Simple Delta, which also employs the Manhattan distance but applies square root normalisation instead of the z-transformation. Hence, Eder's Simple Delta can be defined as:

$$\Delta_S = \sum_{i=1}^{n_w} \left| \sqrt{f_i(D)} - \sqrt{f_i(D')} \right| \tag{3.32}$$

The square root transformation is useful for normalising skewed distributions, as it compresses high values and spreads out low values [63]. In the context of highly inflected languages, the choice of square root normalisation might be influenced by the fact that these highly inflected languages often have a skewed distribution of word forms, with a small number of highly frequent forms and a large number of infrequent forms. However, it is noteworthy that none of the references explains why Eder applied the ERF or square root normalisation to address inflected languages.

3.3. Strengths, Limitations and Relevance

Burrows Delta, a method originally developed in the field of stylometry for authorship attribution, and its variants offer notable strengths and limitations, which are outlined below:

Strengths:

1. **Simplicity and Accessibility:** Burrows Delta is straightforward to implement and understand. It calculates the mean of the absolute differences between the z-scores of word frequencies in different texts [38]. This simplicity makes it accessible for a wide range of applications, requiring minimal computational resources and expertise.
2. **Versatility Across Genres and Languages:** One of the significant advantages of Burrows Delta is its insensitivity to genre and language variations [56]. This characteristic makes the method highly adaptable, allowing it to be applied to diverse types of data without extensive modifications.
3. **Effectiveness in High-Dimensional Spaces:** Variants of Burrows's Delta, such as Cosine Delta and Quadratic Delta, excel in high-dimensional spaces [64]. These variants are particularly effective at capturing subtle differences in feature distributions, which is essential for tasks that involve complex datasets with numerous variables.
4. **Robustness to Noise:** The method demonstrates a degree of robustness to noise, as it relies on aggregate measures of feature differences rather than being affected by individual outliers [56]. This robustness enhances its reliability in real-world scenarios where data can be noisy or incomplete.

Limitations:

1. **Dependence on Feature Selection:** Burrows Delta's accuracy is highly dependent on feature selection [38]. Poor feature selection can result in misleading or inaccurate outcomes, limiting the method's reliability in cases where the feature space is poorly understood.
2. **Lack of Theoretical Justification:** Burrows Delta lacks a strong theoretical foundation despite its empirical success [56]. This absence of theoretical backing can undermine the interpretability of the results and reduce confidence in the method's effectiveness.

Applying Burrows Delta or its variant to log message analysis is highly relevant for identifying anomalous hosts and their behaviours. The method's ability to quantify stylistic differences in the text can be effectively utilised to detect deviations in behavioural patterns, which may indicate unusual or suspicious activities. By transforming log messages into feature vectors based on n-gram frequency, Burrows Delta can precisely highlight deviations from expected behaviour. This capability is particularly valuable for flagging specific hosts and logs that warrant closer scrutiny, enabling more targeted and efficient anomaly detection.

4

Research Objectives

This research investigates whether it is possible, based on the metadata of the logs of the hosts, to identify the outlying host through inter-distances and define the behaviour of the hosts within the network. By defining the behaviour, the aim is to locate the timeframe and cause of the incident accurately. This leads to the following main research question:

RQ: *Can the metadata of the logs of the hosts within the network effectively quantify the inter-host distances to define host behaviour such that it can identify the outlying host and the time frame of the incident?*

Log metadata can be compartmentalised into attributes, such as event IDs and sizes, and textual content, like general messages. This research evaluates both components' effectiveness in quantifying the inter-distance between hosts to identify outlying hosts. Specifically, the first component focusses on attributes of log metadata. Therefore, the first research question investigates the effectiveness of these attributes.

RQ1: *How effectively do the attributes of the log metadata quantify the inter-host distances to define the behaviour of the hosts such that it can identify the outlying host?*

Then, the effectiveness of the second component of log metadata—the textual content—is evaluated in quantifying the inter-host distances to define the behaviour and, in turn, identify outlying hosts. The textual content requires filtering words and creating n-grams. Additionally, a statistical measure is employed due to the variability in textual content, considering semantic content and contextual relevance. This allows for more accurate measurement of inter-host distances and detection of outlying hosts. Specifically, Burrows's Delta, a prominent method in authorship attribution, has been chosen as the statistical measure. Numerous variants of Burrows's Delta have been proposed, and their effectiveness must be assessed. Hence, the optimal n for n-grams and the best delta method must be determined to evaluate the textual content's effectiveness.

RQ2: *How effectively does the textual content of the log metadata quantify the inter-host distances to define the behaviour of the hosts such that it can identify outlying hosts?*

RQ2 (a): *What is the optimal choice of n for constructing n-grams from the textual content, balancing information preservation and contextual clarity?*

RQ2 (b): *What is the best choice for the delta method?*

Once the optimal n for n-grams and the best delta method have been determined, the log entries will be divided into active timeframes. Then, the selected delta method, configured with the chosen n-gram, will be applied to each active timeframe. This approach will allow to measure the inter-distances between hosts and define the behaviour of each host within each timeframe. Based on these analyses, the following research question will investigate the accuracy and effectiveness of describing host behaviour within the timeframes and determining the precise timeframe during which an incident occurred.

RQ3: *Can the textual content - measured utilising the selected delta method with the chosen configuration - accurately define the behaviour of the hosts within each active timeframe and identify the timeframe during which an incident occurred?*

5 Methodology

This chapter initiates by describing the data utilised during this research in section 5.1. Then, section 5.2 outlines the proposed solution. This is followed by defining the research scope in section 5.3. Lastly, the contributions of this research are provided in section 5.4.

5.1. Dataset

This research aims to develop a complementary method that provides insight into the behaviour of the hosts. To achieve this, datasets from APTA Technologies[1] were utilised during the method's development. This dataset consist of multiple unsupervised incident cases where the Windows EVTX files are collected from various hosts. The structure of an incident case within this dataset is illustrated in Figure 5.1. During the data processing stage, the EVTX files were first converted into CSV files, which were subsequently transformed into TXT files, as represented by the dotted CSV and TXT folders in the figure. Examples of the structure of these files are provided in Figure 5.2.

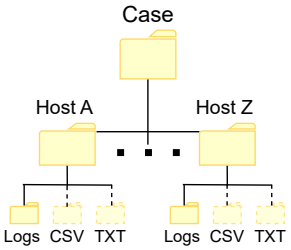
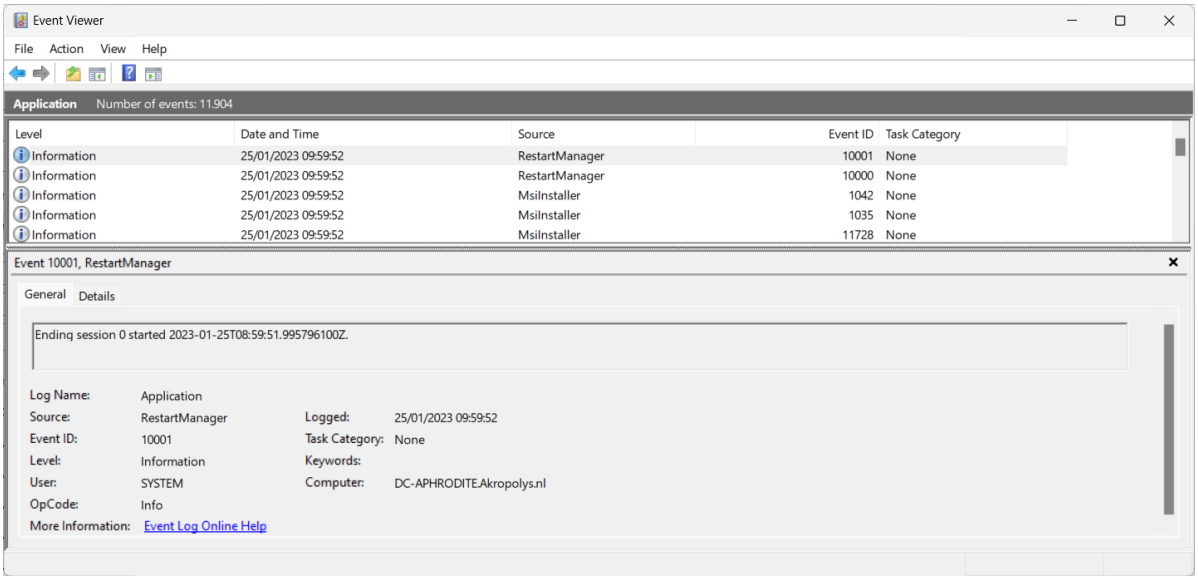


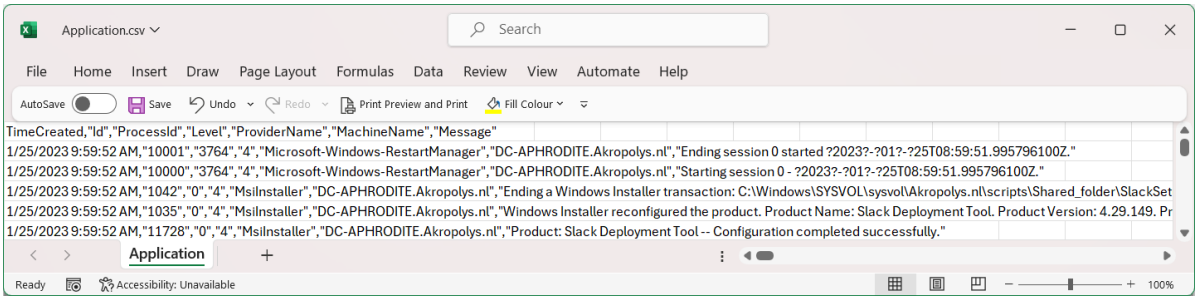
Figure 5.1: The folder structure of an incident case present in the dataset

The APTA Technologies dataset comprises four distinct Demo-Cases, each featuring a different injected attack while maintaining the same base log structure. Specifically, in Demo-Case 1 an RDP Brute Force attack on the Domain Controller occurs, while, in Demo-Case 2 an RDP Brute Force attack on the server occurs followed by a lateral movement. In an RDP Brute Force Attack the attacker repeatedly attempts to gain access to a remote server using the RDP (Remote Desktop Protocol) by guessing the username and password. Moreover, in Demo-Cases 3 and 4 there is a ProxyShell abuse of the exchange server with lateral movements. The Proxyshell Abuse leverages vulnerabilities in the Exchange server to execute arbitrary code and gain unauthorised access.

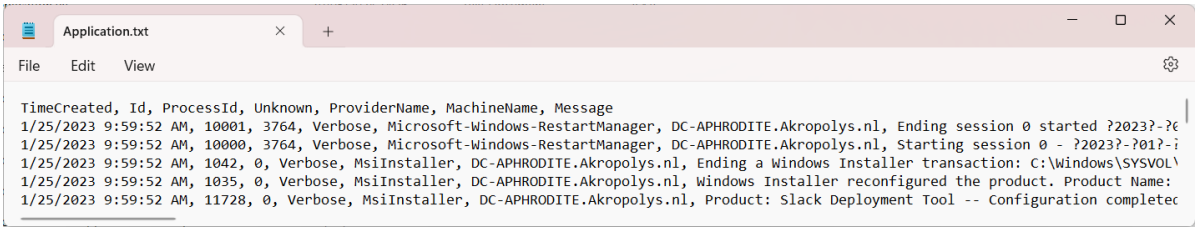
[1] https://www.apta.tech/



(a) EVTX file



(b) CSV file

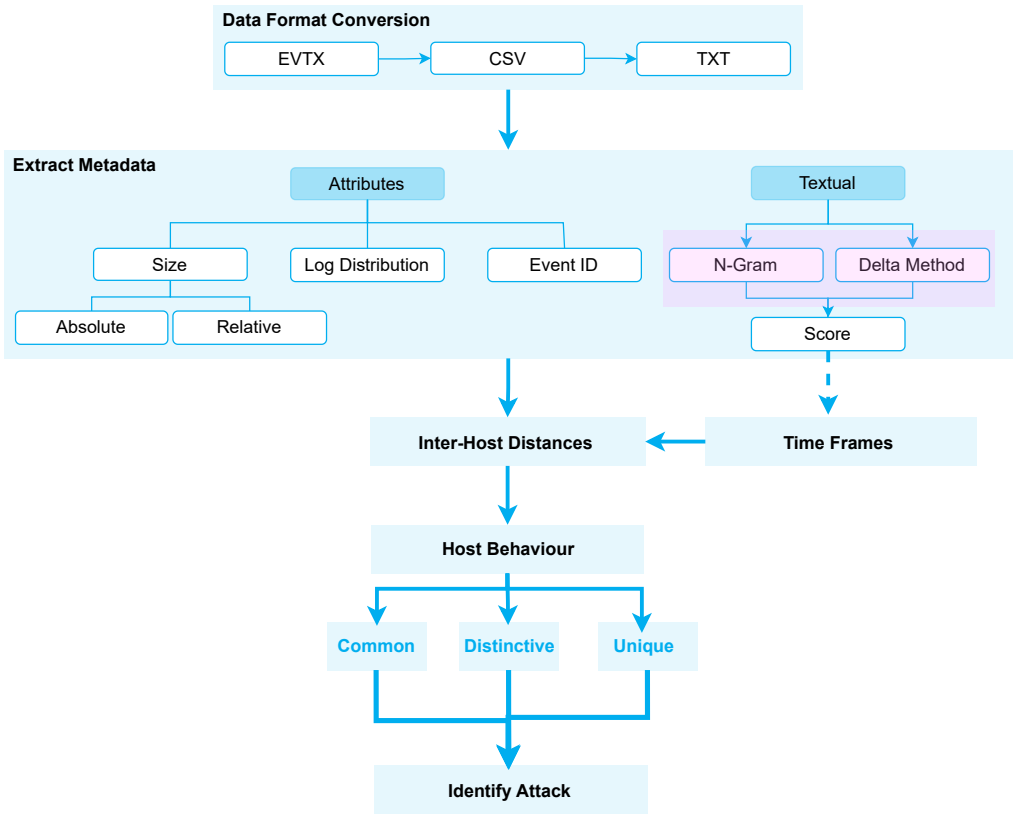


(c) TXT file

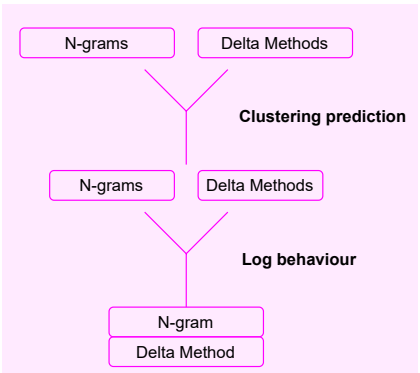
Figure 5.2: Examples of EVTX, CSV, and TXT file formats, showcasing the first five events from Demo-Case 1 on the host DC-Aphrodite, focusing on the 'Application' log type

5.2. Proposed Solution

The core intuition underlying the proposed solution is that general log messages can be interpreted as records documenting the activities of their respective hosts. Just as an author can be identified through the distinctive characteristics of their written documents - an approach known as authorship attribution - similarly, a host can be identified based on the unique characteristics of its log types. Each host generates its own set of records, which are defined here as specific EVTX log types. Figure 5.3 provides an overview of the flow of the method's construction.



(a) The complete flow of the contruction of the method



(b) The flow of the selection of the optimal delta method and n-gram size

Figure 5.3: An overview of the proposed method

Initially, each EVTX file is converted into a CSV file using the Windows Event Viewer's Export-CSV function. Subsequently, the CSV file is transformed into a TXT file using Python's built-in `csv` module [2]. The rationale behind converting the EVTX file to a TXT format is to ensure that the proposed method is not limited to Windows logs but can be generalised to logs from different operating systems.
The log metadata is divided into two components: attributes and the general log message, which is the textual content. Among the attributes, three types are extracted: size, log types, and event IDs. The size attribute is further distinguished into absolute and relative measures. To obtain numerical representations of the log types and event IDs, the frequency of each type and ID is computed for each specific host. Likewise, a numerical representation of the textual content is constructed by dividing the message of each log event into n-grams, where 'n' refers to the number of words. These scores are then normalised using Min-Max normalisation. These resulting numerical representations of the attributes and the textual content are utilised to quantify the inter-host distances, with the goal of defining the

[2] https://docs.python.org/3/library/csv.html

behaviour of the hosts based on these distances. Then, the defined behaviour is analysed to argue why the particular host is identified as the outlier based on its behaviour.
In order to determine the most appropiate n-gram size and delta method, which are detailed in chapter 3, they are first shortlisted based on whether their clustering aligns with the prediction. These shortlisted delta method and n-gram sizes are analysed based on their quantification of specific logs. This result into a selection of the most appropiate Delta method and n-gram, which are used for further analysis such as defining the behaviour of the hosts and identification of the attack.
Finally, the most appropriate n-gram size and Delta method are employed to quantify the inter-host distances for each time frame within the case. This allows for an analysis of the hosts' temporal behaviour, ultimately enabling the identification of the time frame in which the incident most likely occurred.

5.3. Research Scope

The effectiveness of log-based anomaly detection methods is significantly influenced by the post-processing techniques applied to the data [15]. In this research, basic post-processing steps, such as de-capitalisation, removal of numerical values and stopwords, have been implemented. These steps are intended to standardise the textual content and reduce noise, thereby facilitating more accurate analysis. However, the investigation of more advanced or optimised post-processing techniques, which could potentially enhance the method's performance, falls outside the scope of this study. Future work could explore the impact of various post-processing strategies on the efficacy of the proposed method, but this project focuses on establishing a baseline approach that balances simplicity and functionality.

Moreover, the proposed method is inherently designed with adaptability in mind, allowing it to be generalised across different types of operating systems and log formats. However, this study focuses explicitly on Windows Event Log (EVTX) files and does not exhaustively test the method's applicability to various operating systems or diverse log formats. While converting EVTX files into text format is a step toward broader applicability, the full exploration of the method's effectiveness across various environments and data structures is beyond the scope of this research. Future studies could extend this work by rigorously evaluating the method across different operating systems and log formats to confirm its generalisability.

5.4. Contributions

State-of-the-art anomaly detection methods predominantly rely on AI techniques, lacking transparency. This heavy dependence on AI approaches has overshadowed the search for alternative methods that are more understandable.

The solution proposed in this research is innovative in that it leverages the textual content of log metadata without employing machine learning techniques, instead relying on statistical methods. While statistical methods have been applied in anomaly detection in the past, the novelty of this approach lies in applying a technique rooted in stylometry - a field traditionally associated with authorship attribution and text analysis. To the best of our knowledge, this method has not been previously utilised within the context of anomaly detection.

This novel approach offers several advantages: it is easily integrable due to its reliance on log messages and through its reliance on the textual content, the need for domain-expertise is lowered. More importantly, this method serves as a complementary tool within the broader anomaly detection framework. It allows for localisation of suspicious hosts, log types and time frames, providing a preliminary analysis to guide where more advanced, precise, and resource-intensive methods should be applied. This capability enables organisations to take precautionary measures quickly and lowers the barrier to adopting some level of protection, even if they lack the resources for more complex solutions.

Hence, the following are the contributions of this research:

1. **Introduction of a Novel Method:** While traditional anomaly detection methods have utilised statistical techniques, to the best of our knowledge, no existing method in the field has employed

statistical measures from stylometry, specifically those used in authorship attribution. This research utilises the Burrows Delta Method, a technique traditionally associated with stylometric analysis, as a novel approach within anomaly detection.

2. **Comparison of Burrows Delta Method Variants:** This study outlines and compares various adaptations of the original Burrows Delta Method, analysing their effectiveness in identifying outlying hosts and defining host behaviour. These comparisons aim to determine which variant most effectively serves as an indicator of potential attacks.
3. **Behavioural Summaries of Hosts:** The research provides detailed behavioural summaries of each host, categorising behaviours as common, unique, or distinctive. This allows security analysts to quickly assess typical versus atypical activities, facilitating rapid identification of potential threats.
4. **Temporal Analysis:**

   (a) **Clustering-based:** The study examines host clustering across different time periods.
   (b) **Behavioural-based:** The study examines host behaviour across different time periods

   This enables the identification of time intervals where significant behavioural changes occur. This analysis helps pinpoint the likely time of an incident and identifies pre-incident activities that may have led to a compromise or require in-depth analysis.

6

Attribute-Based Inter-Host Distances

This chapter focuses on the use of attributes extracted from metadata of the logs of the hosts to determine the inter-host distances and hosts' behaviour. These inter-host distances are utilised to identify the outlying host. The Euclidean distance, calculated based on these attributes, serves as the measure of inter-distances. The choice of Euclidean distance is motivated by its intuitive interpretation, robustness to scale, computational efficiency, compatibility with clustering algorithms, geometric interpretation, and well-established theory, rendering it an ideal selection for the attribute analysis. These calculated distances, serving as proxies for inter-distances, are crucial inputs for hierarchical clustering. This technique is employed to group hosts based on their inter-distances.

The attributes considered for clustering include:

- **Absolute Size**: The footprint of the hosts, particularly the log files' storage size measured in bytes.
- **Relative Size**: The host size in terms of the number of events recorded within the log files.
- **Log Distributions**: The distribution of the sizes of different types of logs for each host, capturing the spread of different log types across hosts.
- **Event ID Frequency**: The frequency of each event ID, providing insight into the occurrences of specific events within the host.

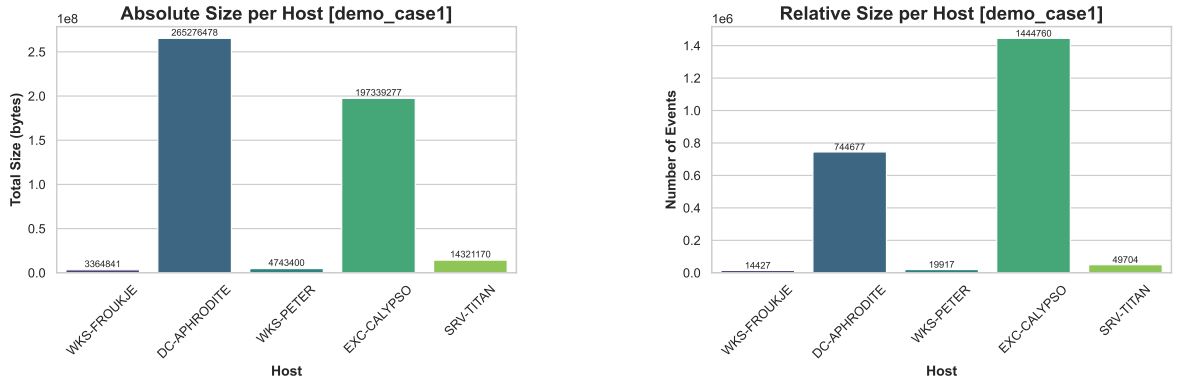
The absolute and relative sizes, alongside their distributions, provide an understanding of the resource utilisation patterns of the hosts. Hosts with similar workloads typically exhibit parallel log patterns. Consistency or divergence in sizes and log distributions across hosts sheds light on shared functionalities or characteristics among them. Moreover, the log distributions offer valuable insights into how hosts generate and handle events. By scrutinising the frequency and variety of logs produced by each host, unique behavioural patterns are unveiled. Furthermore, examining the frequency of event IDs, which denote specific host actions, provides deeper insights into their behavioural profiles. This is particularly true as hosts with similar functionalities tend to generate analogous types of events.

Thus, these attributes offer valuable insights into the hosts. The findings of these attributes are outlined in the following subsections: section 6.1 discusses the absolute and relative sizes, section 6.2 delves into the examination of log distributions and section 6.3 addresses the frequency of event IDs. Finally, section 6.4 summarises and compares the results from these attributes.

6.1. Absolute and Relative Size

This section analyses the clustering of hosts based on the inter-distances between their sizes within the Demo-Cases. The sizes of the hosts are evaluated in both absolute terms - denoting the storage capacity occupied by various log types in bytes - and relative terms, reflecting the number of events within each log file type. Subsection 6.1.1 examines the clustering behaviour based on the sizes of the hosts. Next, subsection 6.1.2 assesses the clusterings of the Demo-Cases on their capability to identify the outlying host - the host under attack within the Demo-Case. Finally, subsection 6.1.3 revisits the key findings, evaluating how effectively the absolute and relative size attributes identify outlying hosts and describe their behaviour.

Figure 6.1a and Figure 6.1b visually represent the cumulative byte sizes and total event counts of log types for each host within Demo-Case 1, offering insights into the distribution of data across hosts.

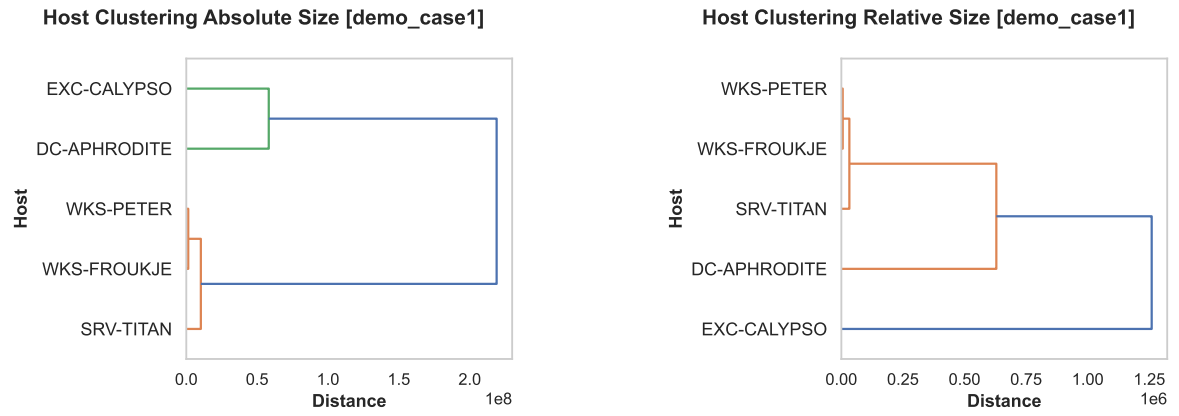


(a) Absolute Sizes

(b) Relative Sizes

Figure 6.1: Sizes of Demo-Case 1

Furthermore, Figure 6.2a and Figure 6.2b depict the clustering of hosts based on their absolute and relative sizes within Demo-Case 1, respectively. The corresponding sizes and clusterings for other Demo-Cases are provided in Appendix A Figures A.1 through A.6, offering a comparative perspective across different scenarios.



(a) Absolute Sizes

(b) Relative Sizes

Figure 6.2: Clustering based on sizes for Demo-Case 1

6.1.1. Clustering

Workstations and Server

The workstations and server clustering based on size - irrespective of whether expressed in terms of bytes or the number of events - within each Demo-Case demonstrate identical clustering outcomes. Due to their operational similarity they typically handle similar tasks and generate logs at a comparable rate and volume. Consequently, there is a direct correlation between the log size in bytes and the number of events logged, resulting in the same clustering for these hosts regardless of the size metric used. More specifically, within these size-based clusterings, the workstations WKS-Froukje and WKS-Peter consistently form the initial grouping in each Demo-Case. This grouping is due to the minimal disparity in the sizes of these hosts, as illustrated in Figures 6.1, and A.1 through A.3. Subsequently, SRV-Titan is clustered with the workstations, as its size differs slightly from their average size.

Domain Controller and Exchange Server

DC-Aphrodite and EXC-Calypso join at last when clustering based on size - irrespective of whether expressed in bytes or the number of events - within each Demo-Case due to their notably larger sizes. Due to their critical roles within the network, their higher frequency of logged events is expected. However, unlike the identical clusterings of the workstations and server across relative and absolute sizes, the exchange server and domain controller exhibit different clustering behaviours based on these size attributes. When clustering by absolute size, these hosts first cluster together and then with the remaining hosts. Conversely, they cluster as independent groups when clustered by relative size, with the domain controller joining first, followed by the exchange server. This discrepancy is due to the distinct nature of their logging behaviours stemming from their unique operational roles. The exchange server handles a high volume of email traffic, generating numerous small events related to each email transaction. These events quickly accumulate in number, resulting in a larger log size when counted by events. In contrast, the domain controller logs events related to network authentication and policy enforcement, which involve fewer but larger events. This difference in logging behaviour arises because the volume of logs and the frequency of events do not correlate similarly for these two hosts, leading to different clustering outcomes depending on the size attribute used. The exchange server and domain controller generate large log files, causing them to cluster based on absolute size. Their large log sizes are due to either numerous small events - exchange server - or fewer but larger events - domain controller. The nature of these events makes a significant difference: the exchange server's numerous small events cause it to stand out when clustering by relative size, whereas the domain controller, with fewer but larger events, clusters more closely with workstations and servers that log frequent but smaller events related to routine operations. As a result, the exchange server and domain controller show distinct groupings based on whether the size attribute is expressed absolutely or relatively.

6.1.2. Identifying Attacks

Demo-Case 1 & 2: RDP Brute Force Attack

Based on its operational role, DC-Aphrodite handles authentication requests, and, thus, generates events for each RDP login attempt, whether successful or unsuccessful. Hence, the volume of DC-Aphrodite's logs will depend on the number of login attempts. Yet, since the amount of events under normal conditions on DC-Aphrodite is unknown, there is no evidence that the great size of the domain controller is a consequence of the RDP attack. Moreover, the size attribute does not reveal that the greatest amount of events are caused by authentication attempts.

Demo-Cases 3 & 4: ProxyShell Abuse

ProxyShell abuse attacks exploit vulnerabilities in Exchange servers, leading to a significant increase in the size of EXC-Calypso's logs. However, even under normal conditions, the Exchange server is expected to generate a large number of events due to its critical role in managing email communication and user authentication within the network. The size attribute alone does not indicate whether the elevated number of events is due to routine access and authentication attempts, error logs, warnings related to the exploitation attempt, or logs generated by security measures. Therefore, the presence of a ProxyShell abuse attack cannot be confirmed based solely on the size attribute.

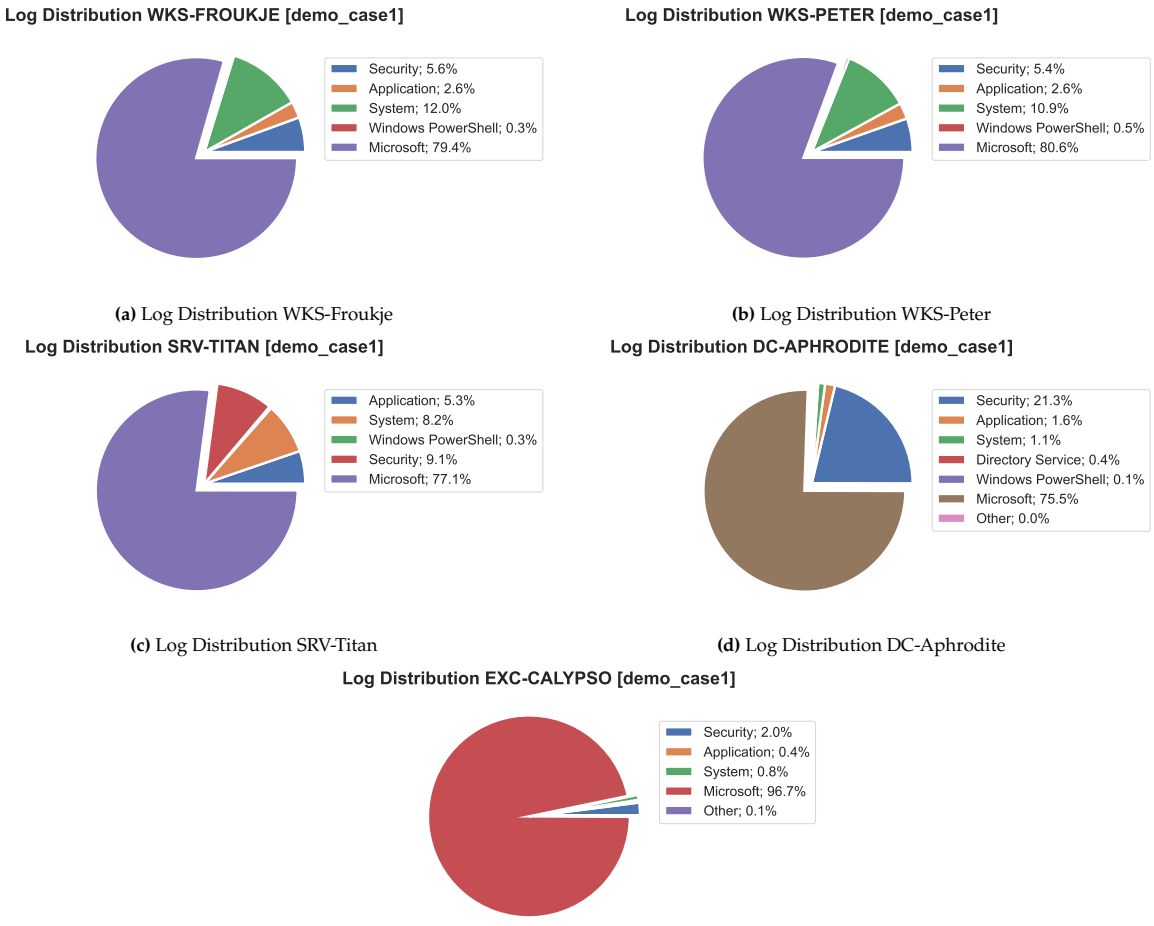
6.1.3. Summary: Absolute and Relative Size

Regardless of whether the hosts are clustered based on absolute size - volume in bytes - or relative size - number of events -, the workstations WKS-Froukje and WKS-Peter consistently form the initial cluster in each Demo-Case. SRV-Titan then joins, exhibiting a slight distance from the workstations. However, DC-Aphrodite and EXC-Calypso vary between the absolute and relative size clusterings. In absolute size clustering, both DC-Aphrodite and EXC-Calypso are identified as outliers, whereas in relative size clustering, EXC-Calypso is the primary outlier, followed by DC-Aphrodite.

These clustering results highlight EXC-Calypso and DC-Aphrodite as outlying hosts, likely reflecting their inherent operational roles rather than identifying them as distinct due to attacks. Exchange servers and domain controllers typically generate more logs due to their critical functions in the network. Thus, the clustering results may capture these operational characteristics rather than exclusively identifying issues like RDP brute force attacks, ProxyShell abuse, or lateral movements.

6.2. Log Distribution

This section analyses the clustering of hosts based on the distribution of log types within each host in the Demo-Cases. A host's log distribution delineates the proportional representation of each log type within it. Subsection 6.2.1 examines the clustering behaviour of the hosts based on their log distributions. Then, subsection 6.2.2 evaluates the ability of these clusterings to identify the outlying host under attack within the Demo-Case. Finally, subsection 6.2.3 revisits the key findings, assessing how effectively the log distribution attribute defines their behaviour and identifies the attacks.



(a) Log Distribution WKS-Froukje

(b) Log Distribution WKS-Peter

(c) Log Distribution SRV-Titan

(d) Log Distribution DC-Aphrodite

(e) Log Distribution EXC-Calypso

Figure 6.3: Log Distribution of Demo-Case 1

Figures 6.3a through Figure 6.3e illustrate the log distributions for each host within Demo-Case 1. Here, the logs related to Microsoft services have been aggregated into a single category labelled 'Microsoft' to enhance readability. The detailed distributions of the Microsoft logs can be found in Appendix B.3 Table B.1. The log and detailed distributions are shown in Appendix B.1 and B.2 for the remaining Demo-Cases, with Figures B.1 to B.3 and Tables B.2 to B.4. Additionally, Figure 6.4 shows the clustering of hosts based on their log distributions in Demo-Case 1. Appendix B.2, Figures B.4a through B.4c depict corresponding clusterings for other Demo-Cases.

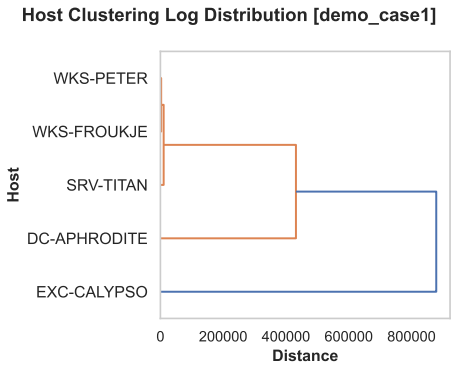


Figure 6.4: Clustering based on log distribution for Demo-Case 1

The clustering sequence of the hosts based on their log distribution is consistent across all Demo-Cases. WKS-Froukje and WKS-Peter consistently form the primary cluster, followed by SRV-Titan, which joins with a minor distance. DC-Aphrodite joins with a significantly larger distance, and EXC-Calypso finally joins last. The distances between clusters are identical across Demo-Cases, with the exception of Demo-Case 1, where EXC-Calypso is notably closer to the other hosts compared to the distances observed in the other Demo-Cases. For a closer analysis, Figure 6.5 provides the complete cluster map, depicting the number of events within each log type.

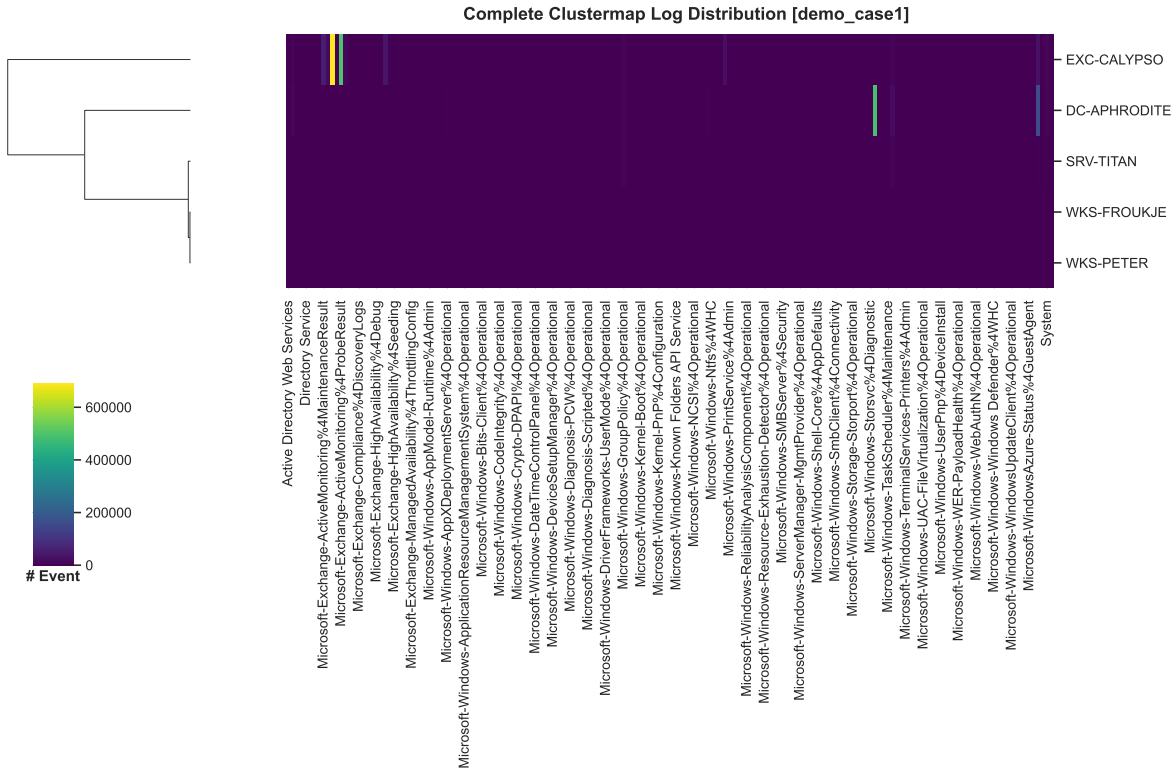


Figure 6.5: Complete clustermap encompassing all log types for Demo-Case 1

6.2.1. Clustering

Workstations and Server

Figure 6.5 shows minimal disparities in the distribution of log types between WKS-Froukje, WKS-Peter, and SRV-Titan, thus rendering them nearly indistinguishable. However, upon zooming into the core logs, namely Application, System and Security, as demonstrated in Figure 6.6, a subtle variance is discernible between these log types of SRV-Titan compared to the workstations. In contrast, such variance is absent between the two workstations themselves. This disparity elucidates why WKS-Froukje and WKS-Peter were initially grouped. In contrast, the slight differences, especially in these core logs, for SRV-Titan suggest it is performing additional or slightly different tasks compared to the workstations.

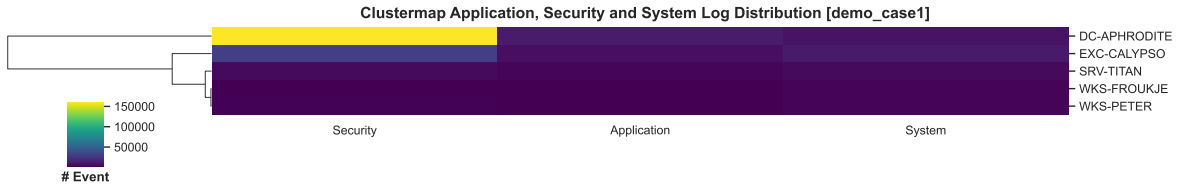
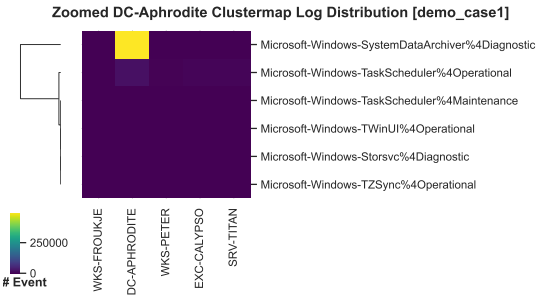


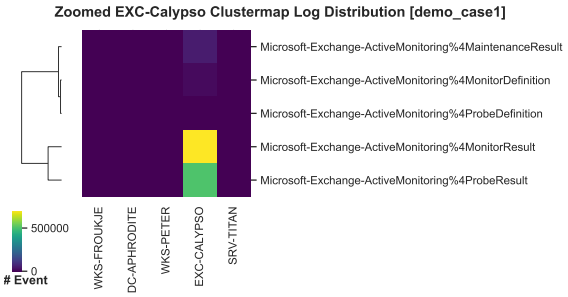
Figure 6.6: Clustermap focused on application, system, and security logs

Domain Controller and Exchange Server

DC-Aphrodite and EXC-Calypso stand out prominently from the workstations and the server due to their critical functions, which inherently prioritise security. This emphasis on security leads to a higher frequency of security events on these systems, as shown in Figure 6.6. Furthermore, Figure 6.5 reveals significant deviations in specific log types for the domain controller and exchange. Figure 6.7 zooms into the deviating log types of these hosts.



(a) Clustermap focused on outlier logs for DC-Aphrodite



(b) Clustermap focused on outlier logs for EXC-Calypso

Figure 6.7: Clustermap focusing on the oulying log types based on the log distribution for Demo-Case 1

Figure 6.7a illustrates that the distinctiveness of DC-Aphrodite primarily stems from the log type `Microsoft-Windows-SystemDataArchiver%44Diagnostic`. This log type typically stores information related to data archiving and system diagnostics. Data archiving ensures that logs are retained for long-term analysis and regulatory requirements. System diagnostics are essential for identifying and addressing potential issues before they affect the entire network, ensuring smooth operation and minimising downtime. While important, other hosts, such as workstations, servers, and Exchange servers, do not bear the same level of network-wide responsibility and centralised management functions as the domain controller. Therefore, the rise of events in this log type highlights the unique and critical role of DC-Aphrodite, setting it apart from other hosts in the clustering analysis.

Next, Figure 6.7b shows that the distinctiveness of EXC-Calypso primarily stems from the log group `Microsoft-Exchange-ActiveMonitoring`, specifically the `MonitorResult` and `ProbeResult` types. The `ActiveMonitoring` service continuously scrutinises various components and services to ensure optimal performance and reliability. This monitoring involves running predefined probes and responders to check the health of these components. Upon detecting an issue, the responders kick in.

`ProbeResult` logs record the outcomes of these probes, including success or failure and relevant metrics. `MonitorResult` logs capture aggregated monitoring results, trends, and responder actions. Given the critical role of the exchange server in handling email traffic and communication services, regular health checks are required to maintain high availability.

6.2.2. Identifying Attacks

Demo-Case 1 & 2: RDP Brute Force Attack

The cluster maps of core log types in Demo-Case 1 and 2 show a significantly higher number of events in the `Security` log for DC-Aphrodite than other hosts. The `Security` log is crucial for detecting RDP brute force attacks, as it records both successful and unsuccessful logon attempts, which are expected to spike during such attacks. Additionally, it logs activities attackers might use to cover their tracks, such as privilege escalation attempts. Yet, the log distribution only indicates an elevated number of `Security` log events, not the type of events within this log. The elevated number of events could be attributed to operational differences rather than malicious activities. The domain controller inherently maintains a higher baseline of `Security` logs due to its authentication, policy enforcement, and account management responsibilities. During an RDP brute force attack, this baseline would be significantly surpassed. Therefore, while there is a higher number of `Security` log events on DC-Aphrodite, the difference alone does not conclusively indicate an RDP brute force attack without considering the volume of `Security` logs under regular operating conditions and the type of activities within this log.

Demo-Cases 3 & 4: ProxyShell Abuse

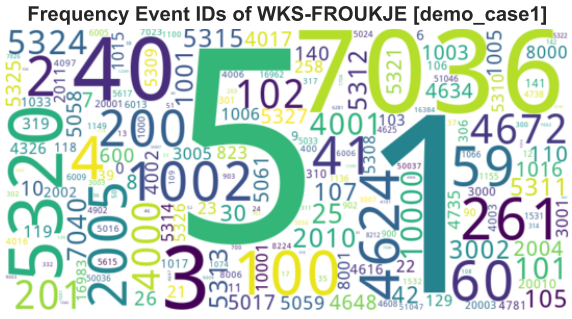
Demo-Case 3 and 4 both show a significant deviation for the log types `Microsoft-Exchange-ActiveMonitoring%4MaintenanceResult` and `Microsoft-Exchange-ActiveMonitoring%4ProbeResult` for EXC-Calypso compared to the other hosts. During a ProxyShell exploitation, an elevated number of events within these log types could be caused by disruptions or performance degradation of the Exchange server's services. Yet, the heightened number of events within these logs could equally likely be attributed to its operational responsibility to ensure reliability and performance. Hence, without knowing the number of events in these log types during normal conditions and the type of events within these log types, the high number of events within these log types provides insufficient evidence of ProxyShell abuse.

6.2.3. Summary: Log Distribution

Upon clustering based on the log distributions of the hosts, each Demo-Case highlights EXC-Calypso as the main outlier, followed by DC-Aphrodite. The workstations form the initial cluster. Next, SRV-Titan joins them with minimal disparity, with slight variations primarily in the core logs, especially the security log. The domain controller and exchange server subsequently join as independent groups, both showing significant disparity. This disparity of DC-Aphrodite mainly arises due to the `Security` and `Microsoft-Windows-SystemDataArchiver%44Diagnostic` logs, whereas the logs primarily causing the dissimilarity of EXC-Calypso belong to the `Microsoft-Exchange-Active Monitoring` group.

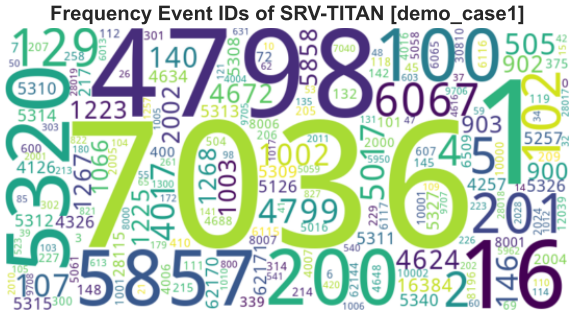
6.3. Event ID Frequency

This section delves into the clustering of hosts based on the frequency distribution of event IDs across various log types within each host. Event IDs encapsulate specific occurrences and actions within a system, varying from login attempts to file accesses and system errors. By examining the frequency of these event IDs, valuable insights are gained into the behaviour of individual hosts. Subsection 6.3.1 examines the clustering behaviour of the hosts based on their event ID frequencies. Then, subsection 6.3.2 evaluates the ability of these clusterings to identify the outlying host under attack within the Demo-Case. Finally, subsection 6.3.3 revisits the key findings, assessing how effectively the event ID attribute defines their behaviour and identifies outlying attacks.

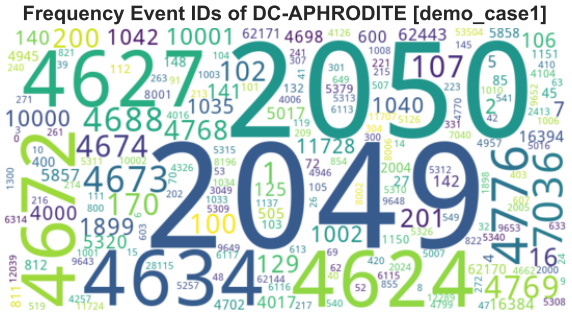


(a) Event ID Frequencies WKS-Froukje

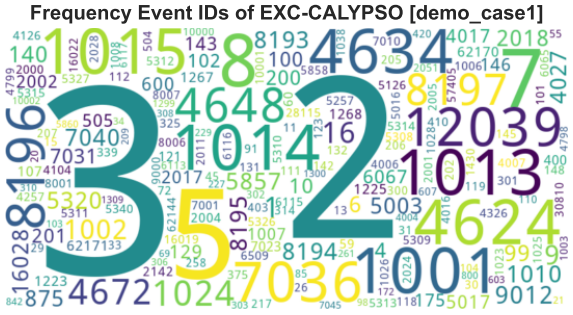
(b) Event ID Frequencies WKS-Peter



(c) Event ID Frequencies SRV-Titan



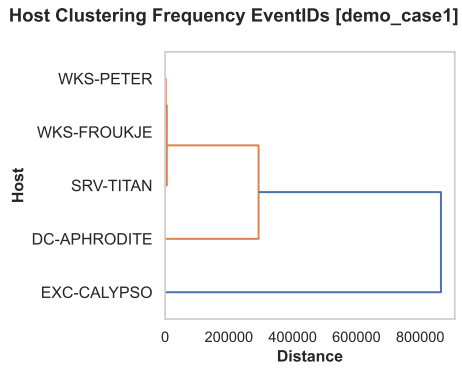
(d) Event ID Frequencies DC-Aphrodite



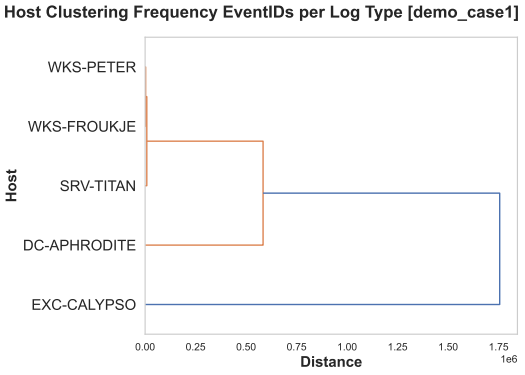
(e) Event ID Frequencies EXC-Calyso

Figure 6.8: Frequency of the Event IDs of Demo-Case 1

Figures 6.8a through 6.8e visually represent the event ID frequencies across all log types for each host in Demo-Case 1, serving as powerful tools for understanding the overall event ID distribution across hosts. The event ID frequencies for the remaining Demo-Cases are represented in Appendix C.1, Figures C.1 to C.3.



(a) All Log Types



(b) Per Log Type

Figure 6.9: Clustering based on frequency of the Event IDs

Figure 6.9 illustrate the clustering of hosts within Demo-Case 1 based on their event ID frequencies. Here, Figure 6.9a showcases host clustering considering event ID frequencies across all log types, providing a broad view ideal for identifying system-wide issues efficiently. Conversely, Figure 6.9b focuses on clustering hosts based on event ID frequencies per log type, offering insights into issues unique to specific log types. The clusterings based on the event ID frequencies across all log types and per log type of the remaining Demo-Cases can be found in Appendix C.2, Figures C.4 through C.6.

Clustering based on event ID frequencies across all log types streamlines analysis and runtime efficiency. However, this approach may overlook nuanced issues specific to certain log types. In contrast, clustering by event ID frequencies per log type offers more profound insights into the behaviour of individual components within hosts. Nevertheless, Figures 6.9 and C.4 to C.6 reveal consistent clustering regardless of the approach. Thus, whether considering all logs collectively or per log type, the clustering remains consistent, emphasising the systemic nature of event ID occurrences within the system.

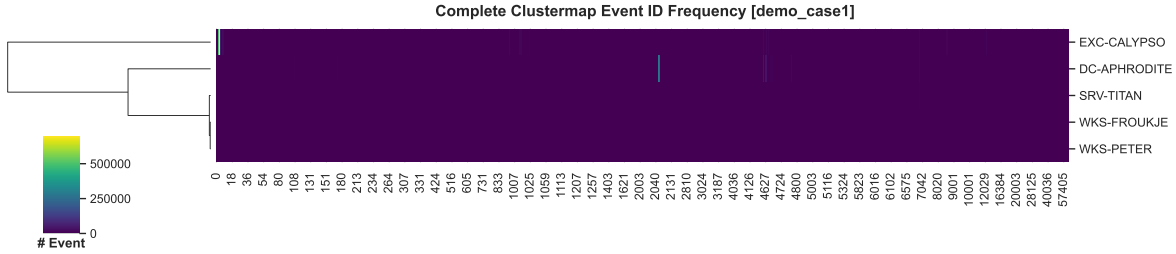


Figure 6.10: Complete clustermap encompassing all event IDs for Demo-Case 1

Upon examining the complete clustermap in Figure 6.10, which depicts the occurrence of each event ID, it becomes evident that the identical clusters are due to the dominance of specific event IDs, particularly `2`, `3`, `4`, `2049`, and `2050`. These event IDs have notably high frequencies compared to others, disproportionately influencing distance computations because Euclidean distance is sensitive to data scale. These dominant event IDs are associated with a specific log type: event IDs `2`, `3`, and `4` are tied to the `ActiveMonitoring` group log, while event IDs `2049` and `2050` are linked to the `Microsoft-Windows-SystemDataArchiver%4Diagnostic` log. Consequently, their presence across different log types does not impact the clustering algorithm. Therefore, whether clustering is based on all log types simultaneously or per log type, the inclusion of these dominant event IDs leads to consistent clustering outcomes unaffected by their distribution across log types.

6.3.1. Clustering

Workstations and Server

In the clustermap depicted in Figure 6.10, uniformity is present among the workstations WKS-Froukje, WKS-Peter, and the server SRV-Titan. However, a closer examination of the core log types - `Application`, `System`, and `Security` - in Figure 6.11 reveals nuances. The comparison between WKS-Froukje and WKS-Peter shows minimal disparity, validating their consistent grouping. Conversely, SRV-Titan exhibits a slightly elevated frequency of event ID `7036`. This slight deviation, coupled with similar frequencies among the remaining event IDs, results in SRV-Titan clustering closely with the workstations. Event ID `7036` indicates a change in service state due to user actions, system shutdowns, or errors. Since servers typically host a more extensive array of services than workstations, the increased occurrence of this event ID on SRV-Titan is expected.

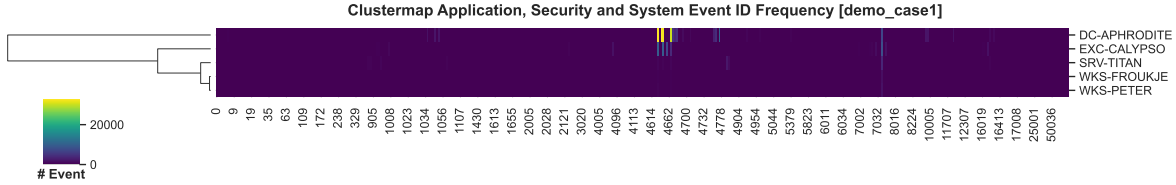
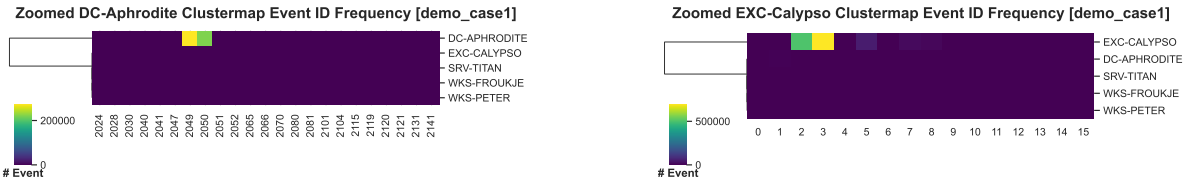


Figure 6.11: Clustermap focused on application, system, and security logs in Demo-Case 1

Domain Controller and Exchange Server

DC-Aphrodite joins as an independent group alongside the workstations and server, followed by EXC-Calypso. Figure 6.12 highlights the most significant deviating event IDs for the domain controller and exchange server.



(a) Clustermap focused on outlier event range (2024 – 2141) for DC-Aphrodite

(b) Clustermap focused on outlier event range (0 – 15) for EXC-Calypso

Figure 6.12: Clustermap focusing on the oulying event IDs of DC-Aphrodite and EXC-Calypso in Demo-Case 1

Figure 6.12a shows that DC-Aphrodite primarily diverges in the frequency of event IDs `2049` and `2050`. Here, event ID `2049` indicates a missing or corrupted component required by the Windows Installer service, while event ID `2050` signifies an attempt to repair the missing component encountered during installation issues. The elevated frequency of these event IDs on DC-Aphrodite can be attributed to the domain controller's crucial role, which often receives updates and patches, leading to more frequent installation or repair attempts for components.

Next, Figure 6.12b shows that EXC-Calypso primarily diverges in the frequency of event IDs `2`, `3`, and `4`. These event IDs indicate issues related to installing, modifying, or removing software components due to an inability to locate the installation package file or path. Given that the components required by the Windows Installer service are either missing or corrupted, as indicated by event IDs `2049` and `2050`, the service fails to locate the necessary files to complete the installation, resulting in event IDs `2`, `3`, and `4`. Consequently, the deviating event IDs on EXC-Calypso are directly correlated with those on DC-Aphrodite, all tied to the Windows Installer service's functionality in locating and installing packages.

6.3.2. Identifying Attacks

Demo-Case 1 & 2: RDP Brute Force Attack

During an RDP brute force attack, a significant increase in specific event IDs related to failed and successful login attempts would be expected. The event IDs `4624` - indicating a user successfully logged on - `4634` - indicating a user logged off - and `4776` - indicating that the domain controller validates credentials - all occur at significant rate on DC-Aphrodite. However, event IDs `4624` and `4634` occur at similar rates, but notably, event ID `4776` appears less frequently. A high rate of successful logons could indicate an RDP brute force attack; however, as the regular amount of logon events on the domain controller is unknown, there is no certainty whether this amount of logons can be classified as deviating high. The logoff events naturally follow successful logons, explaining why these two event IDs appear at similar rates. However, for each login attempt, the domain controller validates the credentials, generating event ID `4776`. Moreover, the number of `4776` events should typically be even higher, especially if there are many failed attempts before a successful login. Therefore, the disparity between loggon events and validation credentials, strongly indicates that the attacker is bypassing the standard authentication procedure.

Demo-Cases 3 & 4: ProxyShell Abuse

Within Demo-Case 3 and 4, EXC-Calypso exhibits a high frequency of event ID `4`. This event ID indicates that a server has received a Kerberos ticket from a client that cannot be decrypted. This usually points to an issue with the Kerberos authentication process. Yet, it does not serve as a direct indicator of ProxyShell exploitation. However, it could be idicative of the lateral movement. Here, the usage of compromised credentials by the attacker to access the resources of other hosts leads to the generation of this event. While frequent occurrences of event ID `4` can indicate lateral movements, they can also be related to routine maintenance, such as updates and installations, which are ongoing, as shown by the presence of event IDs `2` and `3`. Hence, if such activities were planned, this event alone cannot serve as a definitive factor of a lateral movement.

6.3.3. Summary: Event ID Frequency

When clustering the hosts based on the frequency of their event IDs, identical clusters are obtained, regardless of whether frequencies are considered across all logs or per log. This consistency arises from the clustering algorithm's sensitivity to outliers, with certain event IDs dominating due to their association with specific log types. Consequently, the clustering remains unaffected by the specification of the consideration of the log types.

Across all Demo-Cases, the clustering outcomes are consistent: the workstations form the initial cluster, followed closely by SRV-Titan with a slight disparity. This disparity is primarily due to SRV-Titan's more frequent changes in service states. Subsequently, DC-Aphrodite joins as an independent group, followed by EXC-Calypso. The domain controller and exchange server are significantly more involved in resolving issues related to applications and updates compared to other hosts, with the domain controller focusing on locating and repairing software components and the exchange server indicating the absence of related files.

Within Demo-Case 1 and 2, there is a disparity between the amount of loggon events and the amount of validations of the credentails. This strongly suggests that the standard authentication procedure is being bypassed. Moreover, in Demo-Case 3 and 4 the server frequenty encounters Keberos tickets that cannot be decrypted. Since the server is equally occupied with updates and installations, this behaviour does not indicate a lateral movement.

6.4. Summary

This section addresses the first sub-research question:

> **RQ1:** *How effectively do the attributes of the log metadata quantify the inter-host distances to define the behaviour of the hosts such that it can identify the outlying host?*

To this end, four attributes have been examined: absolute size, relative size, log distribution, and event ID. Regarding clustering, all these attributes, except the absolute size, produced identical results. Specifically, they initiated by grouping the workstations, followed by SRV-Titan with slight disparity. DC-Aphrodite then joined as an independent group, and EXC-Calypso lastly joined. The absolute size attribute differed, with DC-Aphrodite and EXC-Calypso forming a cluster before joining the workstations and SRV-Titan. Hence, each attribute effectively quantified the inter-host distance between the hosts, driven primarily by their distinct operational characteristics.

The size attributes do not provide insight into the underlying causes of the observed differences in log sizes. It is clear that the overall log sizes of the Exchange server and domain controller are significantly larger than those of the workstations and server. Furthermore, EXC-Calypso generates a higher volume of smaller events, while DC-Aphrodite logs fewer but larger events. However, the specific host behaviours responsible for these differences remain unclear.

The log distribution attribute revealed that SRV-Titan logged more security events than the workstations, DC-Aphrodite was mainly engaged in data archiving and diagnostics, and EXC-Calypso focused on continuous monitoring for optimal performance and reliability. Although the log distributions provided insight into which log types contribute to the distinctiveness of the hosts, accurately identifying the nature of the actions within the log types is essential for differentiating between normal operations and signs of malicious activity.

The event ID attribute provided insight into the activities of the hosts. More specifically, SRV-Titan frequently changes state. EXC-Calypso and DC-Aphrodite are mostly occupied in resolving issues of the Windows Installer Service. Yet, details about these activities are still missing. For instance, it remains unknown which specific components are encountering issues.

In short, the log distribution and event ID attributes offer some understanding of host behaviour; however, they necessitate extensive lookup work to determine specific activities and lack pivotal details on the activities of the hosts. This leads to the second sub-research question, investigated in the following section, which examines whether the textual content of logs can define host behaviour, which lowers the need for further investigation and reveals these missing details, leading to a more efficient and precise analysis of host behaviour.

7

Texual-Based Inter-Host Distances

This chapter focuses on the ability of the textual content extracted from metadata of the logs of the hosts to determine the inter-host distances, their behaviour and to identify attacks. The textual content is measured by the Burrows Delta method and its variants. Traditionally, the Delta method assesses document similarity to identify the author. In this context, however, it is adapted to evaluate the inter-host distance. Therefore, in this case, the collection consists of all the log types from each host, with each log type comprising the messages contained within that log type.

A delta score is calculated for each host pair and each log type. This score quantifies the similarity between a host pair based on a specific log type. A higher delta score between two hosts indicates greater dissimilarity in the content of that particular log type. Specifically, a log type with no common words between two hosts is assigned the highest delta score, signifying the maximum difference between the hosts in terms of that specific log.

After computing the delta score for each log type, Min-Max normalisation is applied to standardise the scores across all log types and hosts. This normalisation ensures that the delta scores and their n-gram contributions are scaled to a range between 0 and 1, preventing any single log type or n-gram from disproportionately influencing the overall score. Additionally, normalising enables a more effective comparison of the delta scores across different log types and hosts, facilitating a more balanced and meaningful analysis.

Chapter 3 provides a comprehensive overview of the mathematical foundation of the original Delta method (section 3.1) and its variants (section 3.2). It lays the groundwork for understanding the calculations and adaptations involved in applying these methods to the texual content. Following this theoretical framework, this chapter discusses the textual content's ability to quantify the inter-host distances, define the behaviour, and identify outlying hosts and attacks when measured using a delta method.

More specifically, section 7.1 clusters the hosts based on delta scores obtained from the applied method and n-gram size. Since a delta score is calculated for each log type of every host pair, to derive an overall similarity measure for each host pair, these individual delta scores across all log types are averaged into a single delta score. Hierarchical clustering is then applied using the single linkage method based on these average delta scores. By evaluating these results, the delta methods and n-gram sizes are shotlisted.

Next, section 7.2 assesses the shortlisted delta methods and n-gram sizes based on their ability to define the behaviour of the host. Here, the behaviour is expressed in logs and n-grams, and categorised into common, distinctive and unique. This section will provide insights into which method and size most accurately identify similar logs and n-grams. Then, upon deciding upon a delta method and n-gram size, section 7.3 discusses the behaviour of each host within Demo-Case 1 identified by this delta method and n-gram size. This will enhance the understanding of the hosts' behaviour. Subsequently, the ability to identify the attacks of this delta method and n-gram size will be evaluated in section 7.4. Finally, subsection 7.3.4 summarises the results.

7.1. Clustering

This section evaluates the clustering results of the delta methods and n-gram sizes to shortlist those that demonstrate the most promising clustering behaviour. These clustering results are derived from the delta methods, explicitly focusing on the impact of varying n-gram sizes on the clustering outcomes. The decision to experiment with n-gram sizes ranging from 1 to 5 is motivated by the need to balance granularity, computational efficiency, and the richness of contextual information in log message analysis. The resulting clusters and their analysis can be found in Appendix D.

To execute this evaluation, this section starts discussing the predicted clustering behaviour in subsection 7.1.1. This initial step will outline the expected clustering patterns of the hosts based on their operational roles, communication patterns, and resource utilisation. Then, after understanding these predictions, which provide a benchmark, the delta results are compared against this prediction in subsection 7.1.2. This comparison will highlight how closely each method's results align with the expectations. Next, subsection 7.1.3 compares the predicted clustering behaviour to the clustering results obtained using different n-gram sizes. This step will identify how changes in n-gram size affect the clustering outcomes and whether they enhance or detract from the alignment with predictions. Finally, an overview of the shortlisted delta methods and n-gram sizes is provided in subsection 7.1.4.

7.1.1. Prediction

Considering the clustering of different types of network devices - precisely, workstations, servers, an Exchange server, and a domain controller - their clustering behaviour can be predicted based on their roles, communication patterns, and resource utilisation.

Workstations - WKS-Froukje and WKS-Peter - typically communicate within the same subnet with similar frequency and use network resources consistently. Therefore, it is expected that workstations will cluster together, exhibiting the most minor delta score due to their uniform interaction and resource usage patterns.

The **server** and **Exchange server** - SRV-Titan and EXC-Calypso - are likely to communicate more frequently with each other than with the workstations and domain controller. This frequent communication is due to tasks such as user authentication and accessing user data. Additionally, they handle more data throughput and perform computationally intensive computations. Although the server is responsible for general purposes and the Exchange server handles emails, their functions are aligned closely enough to cluster together. However, they are expected to have a higher delta score than between the workstations due to their specific yet distinct operational roles.

The **domain controller** (DC) - DC-Aphrodite - is central to network security, managing user authentications and permissions across the network. Given that the DC's responsibilities align more closely with those of the servers than with the workstations, it is expected to cluster with the servers. While the DC communicates with all network devices, its interactions are more critical and less frequent than communication between workstations or servers. Due to its specialised role, the domain controller exhibits distinct utilisation patterns, leading to the expectation of displaying the highest delta score when clustered with the servers.
However, considering its unique and critical role, the DC could also be clustered as an independent group. Unlike servers and workstations that perform a wide range of tasks, the DC's functions are highly specialised and security-focused. This specialisation means its communication patterns and resource usage are fundamentally different from those of other network devices. The DC often operates in a more isolated manner, handling sensitive authentication and authorisation tasks, which do not require frequent or intensive interaction with other devices. This operational isolation and the distinct nature of its responsibilities justify treating the DC as an independent cluster, reflecting its singular importance and unique utilisation patterns within the network.

7.1.2. Results Delta Methods

Recall BD is based on the Manhattan Distance and z-transformation. BD constantly initially groups the workstations, followed by the cluster of the servers. Notably, applying Euclidean distance - QD - instead of Manhattan distance, using square root normalisation - ESD - instead of z-transformation, or introducing Eders Ranking Factor - ED - does not affect the grouping of the workstations and the servers. Specifically, square root normalisation and the ranking factor lead to identical clusterings. These observations align with the prediction that workstations are grouped together, servers are grouped together, and the inter-host distance of the workstations is smaller than that of the servers, thus exhibiting the lowest score of dissimilarity.

Moreover, applying cosine distance - CD - instead of Manhattan distance also groups the workstations and servers accordingly for all n-gram sizes, except for trigrams. However, applying cosine distance for the n-gram sizes other than trigrams results in a higher inter-distance for the workstations than the servers. This clustering outcome does not align with the prediction that workstations would exhibit the lowest dissimilarity.

Standardising based on the Laplace distribution - LD - is expected to perform better since the n-gram distribution, up to tri-grams regardless of the host, adheres to the Laplace distribution, as shown in Appendix E Figures E.1 to E.5. The clustering outcomes are significantly influenced by the definition of the document collection. Four distinct definitions of the document collection have been examined, namely the two logs under consideration (LD1), all the logs of the two hosts under consideration (LD2), all the logs of the log type under consideration (LD3) and all logs (LD4). Formal definitions and the detailed results of the different definitions of the document collection can be found in Appendix D.3. LD1 and LD2 deviate from all predictions, rendering them unsuitable for accurately depicting the similarities of the hosts present in the network. However, LD3 and LD4, cluster the workstations together. Yet, besides the domain controller, the servers also cluster as independent groups, which does not align with the prediction. Despite their frequent communication, the server and Exchange server perform distinct tasks: the server handles general computational tasks, while the Exchange server is dedicated to email management. Treating them as independent groups after the workstations underscores their specialised roles and distinct operational functions. This potentially reflects a more granular view of their relationship and emphasises their specific functions while acknowledging their similarity in handling high data throughput and computational tasks. Nevertheless, clustering the server and Exchange server as seperate groups risk overstating their operational differences. While they perform distinct functions, their frequent interactions and shared responsibilities might be better represented by clustering them together initially.

7.1.3. Results N-gram Sizes

Considering only the variants in which the workstations and servers cluster, the most suitable n-gram size can be determined based on the clustering behaviour of the DC:

- **Unigrams:** Regardless of the variant, the DC clusters as an independent group. While this clustering outcome aligns with the predictions, the usage of unigrams may not be suitbale due to their, among others, lack of contextual information and limited representation of relationships.
- **Bi- and Tri-grams:** For most variants, the DC clusters with the workstations, except for CD, for which this only holds under bigrams. This outcome renders bigrams and trigrams unsuitable n-gram sizes since - aligned with the prediction - the DC cannot be effectively grouped with the workstations based on its distinct operational role, communication patterns, and resource utilisation. Grouping the DC with workstations would not accurately reflect these differences and would misrepresent the similarity of these network devices.
- **4- and 5-grams:** The methods show variation in clustering the DC as an independent group (BD-4, QD-4, QD-5) or with the servers (BD-5, CD-4, CD-5, E(S)D-4, E(S)D-5); both predicted clustering outcomes. The choice between 4-grams and 5-grams for clustering hinges on the desired level of detail and complexity tolerance. While 4-grams emphasise granularity and specificity in capturing immediate patterns, 5-grams broaden the scope to include longer-term dependencies and predictive insights. Ultimately, the decision should align with the need to balance detailed understanding with comprehensive network visibility.

Regarding LD3, under unigrams, they also cluster the workstations and servers as primary clusters, with the inter-distance among the workstations being smaller than among the servers. For bigrams and up to 5-grams, they group the servers and the DC as independent clusters. In these cases, the order in which the servers join the workstations alternates, with the domain controller consistently being the last to join. Specifically, the server joins first under bigrams, while under trigrams through 5-grams, the Exchange server joins first. Both clustering patterns are plausible, indicating that when employing LD3, no specific n-gram size shows a clear preference over the others.

7.1.4. Overview Shortlisting

Based on the clustering outcomes and the evaluation of the delta methods and the n-gram sizes, the following methods and configurations have been shortlisted for further investigation:

- Delta Methods
  - **BD:** This is the original method, serving as a benchmark for comparing the impact of modifications introduced by the variant methods.
  - **QD:** This method alters the distance metric used in BD, allowing the examination of the effect of different distance metrics.
  - **ED:** This method introduces an additional factor to BD, enabling the assessment of the impact of incorporating an additional factor.
  - **ESD:** This method changes the transformation method used in BD, providing insights into the effect of different transformation techniques.
- N-gram Sizes: **Uni-**, **4**- and **5**-grams

These delta methods and n-gram sizes are shortlisted for two primary reasons: first, their varying impact on the original delta method, and second, their consistent alignment with the predicted clustering behaviours. They demonstrate their effectiveness and reliability in producing meaningful clusters.

Additionally, **LD3** is included in the shortlist due to its solid mathematical foundation and ability to produce justified clustering outcomes. Although LD3 and LD4 result in identical clusters, LD3 is preferred for its computational efficiency, making it a more practical choice for real-time settings.

7.2. Contributing Log Types and Words

This section determines the contributing logs and words of a particular host for three behaviour types, namely:

- **Common Behaviour:** This behaviour is defined by the lowest-scoring log types and words the particular host exhibits with the host with which it exhibits the lowest delta score. To determine which delta method and n-gram size most effectively define the common behaviour between WKS-Froukje and WKS-Peter, a comparative analysis is conducted on log types that show a zero score under one method but not the other. The results of these comparisons are provided in Appendix F.1.1, Table F.1, F.2 and F.3. Then, Appendix F.1.2 delves into the specific log types that are uniquely identified as exhibiting no dissimilarity for each method.
- **Distinctive Behaviour:** This behaviour is defined by the highest-scoring log types and words the particular host exhibits with the host with which it exhibits the highest delta score. Table F.4 in Appendix F.2.1 shows the log types that exhibit full dissimilarity between WKS-Froukje and DC-Aphrodite for each delta method and n-gram size. Then, Appendix F.2.2 delves into the specific log types that uniquely identify full dissimilarity for each method.
- **Unique Behaviour:** This behaviour is defined by the highest-scoring log types and words the particular host exhibits with the host with which it exhibits the lowest delta score. Table F.5 in Appendix F.3.1 shows the log types that exhibit full dissimilarity between WKS-Froukje and WKS-Peter for each delta method and n-gram size. Then, Appendix F.3.2 delves into the specific log types that uniquely identify full dissimilarity for each method.

These behaviour types will be examined for the delta methods and n-gram sizes that have been shortlisted in subsection 7.1.4. This examination will focus on the workstation WKS-Froukje within Demo-Case 1. Due to constraints, performing this analysis on all hosts is infeasible. However, WKS-Froukje is selected as it is expected to exhibit the most similarity with the other workstation, WKS-Peter, while showing the least similarity with DC-Aphrodite. This strong expectation makes WKS-Froukje an ideal candidate for a focused analysis as the predictive certainty is higher, allowing for more reliable conclusions. Nevertheless, this analysis helps to understand the reliability and sensitivity of each method in detecting subtle differences. Thus, sheds light on the effectiveness of capturing behaviour, thereby aiding in the selection of the most appropriate method for further investigations.

This analysis has led to the selection of **Eders Delta (ED)** based on the following key observations:

1. **QD assigns lower scores than BD and ED:** This suggests that BD and ED always perceives less similarity between hosts than QD. This discrepancy stems from the inherent nature of the distance measures used. BD and ED employ the Euclidean distance, which is more sensitive to larger discrepancies due to its use of squared differences. This sensitivity amplifies the impact of any significant frequency differences between n-grams. Conversely, QD uses the Manhattan distance, which aggregates differences linearly through absolute differences, resulting in a more stable and moderated assessment. Consequently, the Euclidean distance yields higher scores, especially in the presence of notable frequency variations, whereas the Manhattan distance provides a more balanced evaluation of similarity.
2. **Ranking-order:** BD, QD and ED exhibit identical ranking orders. The ranking refers to the sequence of n-grams that receive the lowest to the highest score.
3. **Inter-occurence ratio:** The inter-occurence ration is defined as the relative frequency of one word to another within the same host. Under identical ratios between the hosts, QD shows complete absence of discrepancies, whereas BD, ED and LD3 indicate a minor presence of discrepancy. Whereas, under mostly similar but not identical ratios, LD3 neglicts this slight variance completely. ED penalises this stronger than BD and QD, which assign insignificant scores, insuffiently refecting the dissimilarity.
4. **Absence of variation:** Here, all n-grams within each host exhibit the same frequency or there is only a single n-gram present. This absence of variation within the n-grams of each host is constantly accurately reflected in the ED and ESD scores. The additional ranking factor introduced by ED makes it a more reliable measure in cases where normalisation might introduce discrepancies. Moreover, since ESD employs square root normalisation, it directly compares the absolute frequencies. Hence, under identical frequencies, the absolute difference is zero, and the square root of zero remains zero, avoiding minor numerical errors introduced by statistical transformations within the z-transformation.
5. **Deviation overal trend:** BD, QD and ED penalise n-grams for which hostY exhibits a higher frequency than hostX whereas overall hostX exhibits higher frequencies. BD and ED are more sensitive to minor variations, making them better suited for identifying such deviations when subtle.

Upon evaluating the efficacy of unigrams, 4-grams, and 5-grams in defining behaviour, it has been concluded that unigrams lack the necessary surrounding context, limiting the understanding of host behaviour. For instance, comparing hosts based on their usage of the term *action* does not reveal the types of actions they differ in. Thus, while the frequency of the term *action* may be similar, it remains unclear whether the action is initiated or returned, let alone what type of action it entails.

Additionally, 5-grams do not provide significant essential information beyond that offered by 4-grams. Often, 5-grams result in the same grams as 4-grams, with the addition of terms like *service* or *id*, which do not substantially enhance the understanding of host behaviour. Empirical observations show that 5-grams frequently repeat grams with only minor deviations, such as a single word change, which does not add meaningful insight. This redundancy causes 5-grams to be quickly saturated with repetitive information. Consequently, top 4-grams offer more valuable information than 5-grams.

7.3. Host Behaviour

For Demo-Case 1, the common, distinctive, and unique behaviour in terms of logs and 4-grams for each host within the network, using the selected delta method, namely ED, are analysed in subsection 7.3.1, 7.3.2 and 7.3.3 respectively. Their behaviours are depicted in Figures 7.1, 7.2 and 7.3 respectively. In these figures, a fully black pair indicates that the commonality, distinctiveness, or uniqueness is bi-directional. In contrast, pairs with only a single black host signify that the trait is applicable solely to that host. Moreover, as the distinctive and unique behaviour define activities that most vary between the hosts, the arrow points to the host which exhibits the n-gram more often compared to the other host within the pair. A bi-directional arrow means that the n-gram appears an equal number of times in the hosts.

7.3.1. Common Behaviour

In this section the common behaviour of the hosts is analysed. The common behaviour represents the most usual activity of the host. Here, a significant amount of overlap between the hosts is evident - the non-overlapping entries are explicitly marked. The common 4-grams *windows resource exhaustion detector*, *resource exhaustion resolver received*, and *windows resource exhaustion resolver* all revolve around the concept of resource exhaustion and its detection or resolution. Hence, all hosts within the network have a strong emphasis on managing resource exhaustion, with specific mentions of detecting and resolving resource exhaustion issues. This suggests that the hosts monitor resource usage and takes steps to address resource exhaustion problems. Additionally, the presence of *whea successfully error sources* and *error record format version* within the logs suggests activities related to error reporting or logging. This indicates that all the hosts place importance on logging errors accurately and consistently.

Figure 7.1: Top 10 Common Behaviour in terms of logs and words for Demo-Case 1

Based on the non-overlapping common behaviour 4-grams, the workstations are primarily involved in document management, particularly with *Microsoft XPS document*s, and in application management, including monitoring their status and performance. They handle *session stop*ping events, engage in *version* control activities, and use *telemetry agent*s to collect data on application impacts.
In contrast, the servers focus on data collection and monitoring activities, implementing *fault-tolerant* mechanisms for memory management, applying *compatibility fix*es, and managing *driver*s.
Furthermore, the domain controller is involved in *register*ing and managing *user task*s, *creat*ing and managing *device key*s for *user*s, handling *MUI notification*s, *callback*s, and managing *MUI resource*s.

7.3.2. Distinctive Behaviour

In this section the distinctive behaviour of the hosts is analysed. The distinctive behaviour represents the most unsual activity of the host. Here, the distinctive behaviour of the workstations and server exhibit a significant amount of overlap unlike the exchange server and domain controller.

Figure 7.2: Top 10 Distinctive Behaviour in terms of logs and words for Demo-Case 1

The workstations and server differ mostly in areas include managing *local remote desktop session*s, handling *notification*s and *API callback*s, resolving *update errors*, continuously collecting and monitoring data, *troubleshooting* and diagnosing issues, and managing different *engine* versions or states. Additionally, WKS-Froukje differs in managing *exception*s and *rule*s through configuration *file*s or policies. In contrast, WKS-Peter and SRV-Titan vary in defining *server role*s related to session handling and *exception* management within *application*s.

Contrarily, DC-Aphrodite shows difference from the workstation WKS-Peter in its involvement in tasks such as managing *scheduled tasks* via *policies*, running *automated diagnostic*s, handling *user* data uploads in the *background*, applying structured *exception* and *rule* management within *application*s, and *detecting system maintenance* activities. The workstation more frequently (re-)connects to networks as indicated by

network interface network entered. Given that domain controllers are primarily responsible for authentication, and directory services, it is anticipated that their involvement in most of these tasks - except for user data uploads - would be greater than that of workstations. Moreover, due to their critical role, it is indeed expected for the domain controller to not frequenty change its network interface compared to a workstation since it requires a stable network environment.

Moreover, EXC-Calypso exhibits greater activity than DC-Aphrodite in their engagement in tasks like managing *package*s and handling *internet service* issues. Whereas, DC-Aphrodite is more active in performing and *detecting system maintenance* and managing network *transport* and *binding*s. These observations align with their main operational roles.

7.3.3. Unique Behaviour

In this section the unique behaviour of the hosts is analysed. The unique behaviour highlights the specific activities that distinguish each host from those with otherwise similar behaviors. Here, the unique behaviour of the workstations, servers and domain controller exhibit no overlap in terms of the 4-grams.

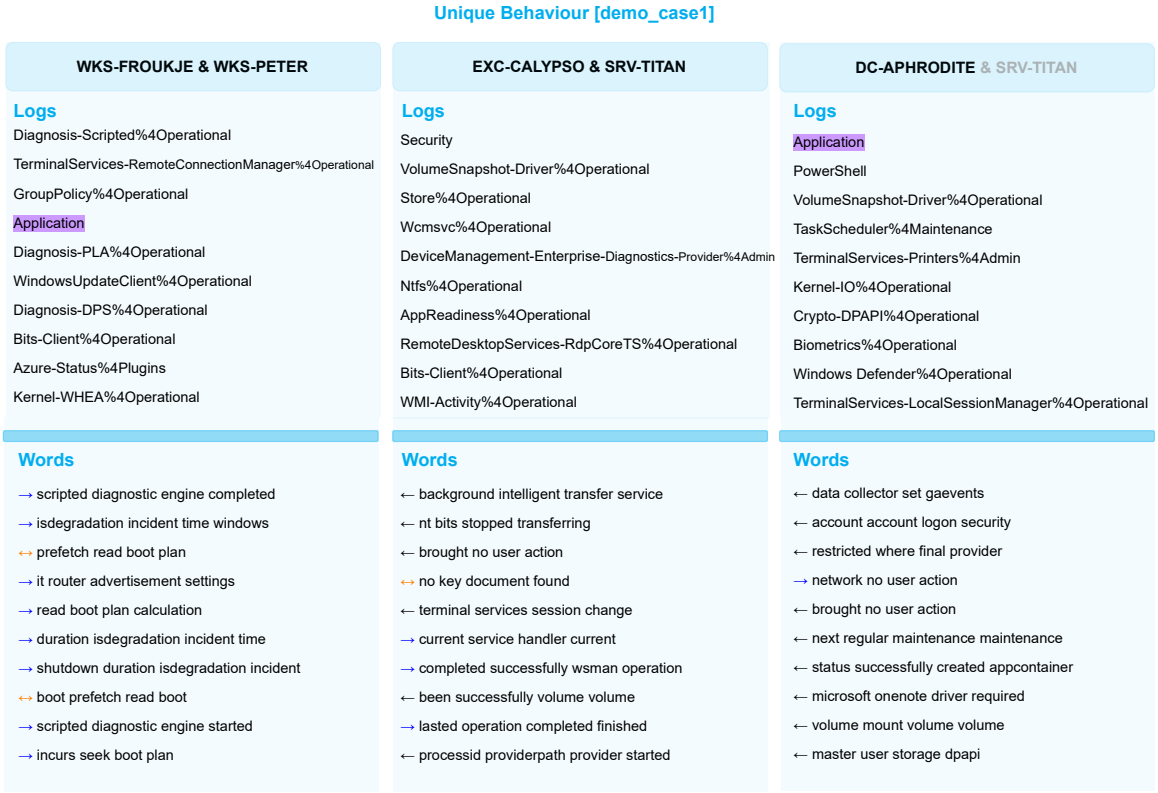


Figure 7.3: Top 10 Unique Behaviour in terms of logs and words for Demo-Case 1

WKS-Peter primarily differs from WKS-Froukje in being more engaged in running and *complet*ing *scripted diagnostic*s, tracking and managing *incident*s of system *degradation*. The greater involvement in these tasks might suggest that WKS-PETER either experiences more performance or stability issues than WKS-Froukje, or it might indicate that it is set up to be more proactive in diagnosing and addressing potential problems.

EXC-Calypso is more actively involved than SRV-Titan in managing *background transfer*s and *volume*s, and handeling remote *session*s. Whereas, SRV-Titan is more occupied in managing and completing operations related to web services (*WSMan*). The domain controller most differs from the server in its focus on *data collect*ion, *regular maintenance*, and managing *account security*.

7.3.4. Overview Behaviour

The selected delta method, Eders Delta, and the n-gram size of 4 have been applied to all hosts within Demo-Case 1 to characterise their behavior. Behaviour is categorised into three types: common, distinctive, and unique. Common behaviour represents the usual activities of a host, distinctive behaviour highlights the most unusual activities, and unique behaviour identifies activities that differentiate the host from its closest counterpart.

Workstations, unlike the other hosts within the network, are primarily engaged in document and application management. They show the greatest divergence from SRV-Titan, particularly in their management of remote desktop sessions, engine states, and troubleshooting activities. Among the workstations, the most notable difference lies in the focus of WKS-Peter on scripted diagnostics and degradation incidents unlike WKS-Froukje.

In contrast to other hosts within the network, servers are predominantly involved in data collection, monitoring activities, and addressing compatibility and driver issues. Among the servers, SRV-Titan is more involved in web services whereas EXC-Calypso in background transfers and remote sessions.

The domain controller distinguishes itself by being mainly engaged in managing user tasks and MUI (Multilingual User Interface) components, and creating keys. Unlike workstations, it is more involved in scheduling tasks via policies and, unlike server, it is less involved in managing packages and internet service issues.

Hence, examining these behaviour types across the entire period, gives insight into the activities of the hosts. This is supported by the fact that the identified activities of the hosts align with their operational characteristics.

7.4. Identifying Attacks

Figures D.9d and G.1 show the clusterings based on the ED scores of the 4-grams for each Demo-Case. Moreover, Appendix G.2 Figures G.2 through G.10 show the top 10 of each behaviour type for each Demo-Case.

Demo-Case 1 & 2: RDP Brute Force Attack
In Demo-Case 1, EXC-Calypso suffers from more internet service issues than DC-Aphrodite. These internet service issues are either indicative of network instability or targeted disruption. Without additional evidence, this observation is not conclusive of an attack, let alone an RDP brute force attack. Moreover, in Demo-Case 2 there is a strong commonality between the servers in applying group policies to Winlogon. This reveals that a tightly integrated environment where configurations applied to one server affects the other. This could be indicative of lateral movement, particularly if an attacker gained control of one server and is propagating changes or malware through group policies. This suspicion is supported by the high amount of biometric service failures in SRV-Titan occuring before the WinLogon group policy changes.

Demo-Cases 3 & 4: ProxyShell Abuse
In Demo-Case 3, all hosts except DC-Aphrodite show significant commonality in updating the Microsoft Defender Antivirus state. This coordinated behaviour suggests modifications to antivirus configurations across these hosts. By altering the antivirus state, the attacker could be attempting to disable protections, whitelist specific files, or modify settings to ensure their payloads remain undetected. Nonetheless, this depends on whether an antivirus update was planned. Moreover, in Demo-Case 4, SRV-Titan and DC-Aphrodite equally launch action-based servicing. Coordinated actions like action-based servicing could be indicative that the attacker is executing commands across multiple vital systems. However, upon a closer look, it becomes apparent that their commonality lies in *not* frequently launching action-based servicing.

7.5. Summary

This section addresses the second sub-research question:

> **RQ2:** *How effectively does the textual content of the log metadata quantify the inter-host distances to define the behaviour of the hosts such that it can identify outlying hosts?*

To determine the effectiveness of the textual content, the optimal delta method and n-gram size have been identified. For this purpose, the clustering results of the delta methods under n-gram sizes from 1 to 5 were evaluated. This evaluation was based on clustering prediction, expecting the initial cluster to consist of the workstations with minimal distance. Following this, the servers were expected to form a separate cluster, and finally, the domain controller would either cluster with the servers or join as an independent group. Based on this prediction, Burrows Delta, Quadratic Delta, Eders Delta, and Eders Simple Delta were shortlisted under n-gram sizes 1, 4, and 5, as they all aligned with the prediction. Linear Delta 3 was also included in the shortlist due to its mathematical foundation and ability to produce justifiable clusterings.

The shortlisted delta methods and n-gram sizes were further evaluated by examining the behaviour of the hosts, where the logs and n-grams define behaviour. This behaviour is categorised into common, distinctive, and unique. Common behaviour is defined by the most similar logs and n-grams when considering the host pair with which the host exhibits the most similarity; hence, it defines the most usual behaviour of the host. Distinctive behaviour is defined by the most dissimilar logs and n-grams when considering the host pair with which the host exhibits the least similarity; hence, it defines the most unusual behaviour of the host. Unique behaviour is defined by the most dissimilar logs and n-grams when considering the host pair with which the host exhibits the most similarity; hence, it defines the distinguishable behaviour of the host.

During the behavioral examination of the shortlisted methods, Eders Delta method outperformed the other delta methods, particularly in cases involving ranking orders, inter-occurrences, uniform frequencies and low variation in n-grams. Additionally, unigrams were found to not provide sufficient understanding due to their lack of surrounding context. Furthermore, 4-grams outperformed 5-grams as 5-grams included nonessential information and redundancy.

Eders Delta under 4-grams has been used to define the behaviour of hosts within Demo-Case 1. Here, the workstation are mostly engaged in document and application management. The servers, on the other hand, are primarily involved in data collection, monitoring and resolving issues. The domain controller is mainly busy managing user tasks and MUI components, as wel as creating keys. Where the exchange server mostly differs from the domain controller in managing installations, maintainence and network transport, the server mainly differs in user intervention levels and managing accoutn security and storage volumes.

In terms of identifying attacks based on the defined behaviour, the internet service issues identified in EXC-Calypso in Demo-Case 1 and the equal amount of launching action-based services by DC-Aphrodite and SRV-Titan did not serve as indicators of an attack. However, the application of Winlogon policies in combination with biometric service failures on SRV-Titan potentially indicate a lateral movement in Demo-Case 2. Moreover, an update of the Microsoft Defender antivirus state across all hosts, except DC-Aphrodite, served as an indicator in case of no planned antivirus update.

Thus, the textual content measured using Eders Delta with 4-grams reveals alternative behaviour of the hosts compared to the attributes, particularly event IDs. This is because the events containing no textual content do not contribute during this analysis. However, unlike the event IDs, this method does not require extensive lookup to derive insights. Yet, the specific time frame in which the incident occurred remains unknown. This leads to the third sub-research question, investigated in the following section, which examines the behaviour of the hosts using the textual content measured using Eders Delta with 4-grams within specified time frames. By focusing on specified time frames, the aim is to locate the timeframe of deviating behaviour. This approach could enhance the ability to pinpoint attack vectors and understand the sequence of events leading up to and following the attacks.

8

Temporal Behaviour

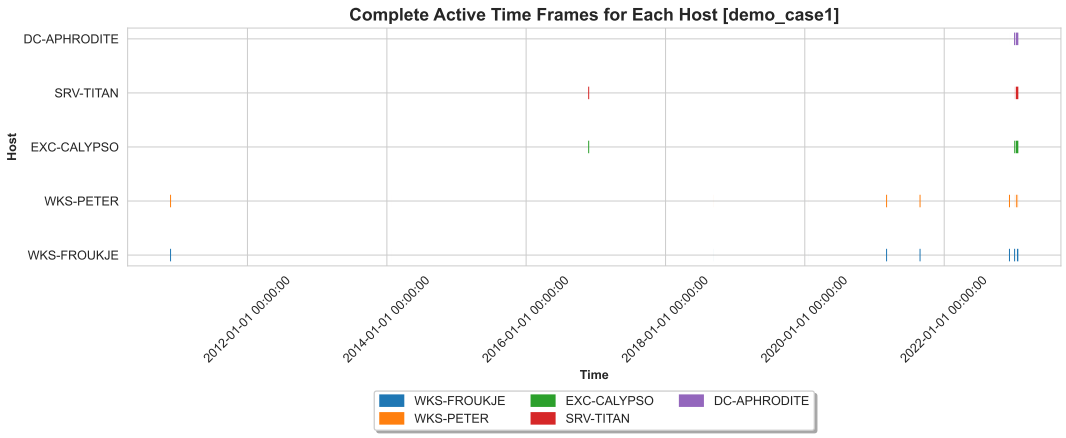
This chapter examines how the textual content extracted from the metadata of host logs, measured using Eders Delta with 4-grams, can define the behaviour of the host within each time frame.

Definition 1 *A time frame is a sequence of log messages within a maximum difference of x minutes from the most recent log message in the frame. Let L be the set of log messages l, where each log message l consists of the host h and log type t from which the log message originates, the date and time dt when the log message was recorded, and the actual message content m. A new time frame is created if the time difference between any new log message and the most recent log message in the current frame exceeds x minutes.*

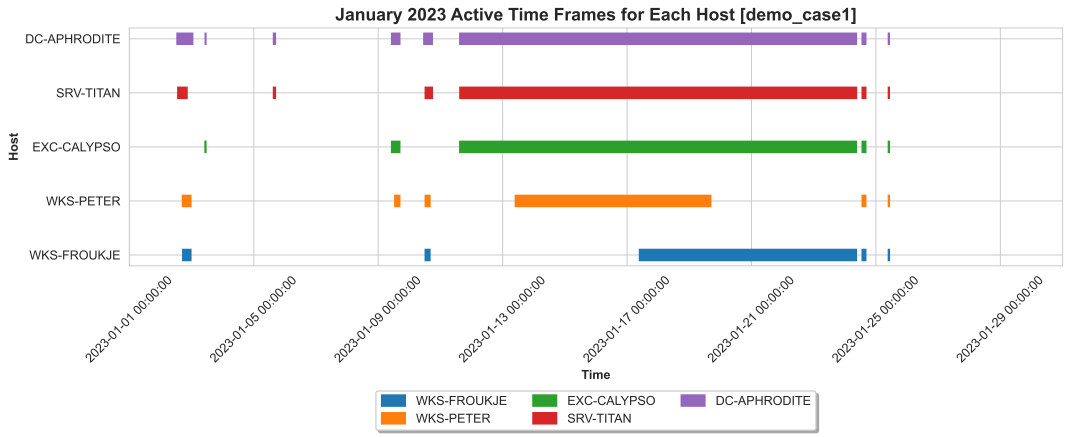
Selecting an appropriate time frame is crucial for capturing meaningful interactions and patterns when analysing behaviour within each time frame. Specifically, the choice of x significantly impacts the granularity of the analysis and the quality of the insights derived. Here, a 60-minute time frame ($x = 60$) has been selected for two main reasons: minimising disruption and maximising multi-host interactive time frames.

Firstly, a 60-minute time frame results in fewer but more comprehensive time frames. This minimises disruption and fragmentation of closely related activities. Shorter time frames excessively fragment the data, leading to a high number of time frames with minimal activity. This fragmentation obscures meaningful patterns and increases the complexity of analysis without adding substantial value. Additionally, fragmentation negatively impacts the feasibility of analysing behaviour within each time frame.

Secondly, shorter time frames often result in many time frames containing only a single active host. This sparse distribution of activity limits the utility of the delta method, which relies on interactions between multiple hosts to identify the host's behaviour. Longer intervals are more effective in capturing simultaneous host activities, which is crucial this method which depends on multi-host interaction.



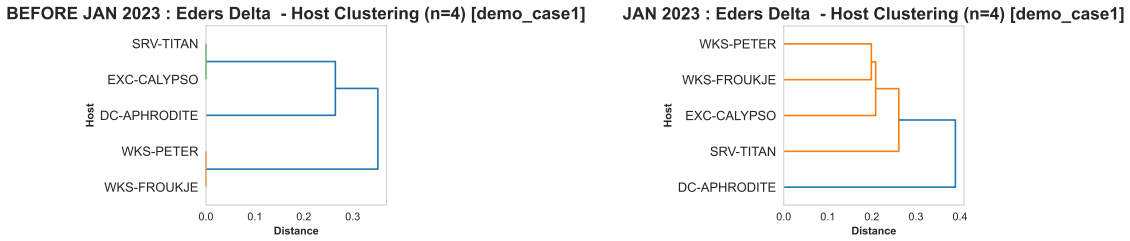
(a) Full active time period



(b) Time period January 2023

Figure 8.1: The 60-minute time frames ($x = 60$) of the hosts in Demo-Case 1

Figure 8.1a displays the active time frames over the entire period within Demo-Case 1. Here, the hosts are mainly active in January 2023. Figure 8.1b focuses on the time frames within this period. For this period, ten 60-minute time frames are obtained, of which only four exhibit activity of all the hosts.



(a) Before January 2023

(b) January 2023

Figure 8.2: Clustering the hosts in Demo-Case 1 based on only the data (before) January 2023

When clustering the hosts in Demo-Case 1 based on the log messages excluding those from 2023, as shown in Figure 8.2a, the resulting clustering is identical to that using all log messages over the entire period in Demo-Case 1 (Figure D.9d). The only notable difference is the reduced inter-host distances among the workstations and servers.

In contrast, clustering the hosts in Demo-Case 1 using only the log messages from January 2023, as shown in Figure 8.2b, yields a significantly different clustering. Although the workstations still form an initial cluster, their inter-host distance is four times larger. Additionally, the servers do not form a cluster but instead cluster as independent groups, with the Exchange server joining first, followed by the general server. Finally, DC-Aphrodite joins the cluster, which is consistently identified as the main outlier regardless of the selected period.

Clustering the servers based solely on January 2023 activities might be suboptimal compared to clustering over the entire period, even if other periods exhibit minimal activity. When clustering based exclusively on a high-activity period like January, the resulting clusters may not generalise well to the entire dataset. This is because the clusters are heavily influenced by the intense activity in this high-activity period, potentially overlooking broader, less frequent patterns that span the entire period. Consequently, low-frequency but potentially significant interactions that occur sporadically may be overlooked.

However, focusing on a high-activity period like January could also highlight anomalous or defective behaviour. The intense activity might reveal deviations from normal operations, flagging unusual patterns that warrant further investigation. These anomalies, though they might be overlooked in a more generalised clustering, could be critical for identifying security issues or operational irregularities.

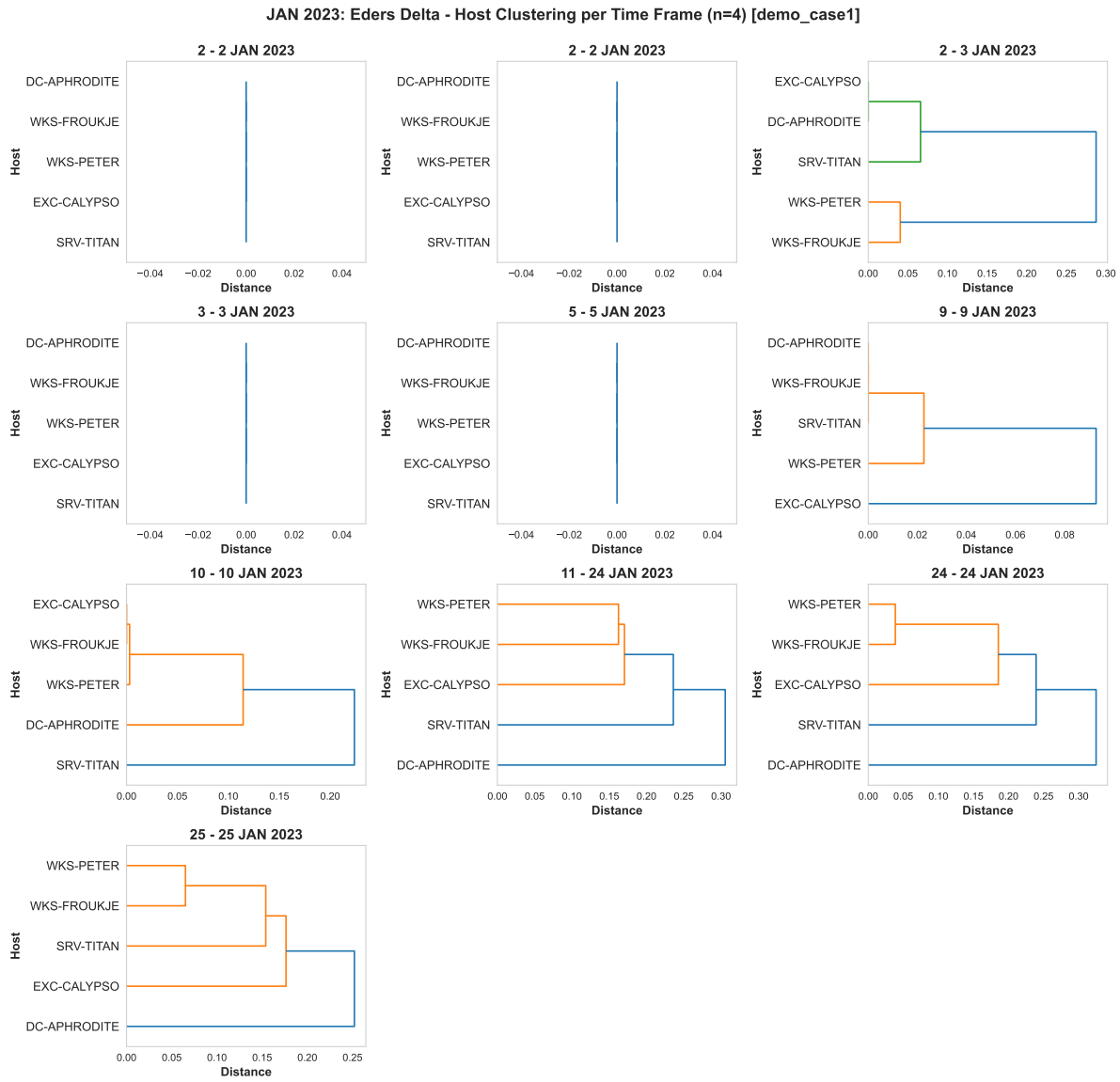


Figure 8.3: Clustering the hosts per time frame in Demo-Case 1 based on only the data of January 2023

Figure 8.3 shows the clustering of each 60-minute time frame in January 2023. It reveals that only the time frames of January 2-3, January 9, January 11-24, and January 24 include all hosts, with the last three being identical to the complete clustering for January 2023. Notably, the clustering for January 2-3 differs from the clustering observed before 2023, which may indicate potential malicious activity. To investigate this further, Figure 8.4 discloses the behaviour types of DC-Aphrodite, the host under attack, during these time frames of January 2023.

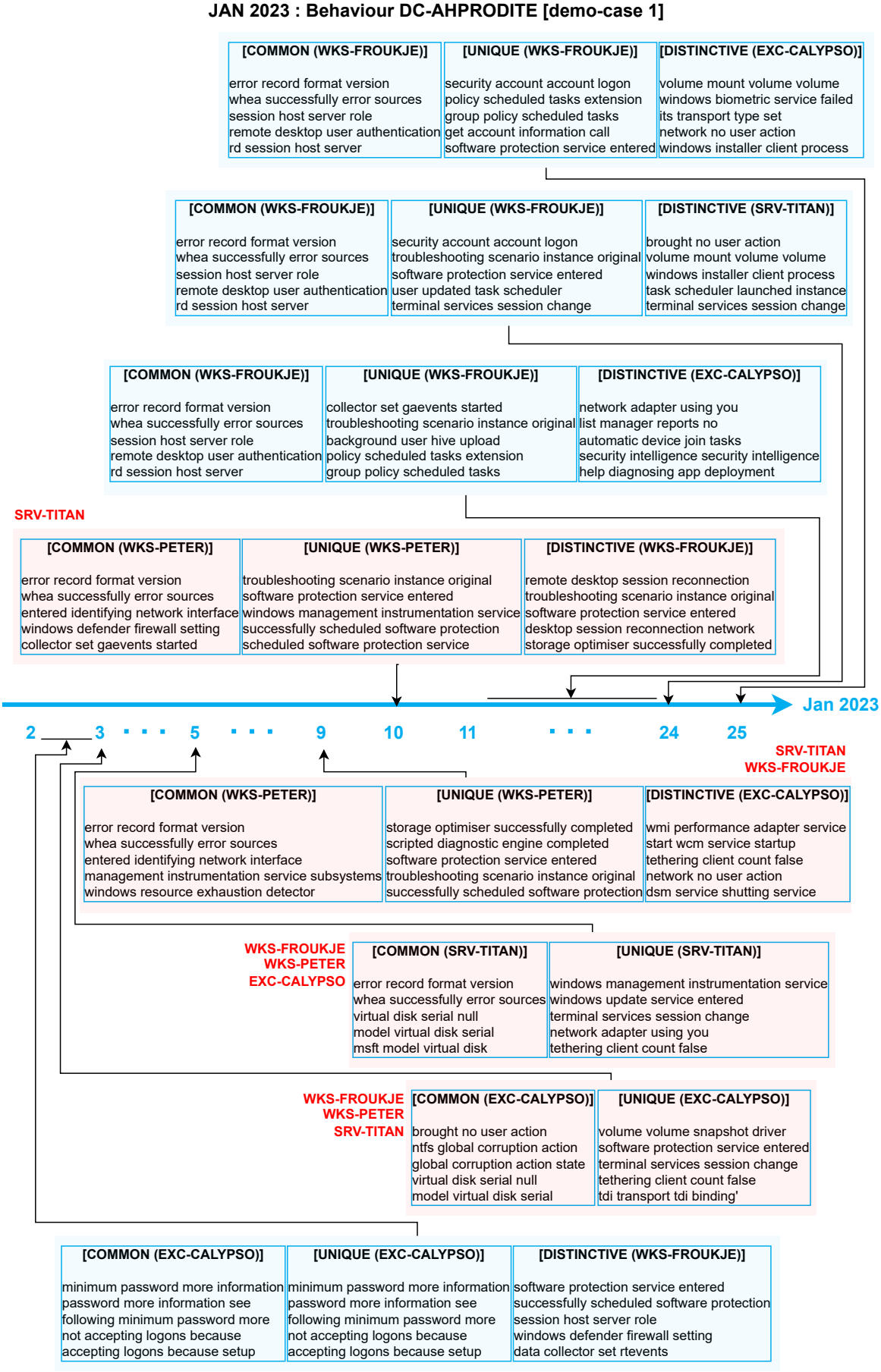


Figure 8.4: Behaviour of DC-Aphrodite within the active time frames of January 2023 in Demo-Case 1

Within 2-3 January, DC-Aphrodite and EXC-Calypso exhibit similarity in *not accepting logons because setup*. This 4-gram located in logtype `Microsoft-Windows-TerminalServices-LocalSessionManager%40perational`. In Listing 8.1 the log entries within this log of DC-Aphrodite and EXC-Calypso can be found, respectively.

Listing 8.1: `Microsoft-Windows-TerminalServices-LocalSessionManager%40perational`

```
DC-APHRODITE

        1/2/2023 12:15:35 PM, 34, 772, Verbose, Microsoft-Windows-TerminalServices-
            LocalSessionManager, DC-APHRODITE, Remote Desktop Services is not accepting
            logons because setup is running
        ...
        1/2/2023 12:15:13 PM, 34, 772, Verbose, Microsoft-Windows-TerminalServices-
            LocalSessionManager, DC-APHRODITE, Remote Desktop Services is not accepting
            logons because setup is running
        1/2/2023 12:15:13 PM, 32, 772, Verbose, Microsoft-Windows-TerminalServices-
            LocalSessionManager, DC-APHRODITE, Plugin RDSAppXPlugin has been successfully
            initialized

EXC-CALYPSO

        1/3/2023 1:53:03 AM, 34, 800, Verbose, Microsoft-Windows-TerminalServices-
            LocalSessionManager, EXC-CALYPSO, Remote Desktop Services is not accepting logons
             because setup is running
        ...
        1/3/2023 9:52:42 AM, 34, 800, Verbose, Microsoft-Windows-TerminalServices-
            LocalSessionManager, EXC-CALYPSO, Remote Desktop Services is not accepting logons
             because setup is running
        1/3/2023 9:52:42 AM, 32, 800, Verbose, Microsoft-Windows-TerminalServices-
            LocalSessionManager, EXC-CALYPSO, Plugin RDSAppXPlugin has been successfully
            initialized
```

The occurrence of *Remote Desktop Service not accepting logons because setup is running* on DC-Aphrodite and EXC-Calypso occur on different days and timings, only for less than a minute. Remarkably, this happens a few seconds after *RDSAppXPlugin has been successfully initialized*. The initialisation of RDSAppXPlugin is expected to temporarily disrupt services, hence, potentially explaining why RDS is not accepting logons. The short appearance on the hosts at different timings, could align with routine system updates or service restarts that happen asynchronously on different hosts. Hence, the classification of this behaviour depends on whether they occur during scheduled maintenance, updates, or configuration changes.

However, suspicion is raised when approximately 8 hours after these messages, WKS-Froukje, WKS-Peter and SRV-Titan show no activity. Specifically, WKS-Froukje remains inactive until January 9, WKS-Peter until January 5, and SRV-Titan until January 10. This sudden inactivity could suggest several possibilities: the hosts may have been shut down, isolated from the network, or compromised and rendered non-functional. The timing of the inactivity, followed by login rejectionss, raises the possibility that these hosts were targeted.

Since January 10, all hosts have resumed their activities; however, a new pattern has emerged. DC-Aphrodite and WKS-Froukje started showing the greatest similarity in activities, particularly in *remote desktop user authentication* present in `Microsoft-Windows-TerminalServices-Remote ConnectionManager%40perational`. In Listing 8.2 the log entries within this log of WKS-FROUKJE and DC-Aphrodite can be found, respectively

Listing 8.2: `Microsoft-Windows-TerminalServices-RemoteConnectionManager%40perational`

```
WKS-FROUKJE

        1/10/2023 4:20:56 PM, 1149, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Remote Desktop Services: User
            authentication succeeded:;;;;User: Froukje;;Domain: AKROPOLYS;;Source Network
            Address: 94.210.224.5
        1/10/2023 4:20:55 PM, 261, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Listener RDP-Tcp received a
            connection
```

```
        1/10/2023 4:20:40 PM, 261, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Listener RDP-Tcp received a
            connection
        1/10/2023 3:53:49 PM, 261, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Listener RDP-Tcp received a
            connection
        1/10/2023 3:53:49 PM, 261, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Listener RDP-Tcp received a
            connection
        1/10/2023 3:28:57 PM, 1149, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Remote Desktop Services: User
            authentication succeeded:;;;;User: Froukje;;Domain: AKROPOLYS;;Source Network
            Address: 94.210.224.5
        1/10/2023 3:28:57 PM, 261, 428, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, WKS-FROUKJE.Akropolys.nl, Listener RDP-Tcp received a
            connection
        ...

DC-APHRODITE

        1/10/2023 6:16:18 PM, 263, 404, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, DC-APHRODITE.Akropolys.nl, WDDM graphics mode is enabled
        1/10/2023 4:34:55 PM, 1149, 404, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, DC-APHRODITE.Akropolys.nl, Remote Desktop Services: User
             authentication succeeded:;;;;User: Aphrodite;;Domain: AKROPOLYS;;Source Network
            Address: 94.210.224.5
        1/10/2023 4:34:54 PM, 261, 404, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, DC-APHRODITE.Akropolys.nl, Listener RDP-Tcp received a
            connection
        ...
```

Both hosts show the same source network address, namely `94.210.224.5`, indicating that the RDP sessions are initiated from the same external IP. This might indicate that an attacker has successfully logged in to the workstation and is now trying to access the critical system - the domain controller - using the same credentials or access method. Especially, since the connections happen relatively close in time, with WKS-Froukje being accessed at 3:28 PM and 4:20 PM, and then DC-Aphrodite being accessed at 4:34 PM. This sequence could indicate an attacker moving laterally from WKS-Froukje to DC-Aphrodite after gaining initial access. Following these log entries, less than a day later, DC-Aphrodite logs the entry shown in Listing 8.3 .

Listing 8.3: `Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational`

```
DC-APHRODITE

        1/11/2023 2:24:17 PM, 1136, 2440, Verbose, Microsoft-Windows-TerminalServices-
            RemoteConnectionManager, DC-APHRODITE.Akropolys.nl, RD Session Host Server role
            is not installed.
```

The RD (Remote Desktop) Session Host is a role in Windows Server that allows to host Windows-based programs or the full Windows desktop for Remote Desktop Services clients. The message indicates that this role is not installed on DC-Aphrodite. Despite this, RDP connections and authentications were observed a day before, raising the possibility that RDP was used in an unauthorised or non-standard way, potentially by an attacker who may have enabled or leveraged RDP capabilities temporarily.

From January 11 until January 24 at 9:00 AM, DC-Aphrodite exhibited significant distinct activity from EXC-Calypso, particularly in activities related to *network adapter* configurations, system *report*ing, and *automatic device join tasks*. While the activities might seem routine administrative or maintenance tasks, certain aspects of these activities can also serve as early indicators of an impending RDP attack.

Listing 8.4: `System`

```
DC-APHRODITE

        1/24/2023 12:51:45 PM, 11, 4496, Verbose, Microsoft-Windows-Hyper-V-Netvsc, DC-
            APHRODITE.Akropolys.nl, Miniport NIC 'Microsoft Hyper-V Network Adapter #2'
            restarted
        1/24/2023 12:51:45 PM, 10, 4496, Verbose, Microsoft-Windows-Hyper-V-Netvsc, DC-
            APHRODITE.Akropolys.nl, Miniport NIC 'Microsoft Hyper-V Network Adapter #2'
            paused
```

```
        ...
        1/24/2023 12:51:35 PM, 11, 4, Verbose, Microsoft-Windows-Hyper-V-Netvsc, DC-APHRODITE
            .Akropolys.nl, Miniport NIC 'Microsoft Hyper-V Network Adapter #2' restarted
        1/24/2023 12:51:35 PM, 10, 4, Verbose, Microsoft-Windows-Hyper-V-Netvsc, DC-APHRODITE
            .Akropolys.nl, Miniport NIC 'Microsoft Hyper-V Network Adapter #2' paused
        ...
```

Listing 8.4 shows that the *Miniport NIC Microsoft Hyper-V Network Adapter #2* is being paused and then immediately restarted. This could be indicative of malicious behaviour to cover tracks. Pausing the NIC could temporarily disable network monitoring tools, and restarting it could allow the attacker to re-establish a network connection under different conditions. Since this involves a Hyper-V virtual network adapter, the suspicious behaviour could extend to virtual environments hosted on DC-Aphrodite. An attacker might has gained access to the Hyper-V environment, and might be trying to manipulate virtual network configurations to possibly exfiltrate data or further move in the network.

Then, a shift in activity pattern occurs only three hours later on January 24 at noon. At this time, DC-Aphrodite becomes most distinctive to SRV-Titan. Specifically in activities such as *brough no user action*, *task scheduler launched instance*, and *terminal services session change*. On January 24 and 25, *Terminal Services session change*s frequently occur either due *connection* to or *disconnection* from the *remote* or *console terminal*; or *a user logged off*. These log entries can be found in Listing 8.5.

Listing 8.5: `Microsoft-Windows-Wcmsvc%4Operational`

```
DC-APHRODITE

        ...
        1/24/2023 9:20:49 AM, 1006, 1536, Verbose, Microsoft-Windows-Wcmsvc, DC-APHRODITE.
            Akropolys.nl, A Terminal Services session change was processed. ;;;; Reason: A
            session was disconnected from the remote terminal
        1/24/2023 9:20:48 AM, 1006, 1536, Verbose, Microsoft-Windows-Wcmsvc, DC-APHRODITE.
            Akropolys.nl, A Terminal Services session change was processed. ;;;; Reason: A
            user has logged off the session
        1/24/2023 9:19:02 AM, 1006, 1536, Verbose, Microsoft-Windows-Wcmsvc, DC-APHRODITE.
            Akropolys.nl, A Terminal Services session change was processed. ;;;; Reason: A
            session was connected to the remote terminal
        ...
        1/24/2023 9:22:12 AM, 1006, 1792, Verbose, Microsoft-Windows-Wcmsvc, DC-APHRODITE.
            Akropolys.nl, A Terminal Services session change was processed. ;;;; Reason: A
            session was connected to the console terminal
```

Observing multiple session changes over a short period could indicate either legitimate activity or something suspicious, depending on the expectations. The accumulation of these activities by solely the user DC-APHRODITE.Akropolys.nl could indicate that an attacker is attempting to exploit or maintain control over this critical host. The concentration of this activity solely by a single user makes it more likely to be targeted and deliberate.

From January 25 onward, DC-Aphrodite's log entries reverted to showing the most disparity with EXC-Calypos. These log messages included activities such as *network no user action*, *fail*ures in the *windows biometric system*, and changes in *transport* settings. While these activities might suggest a return to normal behaviour, the context of the high frequency of Terminal Services session changes raises a strong likelihood that these activities are part either the attack or its aftermath.

Listing 8.6: `Microsoft-Windows-Biometrics%4Operational`

```
EXC-CALYPSO

        ...
        1/24/2023 12:51:37 PM, 1109, 1488, Verbose, Microsoft-Windows-Biometrics, EXC-CALYPSO
            .Akropolys.nl, The Windows Biometric Service failed to configure a Biometric Unit
             for sensor: Face Recognition Infrared Camera (\FacialFeatures\Virtual Sensors
            \{063436EF-2F27-4B5F-9192-A31BE552253B}). The operation failed with error: 0
            x80004005 See the "Details" pane for information about the failing configuration.
        1/24/2023 12:51:37 PM, 1105, 1488, Verbose, Microsoft-Windows-Biometrics, EXC-CALYPSO
            .Akropolys.nl, The Windows Biometric Service failed to initialize an adapter
            binary: Face Recognition Infrared Camera (\FacialFeatures\Virtual Sensors\{063436
            EF-2F27-4B5F-9192-A31BE552253B}). The module's "Sensor Adapter" initialization
            routine failed with error: 0x80004005 See the "Details" pane for information
            about the failing configuration.
```

```
1/24/2023 12:51:33 PM, 1600, 1488, Information, Microsoft-Windows-Biometrics, EXC-
    CALYPSO.Akropolys.nl, The Windows Biometric Service failed to start its secure
    component. Reason for unavailability: 0. The operation failed with error: %2.
```

As shown in Listing 8.6, the errors in the biometric service on EXC-CALYPSO occur simultaneously with suspicious Terminal Services session changes on DC-APHRODITE. This increases the level of suspicion as it might suggest that the attacker is targeting multiple systems to disrupt security mechanisms or create confusion to cover their tracks.

8.1. Summary

This section addresses the third sub-research question:

> **RQ3:** *Can the textual content - measured utilising the selected delta method with the chosen configuration - accurately define the behaviour of the hosts within each active timeframe and identify the timeframes and during which an incident occurred?*

For this purpose, the textual content in Demo-Case 1 was organised into 60-minute time frames. This approach revealed that host activity was primarily concentrated in January 2023. Excluding this highly active period during clustering resulted in consistent clustering over the entire period, with the only difference being reduced inter-host distances among workstations and servers. Conversely, clustering based solely on January 2023 resulted in the servers and domain controller all joining as independent groups. This suboptimal clustering likely stems from failing to account for low-frequency interactions, though it could also indicate anomalous behaviour specific to January 2023. To explore this further, the clustering and behaviour of DC-Aphrodite, the compromised host, were analysed within each time frame.

The examination of DC-Aphrodite's behaviour suggested that on 2-3 January, DC-Aphrodite and EXC-Calypso encountered asynchronously that the Remote Desktop Service Session was not accepting logons after the initialisation of the RDSAppXPlugin. The classification of this behaviour hinges on whether these incidents coincided with expected maintenance activities. Then, approximately 8 hours later, the remaining hosts, namely the workstations and server, exhibited no activity for several days. On January 10, all hosts resumed activity. WKS-Froukje and DC-Aphrodite started remote desktop session on the same network address relatively close in time. However, less than a day later, DC-Aphrodite showed to be not configured as a Remote Desktop Session Host. Subsequently, on January 24, during the morning hours, DC-Aphrodite experienced numerous Terminal Service session changes and underwent repeated pauses and immediate restarts of its Miniport NIC Microsoft Hyper-V Network Adapter. Concurrently, EXC-Calypso encountered multiple Biometric Service failures.

Thus, the textual content measured using Eders Delta with 4 grams applied to each time frame helps tracing actions to identify key points, and understand the extent of a compromise. The insights gained can guide incident response efforts, such as which systems need to be isolated, what logs need further examination, and what preventive measures should be implemented.

9

Threats to Validity

This chapter discusses the potential threats to the validity of the findings obtained throughout this research. Here, section 9.1 focuses on the internal validity and section 9.2 on the external validity. Acknowledging these threats is essential for understanding the study's limitations and framing the interpretation of the results within an appropriate context.

9.1. Internal Validity

Selection Bias

In determining the most optimal Delta method and n-gram size, a specific subset of log types and corresponding n-grams were selected for detailed analysis. The log types chosen exhibited outstanding scores relative to other methods tested. Within these selected logs, the n-grams analysed were those identified as either the most or least influential in contributing to the results.

The rationale behind this selective analysis was to manage the complexity and scale of the data, enabling a more focused investigation. However, this decision introduces a potential for selection bias. By concentrating exclusively on log types and n-grams that demonstrated the most pronounced performance - whether positive or negative -, there is a risk of overlooking other log types and n-grams that, although less striking in their individual performance, may offer critical insights when considered as part of the larger dataset.

This selective focus could skew the overall findings, leading to conclusions that do not fully capture the diversity and subtleties of the data. As a result, the final analysis might disproportionately reflect the characteristics of the high-performing logs and n-grams, potentially masking trends or patterns present in the less prominent data. Such bias could limit the generalisability of the results, as the analysis may not accurately represent the full scope of log types and n-grams available, thereby compromising the robustness of the conclusions drawn.

To mitigate this, it would be essential in future studies to consider a more comprehensive sampling strategy that includes a broader range of log types and n-grams, even if they initially appear less significant. This approach would provide a more balanced view and reduce the risk of selection bias, thereby enhancing the reliability and validity of the findings.

Instrumentation

Within this research, there is a narrow focus on the Burrows Delta method and its variants without considering other established methods in the field of stylometry. While the Burrows Delta method is a widely recognised and effective tool for textual analysis, the exclusive reliance on this single methodological framework may limit the comprehensiveness and generalisability of the findings. Alternative methods could yield varying results or provide alternative explanations for the observed patterns, offering a more holistic understanding of the data.

The reliance on a singular methodological approach could also confound the interpretation of the results. If other methods were employed, they might either reinforce the findings obtained through the Burrows Delta method or, conversely, challenge the validity of these results by highlighting inconsistencies or uncovering additional layers of complexity. The absence of such comparative analysis restricts the scope of the conclusions and raises questions about the robustness of the research findings.

To address this limitation, future research should incorporate a broader range of stylometric techniques to ensure a more comprehensive analysis. By comparing the results across multiple methods, researchers can achieve a more nuanced understanding of the data, validate the findings from the Burrows Delta method, and potentially uncover new insights that would otherwise remain obscured. This multi-method approach would strengthen the validity of the conclusions and provide a more complete representation of the textual patterns under investigation.

9.2. External Validity

Population Validity

In this research, upon determining the optimal delta method and n-gram size, the reliance on a single dataset case for analysis poses a significant threat to the validity of the findings. This concern is further exacerbated by the fact that the analysis was conducted on only a single host within this case. By focusing solely on one case and one host, variations that could exist in other hosts or dataset cases may have been overlooked. These variations could produce different results, revealing alternative patterns or trends not captured in the current analysis. As a result, the broader applicability of our findings is constrained, as they are based on a limited and possibly non-representative sample of data.

The exclusive usage of one dataset case and host raises concerns about the robustness and generalisability of the conclusions. The findings might be specific to the unique characteristics of the dataset in question and, therefore, may not hold in other contexts where the data differs in structure, content, or complexity. This limitation highlights the importance of validating results across multiple datasets to ensure that the observed patterns are not artifacts of a particular dataset but are indeed representative of a broader phenomenon.

To enhance the external validity of this research, future studies should include analyses across multiple datasets with varying characteristics. By doing so, researchers can test the consistency of their findings, determine the extent to which the results generalise to other contexts, and identify any dataset-specific effects. This approach would provide a more comprehensive understanding and increase confidence in the broader applicability of the conclusions.

Ecological Validity

This research focused exclusively on post-incident data rather than considering real-time monitoring scenarios. While this approach allowed for a detailed examination of the events after they occurred, it significantly restricted the ecological validity of the findings. The dynamics, constraints, and decision-making processes inherent in real-time monitoring differ markedly from those in post-incident analysis, where the luxury of time allows for more thorough investigation and reflection.

The choice to focus solely on post-incident data raises concerns about the applicability of the results to real-time monitoring systems. In real-time scenarios, factors such as the urgency of response, incomplete data, and the need for rapid decision-making introduce complexities that are not present in post-incident analyses. Consequently, the derived insights may not fully translate to or be effective in real-time applications where the operational environment is more fluid and unpredictable.

To address this limitation, future research should incorporate real-time data and scenarios into the analysis. By doing so, researchers can assess the effectiveness of the methodologies and insights in a dynamic, real-time environment, providing a more comprehensive understanding of their applicability across different contexts. This approach would not only enhance the ecological validity of the findings but also ensure that the research contributes meaningfully to the development of real-time monitoring systems, thereby broadening the impact and relevance of the study.

10

Conclusion

The widespread success of AI has fostered an 'AI solutionism' mindset, leading to an increasing reliance on AI-based methods and a neglect of traditional approaches, particularly in anomaly detection. Anomaly detection solutions can generally be categorised into large-scale and log-based methods, but both lack transparency. To move beyond AI solutionism in order to gain more understanding, this research proposes a complementary method designed to localise highly probable anomalous hosts and time frames. This method is designed to reduce the number of logs that advanced log-based anomaly detection methods or security analists need to process, thereby enabling more timely action. Importantly, not using AI in this complementary method provides clear insights into the log types and activities that trigger alarms, enhancing the system's transparency, interpretability and trustlevel.

Hence, this research addresses a number of challenges, leading to the following contributions:

- Introducing the well-known statistical method from stylometry, Burrows Delta, as an innovative approach to anomaly detection,
- Investigating the impact of different variants of Burrows Delta on anomaly detection performance,
- Offering concise summaries of host behaviours, enabling analysts to gain insights into activities quickly,
- Summarising the temporal behaviour of hosts to allow faster identification of attack times and activities by analysts

The answers to the research questions provided in chapter 4 are given below:

> **RQ:** *Can the metadata of the logs of the hosts within the network effectively quantify the inter-host distances to define host behaviour such that it can identify the outlying host and the time frame of the incident?*

The proposed method employs a variant of the Burrows Delta method - a prominent technique within authorship attribution - applied to the textual content of log metadata. More specifically, Eders Delta using 4-grams has been derived to be most effective to define the behaviour of the hosts. This method localises potentially anomalous hosts and time periods by comparing host behaviours in terms of common, distinctive and unique described by the log's content. The method provides insight into the behaviour of the hosts such that specific hosts, log types and time frames are identified as outliers requiring in-depth analysis. This eliminates the need for analysts to review each log manually. Instead, analysts can quickly ascertain the host's activities and determine which logs require more advanced, in-depth analysis and whether immediate precautionary measures are necessary. However, unlike within the attributes, the log entries containing no messages are being neglected during the analysis.

RQ1: *How effectively do the attributes of the log metadata quantify the inter-host distances to define the behaviour of the hosts such that it can identify the outlying host?*

Before relying on the textual content of the metadata, the attributes within the metadata — namely, size, log types, and event IDs — were evaluated in this study. While these attributes appeared to quantify the overall inter-host distances accurately, they required extensive look-up time to gain understanding and lacked details in identifying the behaviour of the hosts. Among them, the event ID attribute provided the most insight into host activities, whereas, log types offered more information than size. The broader activity range covered by log types was narrowed by the event IDs while still encompassing a broad spectrum. Both attributes, however, require extensive lookup for precise activity determination. Therefore, the second research question was introduced to enhance the identification of host activities and reduce the reliance on extensive lookup procedures.

RQ2: *How effectively does the textual content of the log metadata quantify the inter-host distances to define the behaviour of the hosts such that it can identify outlying hosts?*

To evaluate the effectiveness of the textual content of the metadata, the optimal n-gram size and delta method were determined. This involved evaluating the clusterings of various n-gram sizes and delta methods based on a prediction. The shortlisted n-gram sizes and delta methods were further assessed by examining their behaviour. Here, the behaviour has been categorised into common, distinctive, and unique patterns. This analysis concluded that Eders Delta under 4-grams was the most effective method. Utilising this approach to define the behaviour, the textual content revealed alternative behaviour than the attributes; however requiring no extensive lookup and defining host behaviour more precisely. Based on this overall behaviour, the specific timing of incidents and the identification of pre-attack activities remained unclear. Consequently, the third research question was introduced to improve the detection of incident timing and to understand pre-attack activities.

RQ3: *Can the textual content - measured utilising the selected delta method with the chosen configuration - accurately define the behaviour of the hosts within each active timeframe and identify the timeframe during which an incident occurred?*

The textual content was organised into 60-minute time frames, revealing periods of high activity. Clustering based solely on these high-activity periods produced alternative clusterings. Eder's Delta under 4-grams was applied to each of these time frames within the high activity period, successfully uncovering the pre- and post-attack activities of the outlying host. Additionally, the most interesting time frames and logs were identified, with an explanation that led to this conclusion, thereby ensuring transparency.

In short, the proposed solution demonstrates the alternative capability of the textual content within the metadata to understandably locate outlying hosts, logs and time frames, necessitating further advanced in-depth analysis. Hence, this research represents a step towards transcending AI solutionism and re-exploring established methods across various fields. It commences the development of complementary approaches that enhance understanding the decisions of the underlying black boxes and reduce the amount of logs the security analysist need to consider during a compromise by being able to quickly - by lowering the need of extensive lookup - understand the behaviour of the hosts over time.

References

[1] A. Kaplan and M. Haenlein, "Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence," *Business horizons*, vol. 62, no. 1, pp. 15–25, 2019.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[3] M. Khan, B. Jan, H. Farman, J. Ahmad, H. Farman, and Z. Jan, "Deep learning methods and applications," *Deep learning: convergence to big data analytics*, pp. 31–42, 2019.

[4] Y. K. Dwivedi, L. Hughes, E. Ismagilova, *et al.*, "Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International journal of information management*, vol. 57, p. 101 994, 2021.

[5] M. Thomas, "The future of ai: How artificial intelligence will change the world," *Built in*, vol. 10, 2022.

[6] S. Lindgren, *Critical theory of AI*. John Wiley & Sons, 2023.

[7] J. Cunningham, G. Benabdallah, D. Rosner, and A. Taylor, "On the grounds of solutionism: Ontologies of blackness and hci," *ACM Transactions on Computer-Human Interaction*, vol. 30, no. 2, pp. 1–17, 2023.

[8] B. Dhamodharan, "Beyond traditional methods: A novel approach to anomaly detection and classification using ai techniques," *Transactions on Latest Trends in Artificial Intelligence*, vol. 3, no. 3, 2022.

[9] M. Yang and J. Zhang, "Data anomaly detection in the internet of things: A review of current trends and research challenges," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023.

[10] S. Martínez-Fernández, J. Bogner, X. Franch, *et al.*, "Software engineering for ai-based systems: A survey," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 2, pp. 1–59, 2022.

[11] F. O. Usman, N. L. Eyo-Udo, E. A. Etukudoh, B. Odonkor, C. V. Ibeh, and A. Adegbola, "A critical review of ai-driven strategies for entrepreneurial success," *International Journal of Management & Entrepreneurship Research*, vol. 6, no. 1, pp. 200–215, 2024.

[12] S.-M. Cheong, K. Sankaran, and H. Bastani, "Artificial intelligence for climate change adaptation," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 5, e1459, 2022.

[13] W. J. Von Eschenbach, "Transparency and the black box problem: Why we do not trust ai," *Philosophy & Technology*, vol. 34, no. 4, pp. 1607–1622, 2021.

[14] B. Kim, J. Park, and J. Suh, "Transparency and accountability in ai decision support: Axplaining and visualizing convolutional neural networks for text information," *Decision Support Systems*, vol. 134, p. 113 302, 2020.

[15] V.-H. Le and H. Zhang, "Log-based anomaly detection without log parsing," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, IEEE, 2021, pp. 492–504.

[16] S. Thudumu, P. Branch, J. Jin, and J. Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *Journal of Big Data*, vol. 7, pp. 1–30, 2020.

[17] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: A review," *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2663–2693, 2022.

[18] C. B. Azodi, J. Tang, and S.-H. Shiu, "Opening the black box: Interpretable machine learning for geneticists," *Trends in genetics*, vol. 36, no. 6, pp. 442–455, 2020.

[19] H. Liu and H. Wang, "Real-time anomaly detection of network traffic based on cnn," *Symmetry*, vol. 15, no. 6, p. 1205, 2023.

[20] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024.

[21] H. Jin, G. Papadimitriou, K. Raghavan, *et al.*, "Large language models for anomaly detection in computational workflows: From supervised fine-tuning to in-context learning," *arXiv preprint arXiv:2407.17545*, 2024.

[22] Q. V. Liao and J. W. Vaughan, "Ai transparency in the age of llms: A human-centered research roadmap," *arXiv preprint arXiv:2306.01941*, pp. 5368–5393, 2023.

[23] S. Gu, Y. Chu, W. Zhang, P. Liu, Q. Yin, and Q. Li, "Research on system log anomaly detection combining two-way slice gru and ga-attention mechanism," in *2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, IEEE, 2021, pp. 577–583.

[24] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 397–406.

[25] X. Liu, W. Liu, X. Di, *et al.*, "Lognads: Network anomaly detection scheme based on log semantics representation," *Future Generation Computer Systems*, vol. 124, pp. 390–405, 2021.

[26] J. Fischer and R. Ma, "Sailing in high-dimensional spaces: Low-dimensional embeddings through angle preservation," *arXiv preprint arXiv:2406.09876*, 2024.

[27] D. Lv, N. Luktarhan, and Y. Chen, "Conanomaly: Content-based anomaly detection for system logs," *Sensors*, vol. 21, no. 18, p. 6125, 2021.

[28] J. Camacho-Collados and M. T. Pilehvar, "Embeddings in natural language processing," in *Proceedings of the 28th international conference on computational linguistics: tutorial abstracts*, 2020, pp. 10–15.

[29] H. Zhang and M. O. Shafiq, "Survey of transformers and towards ensemble learning using transformers for natural language processing," *Journal of big Data*, vol. 11, no. 1, p. 25, 2024.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin, """ why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[31] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*, PMlR, 2017, pp. 3145–3153.

[32] D. Han, Z. Wang, W. Chen, *et al.*, "Deepaid: Interpreting and improving deep learning-based anomaly detection in security applications," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3197–3217.

[33] P. Du, X. Bai, K. Tan, *et al.*, "Advances of four machine learning methods for spatial data handling: A review," *Journal of Geovisualization and Spatial Analysis*, vol. 4, pp. 1–25, 2020.

[34] M Vimalkumar, A. Gupta, D. Sharma, and Y. Dwivedi, "Understanding the effect that task complexity has on automation potential and opacity: Implications for algorithmic fairness," *AIS Transactions on Human-Computer Interaction*, vol. 13, no. 1, pp. 104–129, 2021.

[35] K. Lagutina, N. Lagutina, E. Boychuk, *et al.*, "A survey on stylometric text features," in *2019 25th Conference of Open Innovations Association (FRUCT)*, IEEE, 2019, pp. 184–195.

[36] P. Juola *et al.*, "Authorship attribution," *Foundations and Trends® in Information Retrieval*, vol. 1, no. 3, pp. 233–334, 2008.

[37] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard, "Surveying stylometry techniques and applications," *ACM Computing Surveys (CSuR)*, vol. 50, no. 6, pp. 1–36, 2017.

[38] S. Evert, T. Proisl, F. Jannidis, *et al.*, "Understanding and explaining delta measures for authorship attribution," *Digital Scholarship in the Humanities*, vol. 32, no. suppl_2, pp. ii4–ii16, 2017.

[39] S. Evert, T. Proisl, T. Vitt, C. Schöch, F. Jannidis, and S. Pielström, "Towards a better understanding of burrows's delta in literary authorship attribution," in *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, 2015, pp. 79–88.

[40] Y. Sari, M. Stevenson, and A. Vlachos, "Topic or style? exploring the most useful features for authorship attribution," Association for Computational Linguistics, 2018.

[41] R. W. Bailey, "Authorship attribution in a forensic setting," D. E. Ager, F. E. Knowles, and J. Smith, Eds., John Goodman, 1979, pp. 1–15.

[42] J. N. G. Binongo and M. W. A. Smith, "The application of principal component analysis to stylometry," *Literary and Linguistic Computing*, vol. 14, pp. 445–465, 1999.

[43] J. Burrows, "'delta'—a measure of stylistic difference and a guide to likely authorship," *Literary and Linguistic Computing*, vol. 17, no. 3, pp. 267–287, 2002.

[44] P. W. Smith and W. Aldridge, "Improving authorship attribution: Optimizing burrows' delta method," *Journal of Quantitative Linguistics*, vol. 18, no. 1, pp. 63–88, 2011.

[45] F. Jannidis, S. Pielström, C. Schöch, and T. Vitt, "Improving burrows' delta. an empirical evaluation of text distance measures," in *Digital Humanities Conference*, vol. 11, 2015.

[46] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[47] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software*, vol. 10, no. 1, pp. 1–20, 2017.

[48] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105 524, 2020.

[49] C. Andrade, "Z scores, standard scores, and composite test scores explained," *Indian Journal of Psychological Medicine*, vol. 43, no. 6, pp. 555–557, 2021.

[50] H. Abdi, "Z-scores," *Encyclopedia of measurement and statistics*, vol. 3, pp. 1055–1058, 2007.

[51] A. Ghasemi and S. Zahediasl, "Normality tests for statistical analysis: A guide for non-statisticians," *International journal of endocrinology and metabolism*, vol. 10, no. 2, p. 486, 2012.

[52] R. E. Shiffler, "Maximum z scores and outliers," *The American Statistician*, vol. 42, no. 1, pp. 79–80, 1988.

[53] S. G. Kwak and J. H. Kim, "Central limit theorem: The cornerstone of modern statistics," *Korean journal of anesthesiology*, vol. 70, no. 2, pp. 144–156, 2017.

[54] M. R. Islam, "Sample size and its role in central limit theorem (clt)," *Computational and Applied Mathematics Journal*, vol. 4, no. 1, pp. 1–7, 2018.

[55] U. Habib, G. Zucker, M. Blochle, F. Judex, and J. Haase, "Outliers detection method using clustering in buildings data," in *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2015, pp. 000 694–000 700.

[56] S. Argamon, "Interpreting burrows's delta: Geometric and probabilistic foundations," *Literary and Linguistic Computing*, vol. 23, no. 2, pp. 131–147, 2008.

[57] S. T. Piantadosi, "Zipf's word frequency law in natural language: A critical review and future directions," *Psychonomic bulletin & review*, vol. 21, pp. 1112–1130, 2014.

[58] S. Stein and S. Argamon, "A mathematical explanation of burrows's delta," in *Proceedings of the Digital Humanities Conference*, Citeseer, 2006, pp. 207–209.

[59] D. Wettschereck, D. W. Aha, and T. Mohri, "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms," *Artificial Intelligence Review*, vol. 11, pp. 273–314, 1997.

[60] D. Stanikūnas, J. Mandravickaitė, and T. Krilavičius, "Comparison of distance and similarity measures for stylometric analysis of lithuanian texts," in *CEUR Workshop proceedings*, 2017, pp. 1–7.

[61] B. Bickel and J. Nichols, "Inflectional morphology," *Language typology and syntactic description*, vol. 3, no. 2, pp. 169–240, 2007.

[62] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[63] J. Osborne, "Notes on the use of data transformations," *Practical assessment, research, and evaluation*, vol. 8, no. 1, p. 6, 2002.

[64] S. Evert, T. Proisl, C. Schöch, F. Jannidis, S. Pielström, and T. Vitt, "Explaining delta, or: How do distance measures for authorship attribution work?" *URL: http://dx. doi. org/10.5281/zenodo*, vol. 18308, 2015.

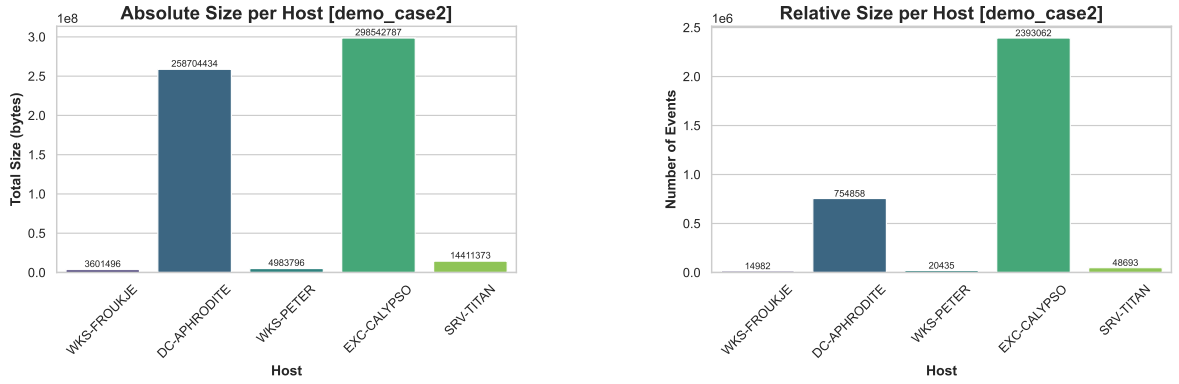
A

Absolute and Relative Sizes

Here, section A.1 contains the size distributions of the hosts in Demo-Case 2, 3 and 4. Then, section A.2 contains the clusterings based on the size distributions of the hosts in Demo-Case 2, 3 and 4.

A.1. Size Distribution of the Hosts

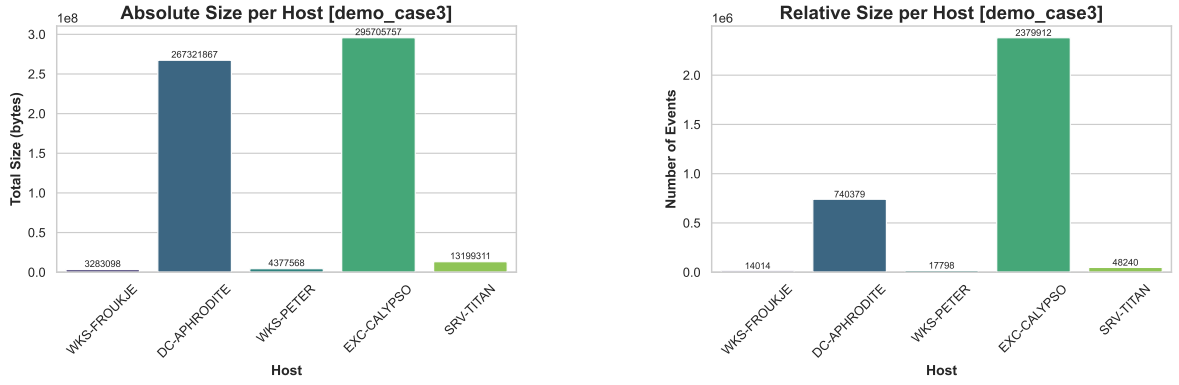
Figures A.1, A.2, and A.3 illustrate the absolute and relative sizes of the hosts observed in Demo-case 2, 3, and 4, respectively.



(a) Absolute Sizes

(b) Relative Sizes

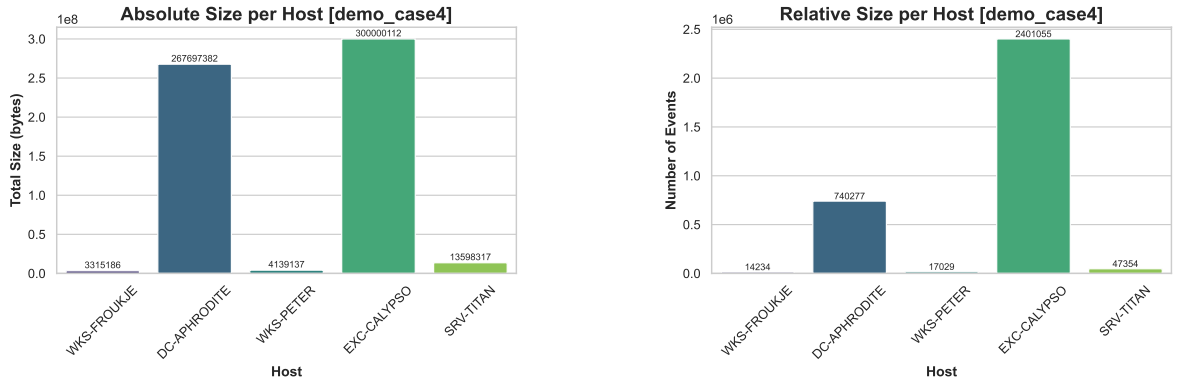
Figure A.1: Sizes of Demo-Case 2



(a) Absolute Sizes

(b) Relative Sizes

Figure A.2: Sizes of Demo-Case 3



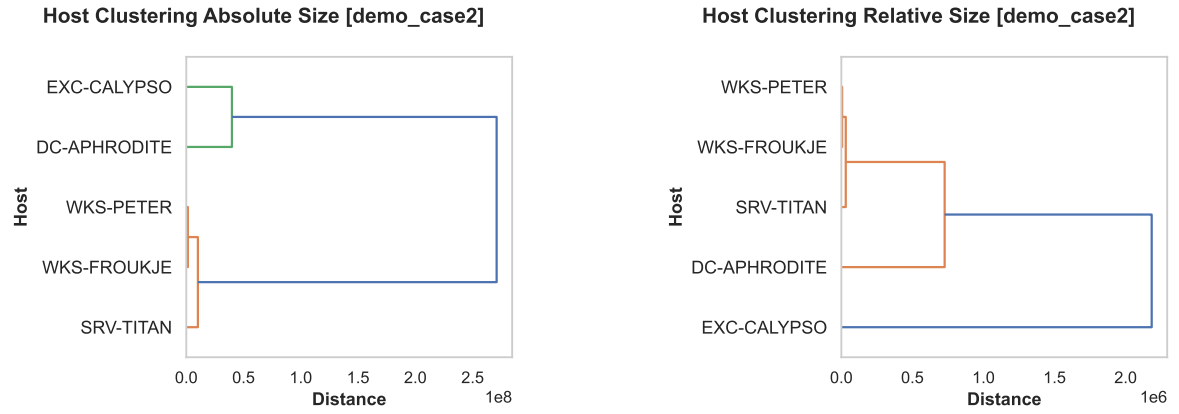
(a) Absolute Sizes

(b) Relative Sizes

Figure A.3: Sizes of Demo-Case 4

A.2. Size Clustering of the Hosts

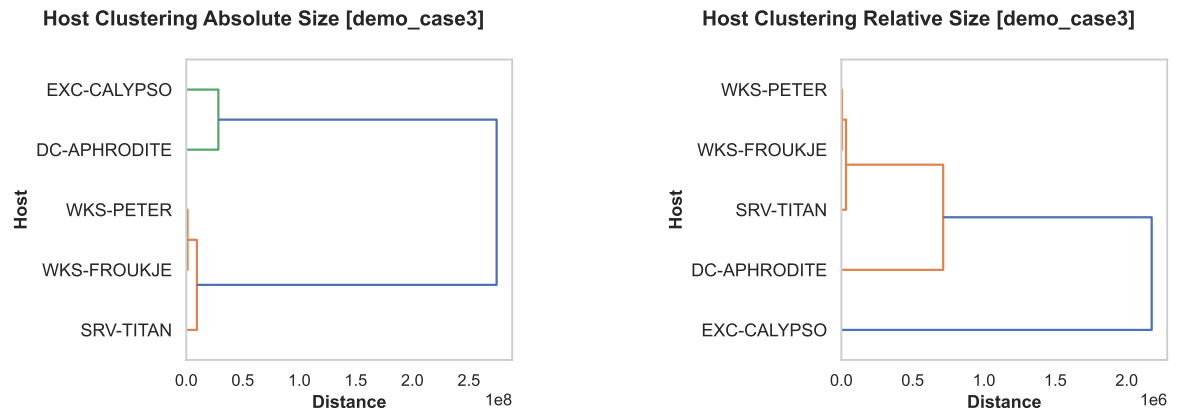
Figures A.4, A.5, and A.6 illustrate the clustering based on the absolute and relative sizes of the hosts observed in Demo-case 2, 3, and 4, respectively.



(a) Absolute Sizes

(b) Relative Sizes

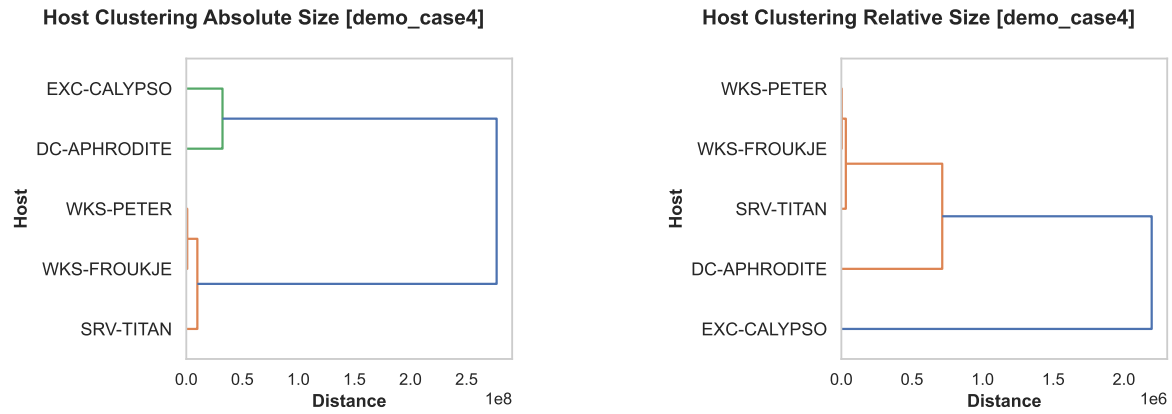
Figure A.4: Clustering based on sizes for Demo-Case 2



(a) Absolute Sizes

(b) Relative Sizes

Figure A.5: Clustering based on sizes for Demo-Case 3



(a) Absolute Sizes

(b) Relative Sizes

Figure A.6: Clustering based on sizes for Demo-Case 4

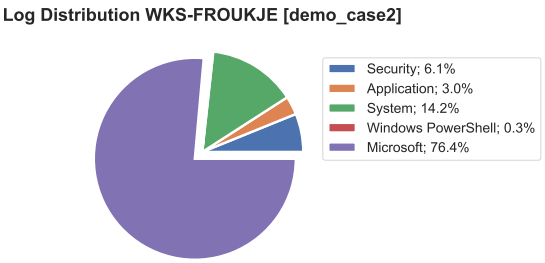
B

Log Distributions

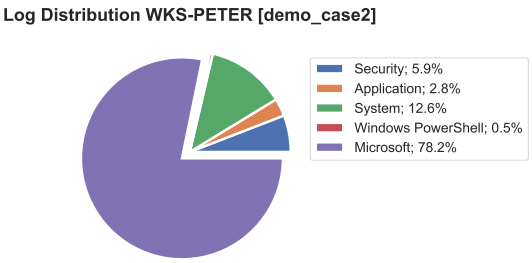
Here, section B.1 contains the log distributions of the hosts in Demo-Case 2, 3 and 4. Then, section B.2 contains the clusterings based on the log distributions of the hosts in Demo-Case 2, 3 and 4.

B.1. Log Distribution of the Hosts

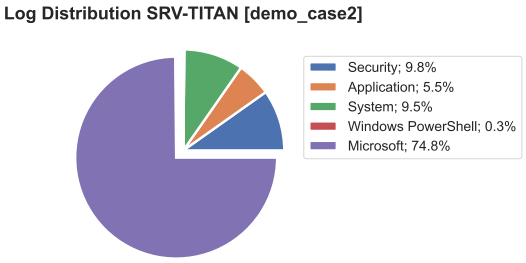
Figures B.1, B.2, and B.3 present the log distribution of the hosts observed in Demo-case 2, 3, and 4, respectively.



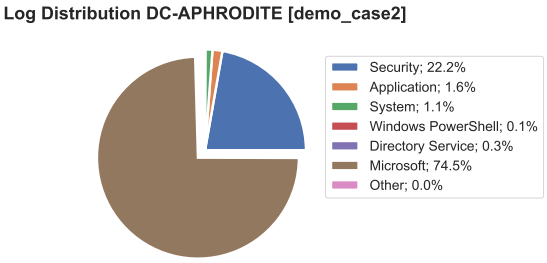
(a) Log Distribution WKS-Froukje



(b) Log Distribution WKS-Peter

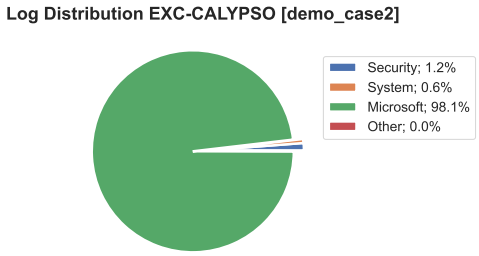


(c) Log Distribution SRV-Titan



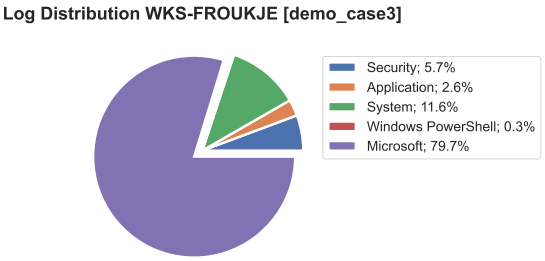
(d) Log Distribution DC-Aphrodite

Figure B.1: Log Distribution of Demo-Case 2

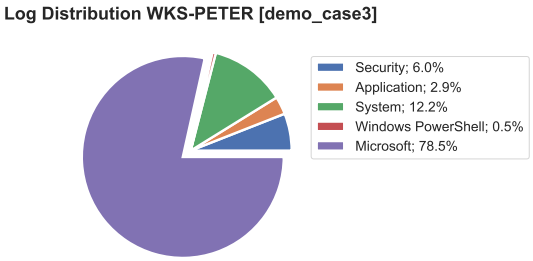


(e) Log Distribution EXC-Calypso

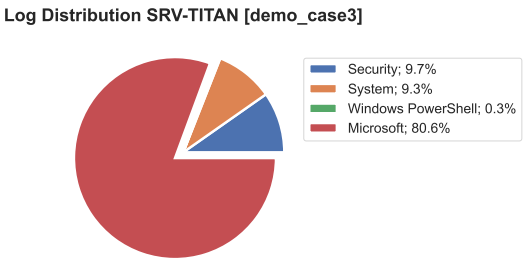
Figure B.1: Log Distribution of Demo-Case 2 (cont.)



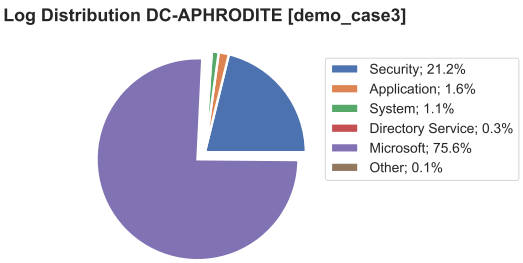
(a) Log Distribution WKS-Froukje



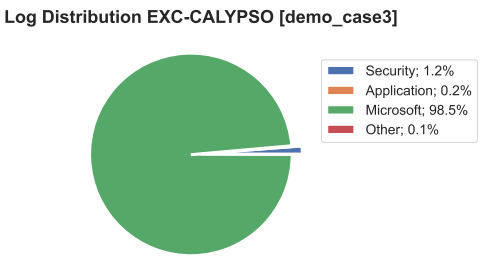
(b) Log Distribution WKS-Peter



(c) Log Distribution SRV-Titan

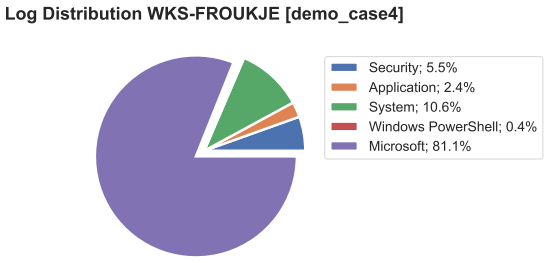


(d) Log Distribution DC-Aphrodite

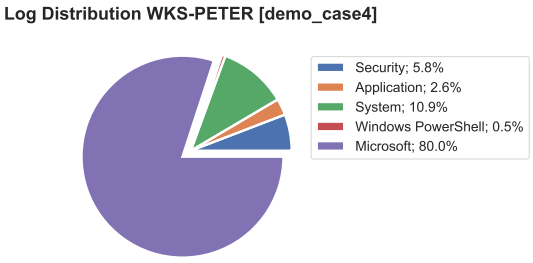


(e) Log Distribution EXC-Calypso

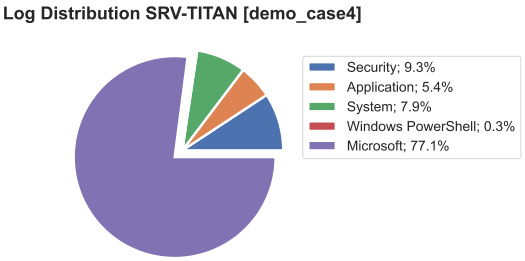
Figure B.2: Log Distribution of Demo-Case 3



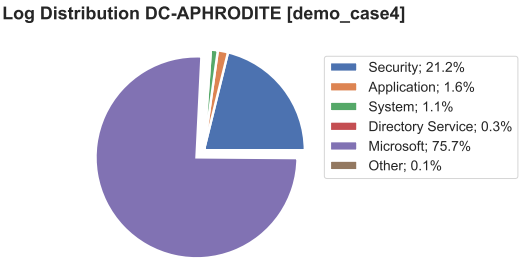
(a) Log Distribution WKS-Froukje



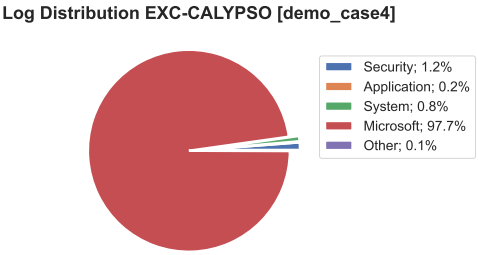
(b) Log Distribution WKS-Peter



(c) Log Distribution SRV-Titan



(d) Log Distribution DC-Aphrodite

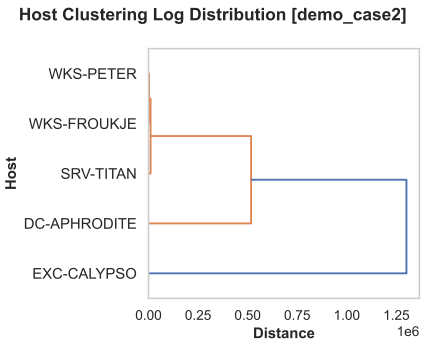


(e) Log Distribution EXC-Calypso

Figure B.3: Log Distribution of Demo-Case 4

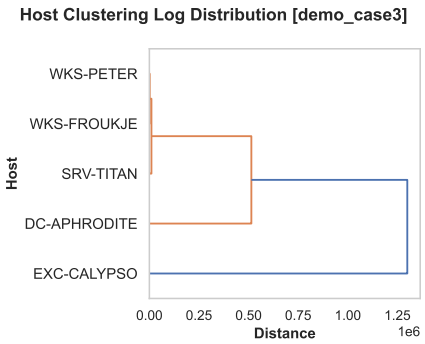
B.2. Log Distribution Clustering of the Hosts

Figures B.4a, B.4b, and B.4c illustrate the clustering based on the log distributions of the hosts observed in Demo-case 2, 3, and 4, respectively.

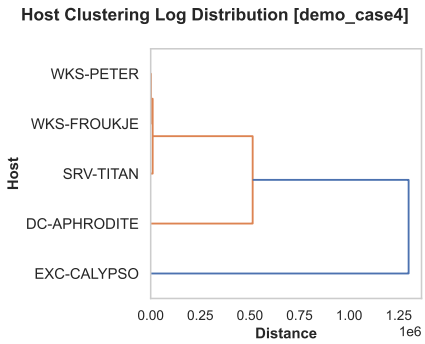


(a) Demo-Case 2

Figure B.4: Clustering based on log distribution



(b) Demo-Case 3



(c) Demo-Case 4

Figure B.4: Clustering based on log distribution (cont.)

B.3. Detailed Log Distribution

Table B.1 through B.4 provide the event counts for each log type per host present in the corresponding Demo-Case 1 till 4.

Table B.1: Absolute Size per Log Type for each Host in Demo-Case 1

Log Event Counts per Host (Part 1) [demo_case1]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Active Directory Web Services	0	105	0	0	0
Application	380	11904	514	6286	2610
DFS Replication	0	96	0	0	0
DNS Server	0	89	0	0	0
Directory Service	0	2618	0	0	0
MSExchange Management	0	0	0	63	0
Microsoft-Client-Licensing-Platform%4Admin	0	125	0	47	35
Microsoft-Exchange-ActiveMonitoring%4MaintenanceDefinition	0	0	0	2018	0
Microsoft-Exchange-ActiveMonitoring%4MaintenanceResult	0	0	0	52730	0
Microsoft-Exchange-ActiveMonitoring%4MonitorDefinition	0	0	0	16360	0
Microsoft-Exchange-ActiveMonitoring%4MonitorResult	0	0	0	689817	0
Microsoft-Exchange-ActiveMonitoring%4ProbeDefinition	0	0	0	1037	0
Microsoft-Exchange-ActiveMonitoring%4ProbeResult	0	0	0	497504	0
Microsoft-Exchange-ActiveMonitoring%4ResponderDefinition	0	0	0	13150	0
Microsoft-Exchange-ActiveMonitoring%4ResponderResult	0	0	0	0	0
Microsoft-Exchange-Compliance%4DarRuntimeLogs	0	0	0	26	0
Microsoft-Exchange-Compliance%4DiscoveryLogs	0	0	0	108	0
Microsoft-Exchange-DxStoreHA%4Server	0	0	0	7	0
Microsoft-Exchange-ESE%4Operational	0	0	0	227	0
Microsoft-Exchange-HighAvailability%4AppLogMirror	0	0	0	0	0
Microsoft-Exchange-HighAvailability%4Debug	0	0	0	144	0
Microsoft-Exchange-HighAvailability%4Monitoring	0	0	0	54	0
Microsoft-Exchange-HighAvailability%4Network	0	0	0	39279	0
Microsoft-Exchange-HighAvailability%4Operational	0	0	0	3641	0
Microsoft-Exchange-HighAvailability%4Seeding	0	0	0	46	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionLogs	0	0	0	3796	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionResults	0	0	0	44	0
Microsoft-Exchange-ManagedAvailability%4StartupNotification	0	0	0	122	0
Microsoft-Exchange-ManagedAvailability%4ThrottlingConfig	0	0	0	59	0
Microsoft-Exchange-PushNotifications%4Operational	0	0	0	18	0
Microsoft-Office Server-Search%4Operational	0	0	0	2082	0
Microsoft-Windows-AAD%4Operational	0	44	0	22	20
Microsoft-Windows-AppModel-Runtime%4Admin	0	400	0	80	76
Microsoft-Windows-AppReadiness%4Admin	0	404	0	143	140
Microsoft-Windows-AppReadiness%4Operational	0	22	0	977	957
Microsoft-Windows-AppXDeployment%4Operational	0	34	0	46	44
Microsoft-Windows-AppXDeploymentServer%4Operational	0	2760	0	865	859
Microsoft-Windows-Application Server-Applications%4Operational	0	0	0	110	0
Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant	0	0	0	2	0
Microsoft-Windows-Application-Experience%4Program-Telemetry	6	0	9	35	27
Microsoft-Windows-ApplicationResourceManagementSystem%4Operational	0	0	0	181	174
Microsoft-Windows-AppxPackaging%4Operational	0	140	0	0	0
Microsoft-Windows-BackgroundTaskInfrastructure%4Operational	0	2	0	23	43
Microsoft-Windows-Biometrics%4Operational	0	4	0	32	44
Microsoft-Windows-Bits-Client%4Operational	1591	127	1596	1129	216
Microsoft-Windows-BranchCacheSMB%4Operational	59	0	59	0	0
Microsoft-Windows-CAPI2%4Operational	152	0	152	0	0
Microsoft-Windows-CloudStore%4Operational	0	266	0	0	0
Microsoft-Windows-CodeIntegrity%4Operational	7	16	37	16	14
Microsoft-Windows-Containers-BindFlt%4Operational	0	16	0	0	0
Microsoft-Windows-Containers-Wcifs%4Operational	0	16	0	16	14
Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc	0	14	0	0	0
Microsoft-Windows-Crypto-DPAPI%4Operational	0	242	0	8	8
Microsoft-Windows-Crypto-NCrypt%4Operational	0	86	0	0	0
Microsoft-Windows-DNSServer%4Audit	0	102	0	0	0
Microsoft-Windows-DataIntegrityScan%4Admin	0	40	0	6	6
Microsoft-Windows-DateTimeControlPanel%4Operational	0	0	0	1	0
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Admin	0	1	0	71	99
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Operational	0	2	0	0	0
Microsoft-Windows-DeviceSetupManager%4Admin	0	474	0	674	221
Microsoft-Windows-DeviceSetupManager%4Operational	0	301	0	136	57
Microsoft-Windows-Dhcp-Client%4Admin	8	4	7	1	1
Microsoft-Windows-Dhcpv6-Client%4Admin	31	0	33	0	0
Microsoft-Windows-Diagnosis-DPS%4Operational	196	47	222	33	56
Microsoft-Windows-Diagnosis-PCW%4Operational	0	2199	0	2288	1728
Microsoft-Windows-Diagnosis-PLA%4Operational	80	110	106	1579	71
Microsoft-Windows-Diagnosis-Scheduled%4Operational	0	21	14	14	0
Microsoft-Windows-Diagnosis-Scripted%4Admin	1	4	4	2	0
Microsoft-Windows-Diagnosis-Scripted%4Operational	4	16	16	8	0
Microsoft-Windows-Diagnostics-Networking%4Operational	0	4	2	0	0
Microsoft-Windows-Diagnostics-Performance%4Operational	62	0	63	0	0
Microsoft-Windows-DirectoryServices-Deployment%4Operational	0	185	0	0	0
Microsoft-Windows-DriverFrameworks-UserMode%4Operational	2	0	2	0	0
Microsoft-Windows-Fault-Tolerant-Heap%4Operational	2	0	2	7	2
Microsoft-Windows-FileServices-ServerManager-EventProvider%4Operational	0	0	0	3	0

Table B.1: Absolute Size per Log Type for each Host in Demo-Case 1 (cont.)

Log Event Counts per Host (Part 2) [demo_case1]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Microsoft-Windows-Forwarding%4Operational	0	13	0	2	2
Microsoft-Windows-GroupPolicy%4Operational	1469	6832	2435	7946	8030
Microsoft-Windows-HelloForBusiness%4Operational	0	54	0	0	0
Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin	0	32	0	0	0
Microsoft-Windows-International%4Operational	1	0	1	1	2
Microsoft-Windows-Kernel-Boot%4Operational	0	64	0	0	0
Microsoft-Windows-Kernel-Cache%4Operational	0	815	0	0	0
Microsoft-Windows-Kernel-EventTracing%4Admin	3	1	43	41	7
Microsoft-Windows-Kernel-IO%4Operational	0	2518	0	1512	1464
Microsoft-Windows-Kernel-PnP%4Configuration	0	497	0	366	240
Microsoft-Windows-Kernel-PnP%4Device Management	0	207	0	0	0
Microsoft-Windows-Kernel-ShimEngine%4Operational	0	3	0	3	3
Microsoft-Windows-Kernel-WHEA%4Operational	44	68	45	16	14
Microsoft-Windows-Known Folders API Service	173	1739	194	1845	605
Microsoft-Windows-LanguagePackSetup%4Operational	30	18	36	10	8
Microsoft-Windows-LiveId%4Operational	0	556	0	585	378
Microsoft-Windows-MUI%4Operational	108	12	108	13	13
Microsoft-Windows-NCSI%4Operational	0	40	0	30	16
Microsoft-Windows-NetworkLocationWizard%4Operational	1	0	1	0	0
Microsoft-Windows-NetworkProfile%4Operational	260	159	268	115	93
Microsoft-Windows-Ntfs%4Operational	0	4361	0	808	795
Microsoft-Windows-Ntfs%4WHC	0	16	0	16	14
Microsoft-Windows-OfflineFiles%4Operational	85	0	86	0	0
Microsoft-Windows-Partition%4Diagnostic	0	29	0	0	0
Microsoft-Windows-PowerShell%4Operational	0	751	0	25173	206
Microsoft-Windows-PrintService%4Admin	32	22	43	2	2
Microsoft-Windows-Privacy-Auditing%4Operational	0	163	0	0	0
Microsoft-Windows-PushNotification-Platform%4Operational	0	406	0	1694	1754
Microsoft-Windows-ReadyBoost%4Operational	93	0	101	0	0
Microsoft-Windows-ReliabilityAnalysisComponent%4Operational	23	0	23	0	0
Microsoft-Windows-RemoteAssistance%4Operational	4	0	4	0	0
Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational	0	1961	0	1927	1162
Microsoft-Windows-RemoteDesktopServices-SessionServices%4Operational	0	2	0	1	5
Microsoft-Windows-Resource-Exhaustion-Detector%4Operational	73	34	80	30	24
Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational	12	10	16	8	15
Microsoft-Windows-SMBServer%4Connectivity	0	0	0	1	0
Microsoft-Windows-SMBServer%4Operational	0	43	0	70	61
Microsoft-Windows-SMBServer%4Security	0	37	0	1	1
Microsoft-Windows-Security-Mitigations%4KernelMode	0	7	0	0	0
Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter	0	6	0	6	6
Microsoft-Windows-ServerManager-DeploymentProvider%4Operational	0	527	0	331	182
Microsoft-Windows-ServerManager-MgmtProvider%4Operational	0	1948	0	419	441
Microsoft-Windows-ServerManager-MultiMachine%4Operational	0	1966	0	2012	878
Microsoft-Windows-SettingSync%4Debug	0	0	0	1102	1141
Microsoft-Windows-SettingSync%4Operational	0	0	0	9	9
Microsoft-Windows-Shell-Core%4AppDefaults	0	952	0	0	0
Microsoft-Windows-Shell-Core%4Operational	0	2208	0	1317	1180
Microsoft-Windows-ShellCommon-StartLayoutPopulation%4Operational	0	197	0	0	0
Microsoft-Windows-SmartCard-DeviceEnum%4Operational	0	0	0	101	42
Microsoft-Windows-SmbClient%4Connectivity	0	78	0	160	120
Microsoft-Windows-SmbClient%4Security	0	0	0	3	0
Microsoft-Windows-StateRepository%4Operational	0	777	0	63	57
Microsoft-Windows-Storage-ClassPnP%4Operational	0	102	0	71	59
Microsoft-Windows-Storage-Storport%4Operational	0	857	0	617	620
Microsoft-Windows-StorageManagement%4Operational	0	30	0	0	0
Microsoft-Windows-StorageSpaces-Driver%4Operational	0	29	0	29	25
Microsoft-Windows-Store%4Operational	0	450	0	1120	760
Microsoft-Windows-Storsvc%4Diagnostic	0	47	0	0	0
Microsoft-Windows-SystemDataArchiver%4Diagnostic	0	489588	0	0	0
Microsoft-Windows-TWinUI%4Operational	0	5	0	73	84
Microsoft-Windows-TZSync%4Operational	0	8	0	12	16
Microsoft-Windows-TaskScheduler%4Maintenance	0	174	0	52	40
Microsoft-Windows-TaskScheduler%4Operational	1021	19609	2486	6093	6587
Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational	112	208	138	198	125
Microsoft-Windows-TerminalServices-PnPDevices%4Admin	1	0	1	0	0
Microsoft-Windows-TerminalServices-Printers%4Admin	0	30	0	36	15
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin	0	10	0	14	7
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational	298	302	1931	109	69
Microsoft-Windows-Time-Service%4Operational	0	404	0	0	0
Microsoft-Windows-UAC-FileVirtualization%4Operational	0	0	0	1198	0
Microsoft-Windows-UniversalTelemetryClient%4Operational	0	753	0	97	78
Microsoft-Windows-User Device Registration%4Admin	0	373	0	12	0
Microsoft-Windows-User Profile Service%4Operational	182	93	226	81	61
Microsoft-Windows-UserPnp%4DeviceInstall	0	12	0	7	5
Microsoft-Windows-VHDMP-Operational	0	0	0	2	0
Microsoft-Windows-VolumeSnapshot-Driver%4Operational	0	162	0	162	142
Microsoft-Windows-WER-Diag%4Operational	1	0	1	0	0
Microsoft-Windows-WER-PayloadHealth%4Operational	0	9	0	0	0
Microsoft-Windows-WFP%4Operational	0	2	0	2	0
Microsoft-Windows-WMI-Activity%4Operational	0	1466	0	1666	1574
Microsoft-Windows-Wcmsvc%4Operational	0	182	0	212	152
Microsoft-Windows-WebAuthN%4Operational	0	29	0	0	0
Microsoft-Windows-WinINet-Config%4ProxyConfigChanged	0	4	0	2	2
Microsoft-Windows-WinRM%4Operational	0	398	0	2336	268
Microsoft-Windows-Windows Defender%4Operational	2	876	2	66	58
Microsoft-Windows-Windows Defender%4WHC	94	0	100	59	50
Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	377	718	409	466	326
Microsoft-Windows-WindowsBackup%4ActionCenter	11	0	15	0	0
Microsoft-Windows-WindowsSystemAssessmentTool%4Operational	39	0	39	0	0
Microsoft-Windows-WindowsUpdateClient%4Operational	1025	170	1034	52	43
Microsoft-Windows-Winlogon%4Operational	0	966	0	0	0
Microsoft-WindowsAzure-Diagnostics%4GuestAgent	1913	1535	1966	1748	1700
Microsoft-WindowsAzure-Diagnostics%4Heartbeat	1650	1592	1658	1576	1576
Microsoft-WindowsAzure-Status%4GuestAgent	112	1382	228	0	0
Microsoft-WindowsAzure-Status%4Plugins	10	1248	12	14	12
Security	805	158324	1076	29319	4521
Setup	0	29	0	107	0
System	1734	8241	2180	11330	4093
Windows PowerShell	48	1091	93	910	164

Table B.2: Absolute Size per Log Type for each Host in Demo-Case 2

Log Event Counts per Host (Part 1) [demo_case2]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Active Directory Web Services	0	116	0	0	0
Application	444	11929	579	0	2683
DFS Replication	0	105	0	0	0
DNS Server	0	95	0	0	0
Directory Service	0	2642	0	0	0
MSExchange Management	0	0	0	76	0
Microsoft-Client-Licensing-Platform%4Admin	0	131	0	51	40
Microsoft-Exchange-ActiveMonitoring%4MaintenanceDefinition	0	0	0	2229	0
Microsoft-Exchange-ActiveMonitoring%4MaintenanceResult	0	0	0	52726	0
Microsoft-Exchange-ActiveMonitoring%4MonitorDefinition	0	0	0	16232	0
Microsoft-Exchange-ActiveMonitoring%4MonitorResult	0	0	0	689838	0
Microsoft-Exchange-ActiveMonitoring%4ProbeDefinition	0	0	0	1019	0
Microsoft-Exchange-ActiveMonitoring%4ProbeResult	0	0	0	496512	0
Microsoft-Exchange-ActiveMonitoring%4ResponderDefinition	0	0	0	13340	0
Microsoft-Exchange-ActiveMonitoring%4ResponderResult	0	0	0	948868	0
Microsoft-Exchange-Compliance%4DarRuntimeLogs	0	0	0	26	0
Microsoft-Exchange-DxStoreHA%4Server	0	0	0	8	0
Microsoft-Exchange-ESE%4Operational	0	0	0	283	0
Microsoft-Exchange-HighAvailability%4AppLogMirror	0	0	0	482	0
Microsoft-Exchange-HighAvailability%4Debug	0	0	0	153	0
Microsoft-Exchange-HighAvailability%4Monitoring	0	0	0	133	0
Microsoft-Exchange-HighAvailability%4Network	0	0	0	39301	0
Microsoft-Exchange-HighAvailability%4Operational	0	0	0	3782	0
Microsoft-Exchange-HighAvailability%4Seeding	0	0	0	62	0
Microsoft-Exchange-ManagedAvailability%4Monitoring	0	0	0	1609	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionLogs	0	0	0	3846	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionResults	0	0	0	50	0
Microsoft-Exchange-ManagedAvailability%4StartupNotification	0	0	0	122	0
Microsoft-Exchange-ManagedAvailability%4ThrottlingConfig	0	0	0	59	0
Microsoft-Exchange-PushNotifications%4Operational	0	0	0	18	0
Microsoft-Office Server-Search%4Operational	0	0	0	2086	0
Microsoft-Windows-AAD%4Operational	0	49	0	28	26
Microsoft-Windows-AppModel-Runtime%4Admin	0	403	0	80	76
Microsoft-Windows-AppReadiness%4Admin	0	404	0	143	140
Microsoft-Windows-AppReadiness%4Operational	0	22	0	977	957
Microsoft-Windows-AppXDeployment%4Operational	0	36	0	46	44
Microsoft-Windows-AppXDeploymentServer%4Operational	0	2763	0	865	859
Microsoft-Windows-Application Server-Applications%4Operational	0	0	0	109	0
Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant	0	0	0	2	0
Microsoft-Windows-Application-Experience%4Program-Telemetry	6	0	9	35	29
Microsoft-Windows-ApplicationResourceManagementSystem%4Operational	0	0	0	181	174
Microsoft-Windows-AppxPackaging%4Operational	0	140	0	0	0
Microsoft-Windows-BackgroundTaskInfrastructure%4Operational	0	2	0	23	43
Microsoft-Windows-Biometrics%4Operational	0	4	0	48	60
Microsoft-Windows-Bits-Client%4Operational	1591	193	1596	1138	216
Microsoft-Windows-BranchCacheSMB%4Operational	59	0	59	0	0
Microsoft-Windows-CAPI2%4Operational	152	0	152	0	0
Microsoft-Windows-CloudStore%4Operational	0	284	0	0	0
Microsoft-Windows-CodeIntegrity%4Operational	6	17	38	19	17
Microsoft-Windows-Containers-BindFlt%4Operational	0	17	0	0	0
Microsoft-Windows-Containers-Wcifs%4Operational	0	17	0	19	17
Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc	0	15	0	0	0
Microsoft-Windows-Crypto-DPAPI%4Operational	0	268	0	8	8
Microsoft-Windows-Crypto-NCrypt%4Operational	0	93	0	0	0
Microsoft-Windows-DNSServer%4Audit	0	90	0	0	0
Microsoft-Windows-DataIntegrityScan%4Admin	0	40	0	6	6
Microsoft-Windows-DateTimeControlPanel%4Operational	0	0	0	1	0
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Admin	0	1	0	71	99
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Operational	0	2	0	0	0
Microsoft-Windows-DeviceSetupManager%4Admin	0	489	0	696	227
Microsoft-Windows-DeviceSetupManager%4Operational	0	308	0	141	54
Microsoft-Windows-Dhcp-Client%4Admin	8	4	7	1	1
Microsoft-Windows-Dhcpv6-Client%4Admin	34	0	36	0	0
Microsoft-Windows-Diagnosis-DPS%4Operational	200	54	228	31	56
Microsoft-Windows-Diagnosis-PCW%4Operational	0	2195	0	2317	0
Microsoft-Windows-Diagnosis-PLA%4Operational	104	114	130	1661	89
Microsoft-Windows-Diagnosis-Scheduled%4Operational	0	21	14	14	0
Microsoft-Windows-Diagnosis-Scripted%4Admin	1	4	4	2	0
Microsoft-Windows-Diagnosis-Scripted%4Operational	4	16	16	8	0
Microsoft-Windows-Diagnostics-Networking%4Operational	0	4	2	0	0
Microsoft-Windows-Diagnostics-Performance%4Operational	62	0	63	0	0
Microsoft-Windows-DirectoryServices-Deployment%4Operational	0	185	0	0	0
Microsoft-Windows-DriverFrameworks-UserMode%4Operational	2	0	2	0	0
Microsoft-Windows-Fault-Tolerant-Heap%4Operational	2	0	2	6	2
Microsoft-Windows-FileServices-ServerManager-EventProvider%4Operational	0	0	0	3	0
Microsoft-Windows-Forwarding%4Operational	0	13	0	2	2
Microsoft-Windows-GroupPolicy%4Operational	1490	6832	2426	7963	7930
Microsoft-Windows-HelloForBusiness%4Operational	0	57	0	0	0
Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin	0	34	0	0	0
Microsoft-Windows-International%4Operational	1	0	1	1	2
Microsoft-Windows-Kernel-Boot%4Operational	0	68	0	0	0
Microsoft-Windows-Kernel-Cache%4Operational	0	832	0	0	0
Microsoft-Windows-Kernel-EventTracing%4Admin	3	1	43	37	7
Microsoft-Windows-Kernel-IO%4Operational	0	2519	0	1588	1540
Microsoft-Windows-Kernel-PnP%4Configuration	0	512	0	369	243
Microsoft-Windows-Kernel-PnP%4Device Management	0	214	0	0	0
Microsoft-Windows-Kernel-ShimEngine%4Operational	0	4	0	4	4
Microsoft-Windows-Kernel-WHEA%4Operational	47	73	48	19	17
Microsoft-Windows-Known Folders API Service	157	1743	186	1910	589
Microsoft-Windows-LanguagePackSetup%4Operational	32	18	38	10	8
Microsoft-Windows-LiveId%4Operational	0	496	0	597	399
Microsoft-Windows-MUI%4Operational	108	12	108	13	13
Microsoft-Windows-NCSI%4Operational	0	39	0	32	20
Microsoft-Windows-NetworkLocationWizard%4Operational	1	0	1	0	0
Microsoft-Windows-NetworkProfile%4Operational	274	164	284	123	105
Microsoft-Windows-Ntfs%4Operational	0	4359	0	833	820
Microsoft-Windows-Ntfs%4WHC	0	17	0	19	17
Microsoft-Windows-OfflineFiles%4Operational	88	0	89	0	0
Microsoft-Windows-Partition%4Diagnostic	0	31	0	0	0
Microsoft-Windows-PowerShell%4Operational	0	859	0	25132	206
Microsoft-Windows-PrintService%4Admin	32	22	44	2	2
Microsoft-Windows-Privacy-Auditing%4Operational	0	163	0	0	0
Microsoft-Windows-PushNotification-Platform%4Operational	0	406	0	1768	1736
Microsoft-Windows-ReadyBoost%4Operational	100	0	108	0	0
Microsoft-Windows-ReliabilityAnalysisComponent%4Operational	23	0	23	0	0
Microsoft-Windows-RemoteAssistance%4Operational	4	0	4	0	0
Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational	0	1950	0	1935	1233
Microsoft-Windows-RemoteDesktopServices-SessionServices%4Operational	0	0	0	1	5
Microsoft-Windows-Resource-Exhaustion-Detector%4Operational	76	37	83	32	27
Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational	12	13	16	8	15
Microsoft-Windows-SMBServer%4Connectivity	0	0	0	1	0
Microsoft-Windows-SMBServer%4Operational	0	44	0	84	73
Microsoft-Windows-SMBServer%4Security	0	37	0	1	1
Microsoft-Windows-Security-Mitigations%4KernelMode	0	8	0	0	0
Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter	0	6	0	6	6
Microsoft-Windows-ServerManager-DeploymentProvider%4Operational	0	539	0	340	159
Microsoft-Windows-ServerManager-MgmtProvider%4Operational	0	1930	0	429	420
Microsoft-Windows-ServerManager-MultiMachine%4Operational	0	1992	0	1959	843
Microsoft-Windows-SettingSync%4Debug	0	0	0	1102	1141
Microsoft-Windows-SettingSync%4Operational	0	0	0	9	9
Microsoft-Windows-Shell-Core%4AppDefaults	0	966	0	0	0
Microsoft-Windows-Shell-Core%4Operational	0	2199	0	1317	1178
Microsoft-Windows-ShellCommon-StartLayoutPopulation%4Operational	0	211	0	0	0
Microsoft-Windows-SmartCard-DeviceEnum%4Operational	0	0	0	101	42
Microsoft-Windows-SmbClient%4Connectivity	0	88	0	181	141
Microsoft-Windows-SmbClient%4Security	0	0	0	3	0
Microsoft-Windows-StateRepository%4Operational	0	807	0	69	63
Microsoft-Windows-Storage-ClassPnP%4Operational	0	110	0	77	65
Microsoft-Windows-Storage-Storport%4Operational	0	864	0	615	618
Microsoft-Windows-StorageManagement%4Operational	0	30	0	0	0
Microsoft-Windows-StorageSpaces-Driver%4Operational	0	31	0	35	31
Microsoft-Windows-Store%4Operational	0	456	0	1137	761
Microsoft-Windows-Storsvc%4Diagnostic	0	47	0	0	0
Microsoft-Windows-SystemDataArchiver%4Diagnostic	0	489562	0	0	0
Microsoft-Windows-TWinUI%4Operational	0	4	0	73	84

Table B.2: Absolute Size per Log Type for each Host in Demo-Case 2 (cont.)

Log Event Counts per Host (Part 2) [demo_case2]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Microsoft-Windows-TZSync%4Operational	0	8	0	12	16
Microsoft-Windows-TaskScheduler%4Maintenance	0	174	0	52	40
Microsoft-Windows-TaskScheduler%4Operational	1102	19533	2577	6291	6743
Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational	114	214	141	203	130
Microsoft-Windows-TerminalServices-PnPDevices%4Admin	1	0	1	0	0
Microsoft-Windows-TerminalServices-Printers%4Admin	0	32	0	36	15
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin	0	11	0	14	7
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational	307	321	1943	124	80
Microsoft-Windows-Time-Service%4Operational	0	416	0	0	0
Microsoft-Windows-UAC-FileVirtualization%4Operational	0	0	0	1198	0
Microsoft-Windows-UniversalTelemetryClient%4Operational	0	760	0	110	86
Microsoft-Windows-User Device Registration%4Admin	0	386	0	16	0
Microsoft-Windows-User Profile Service%4Operational	182	96	228	81	61
Microsoft-Windows-UserPnp%4DeviceInstall	0	15	0	9	7
Microsoft-Windows-VHDMP-Operational	0	0	0	2	0
Microsoft-Windows-VolumeSnapshot-Driver%4Operational	0	174	0	194	174
Microsoft-Windows-WER-Diag%4Operational	1	0	1	0	0
Microsoft-Windows-WER-PayloadHealth%4Operational	0	10	0	0	0
Microsoft-Windows-WFP%4Operational	0	2	0	2	0
Microsoft-Windows-WMI-Activity%4Operational	0	1489	0	1696	1619
Microsoft-Windows-Wcmsvc%4Operational	0	191	0	237	177
Microsoft-Windows-WebAuthN%4Operational	0	30	0	0	0
Microsoft-Windows-WinINet-Config%4ProxyConfigChanged	0	4	0	2	2
Microsoft-Windows-WinRM%4Operational	0	454	0	2329	278
Microsoft-Windows-Windows Defender%4Operational	2	878	2	66	58
Microsoft-Windows-Windows Defender%4WHC	103	0	109	67	58
Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	390	725	423	473	327
Microsoft-Windows-WindowsBackup%4ActionCenter	11	0	15	0	0
Microsoft-Windows-WindowsSystemAssessmentTool%4Operational	39	0	39	0	0
Microsoft-Windows-WindowsUpdateClient%4Operational	1036	173	1045	54	44
Microsoft-Windows-Winlogon%4Operational	0	988	0	0	0
Microsoft-WindowsAzure-Diagnostics%4GuestAgent	1769	1457	1789	1444	1204
Microsoft-WindowsAzure-Diagnostics%4Heartbeat	1589	1551	1569	1493	1488
Microsoft-WindowsAzure-Status%4GuestAgent	111	1380	227	0	0
Microsoft-WindowsAzure-Status%4Plugins	13	1216	15	17	15
Security	919	167822	1201	29182	4770
Setup	0	29	0	107	0
System	2122	8504	2578	15104	4632
Windows PowerShell	48	1124	93	920	164

Table B.3: Absolute Size per Log Type for each Host in Demo-Case 3

Log Event Counts per Host (Part 1) [demo_case3]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Active Directory Web Services	0	106	0	0	0
Application	368	12109	509	4457	0
DFS Replication	0	97	0	0	0
DNS Server	0	89	0	0	0
Directory Service	0	2467	0	0	0
MSExchange Management	0	0	0	94	0
Microsoft-Client-Licensing-Platform%4Admin	0	117	0	47	35
Microsoft-Exchange-ActiveMonitoring%4MaintenanceResult	0	0	0	53256	0
Microsoft-Exchange-ActiveMonitoring%4MonitorDefinition	0	0	0	16113	0
Microsoft-Exchange-ActiveMonitoring%4MonitorResult	0	0	0	690154	0
Microsoft-Exchange-ActiveMonitoring%4ProbeDefinition	0	0	0	998	0
Microsoft-Exchange-ActiveMonitoring%4ProbeResult	0	0	0	495129	0
Microsoft-Exchange-ActiveMonitoring%4ResponderDefinition	0	0	0	13458	0
Microsoft-Exchange-ActiveMonitoring%4ResponderResult	0	0	0	948324	0
Microsoft-Exchange-Compliance%4DarRuntimeLogs	0	0	0	24	0
Microsoft-Exchange-Compliance%4DiscoveryLogs	0	0	0	331	0
Microsoft-Exchange-DxStoreHA%4Server	0	0	0	8	0
Microsoft-Exchange-ESE%4Operational	0	0	0	214	0
Microsoft-Exchange-HighAvailability%4AppLogMirror	0	0	0	875	0
Microsoft-Exchange-HighAvailability%4Debug	0	0	0	135	0
Microsoft-Exchange-HighAvailability%4Monitoring	0	0	0	239	0
Microsoft-Exchange-HighAvailability%4Network	0	0	0	39300	0
Microsoft-Exchange-HighAvailability%4Operational	0	0	0	3854	0
Microsoft-Exchange-HighAvailability%4Seeding	0	0	0	90	0
Microsoft-Exchange-ManagedAvailability%4Monitoring	0	0	0	1318	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionLogs	0	0	0	4083	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionResults	0	0	0	42	0
Microsoft-Exchange-ManagedAvailability%4StartupNotification	0	0	0	95	0
Microsoft-Exchange-ManagedAvailability%4ThrottlingConfig	0	0	0	59	0
Microsoft-Exchange-PushNotifications%4Operational	0	0	0	15	0
Microsoft-Office Server-Search%4Operational	0	0	0	2109	0
Microsoft-Windows-AAD%4Operational	0	38	0	24	24
Microsoft-Windows-AppModel-Runtime%4Admin	0	295	0	80	76
Microsoft-Windows-AppReadiness%4Admin	0	286	0	143	140
Microsoft-Windows-AppReadiness%4Operational	0	16	0	977	957
Microsoft-Windows-AppXDeployment%4Operational	0	28	0	46	44
Microsoft-Windows-AppXDeploymentServer%4Operational	0	2084	0	865	859
Microsoft-Windows-Application Server-Applications%4Operational	0	0	0	112	0
Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant	0	0	0	2	0
Microsoft-Windows-Application-Experience%4Program-Telemetry	6	0	9	33	26
Microsoft-Windows-ApplicationResourceManagementSystem%4Operational	0	0	0	182	176
Microsoft-Windows-AppxPackaging%4Operational	0	136	0	0	0
Microsoft-Windows-BackgroundTaskInfrastructure%4Operational	0	2	0	24	44
Microsoft-Windows-Biometrics%4Operational	0	4	0	36	52
Microsoft-Windows-Bits-Client%4Operational	1598	106	1596	1136	221
Microsoft-Windows-BranchCacheSMB%4Operational	59	0	59	0	0
Microsoft-Windows-CAPI2%4Operational	152	0	152	0	0
Microsoft-Windows-CloudStore%4Operational	0	182	0	0	0
Microsoft-Windows-CodeIntegrity%4Operational	11	16	38	17	16
Microsoft-Windows-Containers-BindFlt%4Operational	0	16	0	0	0
Microsoft-Windows-Containers-Wcifs%4Operational	0	16	0	17	16
Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc	0	14	0	0	0
Microsoft-Windows-Crypto-DPAPI%4Operational	0	231	0	8	8
Microsoft-Windows-Crypto-NCrypt%4Operational	0	81	0	0	0
Microsoft-Windows-DNSServer%4Audit	0	82	0	0	0
Microsoft-Windows-DataIntegrityScan%4Admin	0	40	0	6	6

Table B.3: Absolute Size per Log Type for each Host in Demo-Case 3 (cont.)

Log Event Counts per Host (Part 2) [demo_case3]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Microsoft-Windows-DateTimeControlPanel%4Operational	0	0	0	1	0
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Admin	0	1	0	71	99
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Operational	0	2	0	0	0
Microsoft-Windows-DeviceSetupManager%4Admin	0	475	0	686	222
Microsoft-Windows-DeviceSetupManager%4Operational	0	304	0	152	57
Microsoft-Windows-Dhcp-Client%4Admin	8	4	7	1	1
Microsoft-Windows-Dhcpv6-Client%4Admin	30	0	33	0	0
Microsoft-Windows-Diagnosis-DPS%4Operational	200	44	222	33	56
Microsoft-Windows-Diagnosis-PCW%4Operational	0	2199	0	2273	1992
Microsoft-Windows-Diagnosis-PLA%4Operational	70	110	106	1580	85
Microsoft-Windows-Diagnosis-Scheduled%4Operational	0	21	7	14	0
Microsoft-Windows-Diagnosis-Scripted%4Admin	2	5	3	2	0
Microsoft-Windows-Diagnosis-Scripted%4Operational	8	20	12	8	0
Microsoft-Windows-Diagnostics-Networking%4Operational	2	8	2	0	0
Microsoft-Windows-Diagnostics-Performance%4Operational	62	0	63	0	0
Microsoft-Windows-DirectoryServices-Deployment%4Operational	0	185	0	0	0
Microsoft-Windows-DriverFrameworks-UserMode%4Operational	2	0	2	0	0
Microsoft-Windows-Fault-Tolerant-Heap%4Operational	2	0	2	6	2
Microsoft-Windows-FileServices-ServerManager-EventProvider%4Operational	0	52	0	3	0
Microsoft-Windows-Forwarding%4Operational	0	13	0	2	2
Microsoft-Windows-GroupPolicy%4Operational	1416	6786	2359	8023	8043
Microsoft-Windows-HelloForBusiness%4Operational	0	48	0	0	0
Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin	0	32	0	0	0
Microsoft-Windows-International%4Operational	1	0	1	1	2
Microsoft-Windows-Kernel-Boot%4Operational	0	64	0	0	0
Microsoft-Windows-Kernel-Cache%4Operational	0	827	0	0	0
Microsoft-Windows-Kernel-EventTracing%4Admin	3	1	43	36	7
Microsoft-Windows-Kernel-IO%4Operational	0	2528	0	1536	1484
Microsoft-Windows-Kernel-PnP%4Configuration	0	501	0	381	240
Microsoft-Windows-Kernel-PnP%4Device Management	0	211	0	0	0
Microsoft-Windows-Kernel-ShimEngine%4Operational	0	3	0	3	3
Microsoft-Windows-Kernel-WHEA%4Operational	43	68	45	17	16
Microsoft-Windows-Known Folders API Service	139	1738	174	1896	581
Microsoft-Windows-LanguagePackSetup%4Operational	30	18	34	10	8
Microsoft-Windows-LiveId%4Operational	0	537	0	588	378
Microsoft-Windows-MUI%4Operational	108	12	108	13	13
Microsoft-Windows-NCSI%4Operational	0	38	0	28	16
Microsoft-Windows-NetworkLocationWizard%4Operational	1	0	1	0	0
Microsoft-Windows-NetworkProfile%4Operational	251	157	268	115	94
Microsoft-Windows-Ntfs%4Operational	0	4382	0	820	818
Microsoft-Windows-Ntfs%4WHC	0	16	0	17	16
Microsoft-Windows-OfflineFiles%4Operational	84	0	86	0	0
Microsoft-Windows-Partition%4Diagnostic	0	29	0	0	0
Microsoft-Windows-PowerShell%4Operational	0	770	0	24954	206
Microsoft-Windows-PrintService%4Admin	40	19	42	2	2
Microsoft-Windows-Privacy-Auditing%4Operational	0	118	0	0	0
Microsoft-Windows-PushNotification-Platform%4Operational	0	317	0	1753	1733
Microsoft-Windows-ReadyBoost%4Operational	92	0	101	0	0
Microsoft-Windows-ReliabilityAnalysisComponent%4Operational	23	0	23	0	0
Microsoft-Windows-RemoteAssistance%4Operational	4	0	4	0	0
Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational	0	1942	0	1960	1293
Microsoft-Windows-RemoteDesktopServices-SessionServices%4Operational	0	0	0	0	5
Microsoft-Windows-Resource-Exhaustion-Detector%4Operational	71	32	80	30	24
Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational	12	6	16	8	15
Microsoft-Windows-SMBServer%4Connectivity	0	0	0	1	0
Microsoft-Windows-SMBServer%4Operational	0	41	0	74	69
Microsoft-Windows-SMBServer%4Security	0	37	0	1	1
Microsoft-Windows-Security-Mitigations%4KernelMode	0	6	0	0	0
Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter	0	6	0	6	6
Microsoft-Windows-ServerManager-DeploymentProvider%4Operational	0	509	0	352	162
Microsoft-Windows-ServerManager-MgmtProvider%4Operational	0	1925	0	428	441
Microsoft-Windows-ServerManager-MultiMachine%4Operational	0	1987	0	1979	873
Microsoft-Windows-SettingSync%4Debug	0	0	0	1104	1141
Microsoft-Windows-SettingSync%4Operational	0	0	0	9	9
Microsoft-Windows-Shell-Core%4AppDefaults	0	680	0	0	0
Microsoft-Windows-Shell-Core%4Operational	0	1806	0	1387	1178
Microsoft-Windows-ShellCommon-StartLayoutPopulation%4Operational	0	142	0	0	0
Microsoft-Windows-SmartCard-DeviceEnum%4Operational	0	0	0	107	42
Microsoft-Windows-SmbClient%4Connectivity	0	110	0	171	142
Microsoft-Windows-SmbClient%4Security	0	0	0	3	0
Microsoft-Windows-StateRepository%4Operational	0	722	0	65	61
Microsoft-Windows-Storage-ClassPnP%4Operational	0	102	0	73	61
Microsoft-Windows-Storage-Storport%4Operational	0	863	0	619	622
Microsoft-Windows-StorageManagement%4Operational	0	67	0	0	0
Microsoft-Windows-StorageSpaces-Driver%4Operational	0	29	0	31	29
Microsoft-Windows-Store%4Operational	0	449	0	1120	776
Microsoft-Windows-Storsvc%4Diagnostic	0	47	0	0	0
Microsoft-Windows-SystemDataArchiver%4Diagnostic	0	489613	0	0	0
Microsoft-Windows-TWinUI%4Operational	0	3	0	78	86
Microsoft-Windows-TZSync%4Operational	0	8	0	12	16
Microsoft-Windows-TaskScheduler%4Maintenance	0	174	0	52	40
Microsoft-Windows-TaskScheduler%4Operational	1017	19567	2443	6147	6703
Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational	119	203	136	205	129
Microsoft-Windows-TerminalServices-PnPDevices%4Admin	1	0	0	0	0
Microsoft-Windows-TerminalServices-Printers%4Admin	0	34	0	38	16
Microsoft-Windows-TerminalServices-RDPClient%4Operational	0	14	0	0	0
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin	0	8	0	15	7
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational	86	300	91	119	82
Microsoft-Windows-Time-Service%4Operational	0	411	0	0	0
Microsoft-Windows-UAC-FileVirtualization%4Operational	0	0	0	1198	0
Microsoft-Windows-UniversalTelemetryClient%4Operational	0	750	0	102	82
Microsoft-Windows-User Device Registration%4Admin	0	372	0	16	0
Microsoft-Windows-User Profile Service%4Operational	186	79	224	81	61
Microsoft-Windows-UserPnp%4DeviceInstall	0	12	0	7	5
Microsoft-Windows-VHDMP-Operational	0	0	0	2	0
Microsoft-Windows-VolumeSnapshot-Driver%4Operational	0	162	0	172	162
Microsoft-Windows-WER-Diag%4Operational	1	0	1	0	0
Microsoft-Windows-WER-PayloadHealth%4Operational	0	8	0	0	0
Microsoft-Windows-WFP%4Operational	0	2	0	3	0
Microsoft-Windows-WMI-Activity%4Operational	0	1471	0	1716	1604
Microsoft-Windows-Wcmsvc%4Operational	0	179	0	224	170
Microsoft-Windows-WebAuthN%4Operational	0	29	0	0	0
Microsoft-Windows-WinINet-Config%4ProxyConfigChanged	0	3	0	2	2
Microsoft-Windows-WinRM%4Operational	0	430	0	2440	299
Microsoft-Windows-Windows Defender%4Operational	2	882	2	70	58
Microsoft-Windows-Windows Defender%4WHC	90	0	99	62	56
Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	369	590	406	459	327
Microsoft-Windows-WindowsBackup%4ActionCenter	11	0	15	0	0
Microsoft-Windows-WindowsSystemAssessmentTool%4Operational	39	0	39	0	0
Microsoft-Windows-WindowsUpdateClient%4Operational	1021	173	1034	55	45
Microsoft-Windows-Winlogon%4Operational	0	902	0	0	0
Microsoft-WindowsAzure-Diagnostics%4GuestAgent	1951	1453	1896	1709	1633
Microsoft-WindowsAzure-Diagnostics%4Heartbeat	1624	1561	1640	1537	1475
Microsoft-WindowsAzure-Status%4GuestAgent	119	1392	234	0	0
Microsoft-WindowsAzure-Status%4Plugins	9	1276	12	15	14
Security	796	156698	1060	29143	4697
Setup	0	29	0	107	0
System	1627	8177	2166	0	4481
Windows PowerShell	48	585	93	1044	164

Table B.4: Absolute Size per Log Type for each Host in Demo-Case 4

Log Event Counts per Host (Part 1) [demo_case4]

EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Active Directory Web Services	0	106	0	0	0
Application	348	12091	451	4457	2572
DFS Replication	0	97	0	0	0
DNS Server	0	89	0	0	0
Directory Service	0	2467	0	0	0
HardwareEvents	0	0	0	0	0
MSExchange Management	0	0	0	94	0
Microsoft-Client-Licensing-Platform%4Admin	0	117	0	47	30
Microsoft-Exchange-ActiveMonitoring%4MaintenanceResult	0	0	0	53256	0
Microsoft-Exchange-ActiveMonitoring%4MonitorDefinition	0	0	0	16113	0
Microsoft-Exchange-ActiveMonitoring%4MonitorResult	0	0	0	690154	0
Microsoft-Exchange-ActiveMonitoring%4ProbeDefinition	0	0	0	998	0
Microsoft-Exchange-ActiveMonitoring%4ProbeResult	0	0	0	495129	0
Microsoft-Exchange-ActiveMonitoring%4ResponderDefinition	0	0	0	13458	0
Microsoft-Exchange-ActiveMonitoring%4ResponderResult	0	0	0	948324	0
Microsoft-Exchange-Compliance%4DarRuntimeLogs	0	0	0	24	0
Microsoft-Exchange-Compliance%4DiscoveryLogs	0	0	0	331	0
Microsoft-Exchange-DxStoreHA%4Server	0	0	0	8	0
Microsoft-Exchange-ESE%4Operational	0	0	0	214	0
Microsoft-Exchange-HighAvailability%4AppLogMirror	0	0	0	875	0
Microsoft-Exchange-HighAvailability%4Debug	0	0	0	135	0
Microsoft-Exchange-HighAvailability%4Monitoring	0	0	0	239	0
Microsoft-Exchange-HighAvailability%4Network	0	0	0	39300	0
Microsoft-Exchange-HighAvailability%4Operational	0	0	0	3854	0
Microsoft-Exchange-HighAvailability%4Seeding	0	0	0	90	0
Microsoft-Exchange-ManagedAvailability%4Monitoring	0	0	0	1318	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionLogs	0	0	0	4083	0
Microsoft-Exchange-ManagedAvailability%4RecoveryActionResults	0	0	0	42	0
Microsoft-Exchange-ManagedAvailability%4StartupNotification	0	0	0	95	0
Microsoft-Exchange-ManagedAvailability%4ThrottlingConfig	0	0	0	59	0
Microsoft-Exchange-PushNotifications%4Operational	0	0	0	15	0
Microsoft-Office Server-Search%4Operational	0	0	0	2109	0
Microsoft-Windows-AAD%4Operational	0	38	0	24	16
Microsoft-Windows-AppModel-Runtime%4Admin	0	295	0	80	76
Microsoft-Windows-AppReadiness%4Admin	0	286	0	143	140
Microsoft-Windows-AppReadiness%4Operational	0	16	0	977	957
Microsoft-Windows-AppXDeployment%4Operational	0	28	0	46	44
Microsoft-Windows-AppXDeploymentServer%4Operational	0	2084	0	865	0
Microsoft-Windows-Application Server-Applications%4Operational	0	0	0	112	0
Microsoft-Windows-Application-Experience%4Problem-Steps-Recorder	0	0	0	0	0
Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant	0	0	0	2	0
Microsoft-Windows-Application-Experience%4Program-Compatibility-Troubleshooter	0	0	0	0	0
Microsoft-Windows-Application-Experience%4Program-Inventory	0	0	0	0	0
Microsoft-Windows-Application-Experience%4Program-Telemetry	6	0	8	33	26
Microsoft-Windows-ApplicationResourceManagementSystem%4Operational	0	0	0	182	174
Microsoft-Windows-AppxPackaging%4Operational	0	136	0	0	0
Microsoft-Windows-BackgroundTaskInfrastructure%4Operational	0	2	0	24	41
Microsoft-Windows-Backup	0	0	0	0	0
Microsoft-Windows-Biometrics%4Operational	0	4	0	36	32
Microsoft-Windows-Bits-Client%4Operational	1597	106	1596	1136	214
Microsoft-Windows-BranchCacheSMB%4Operational	59	0	59	0	0
Microsoft-Windows-CAPI2%4Operational	152	0	152	0	0
Microsoft-Windows-CloudStore%4Operational	0	182	0	0	0
Microsoft-Windows-CodeIntegrity%4Operational	12	16	38	17	12
Microsoft-Windows-Compat-Appraiser%4Operational	0	0	0	0	0
Microsoft-Windows-Containers-BindFlt%4Operational	0	16	0	0	0
Microsoft-Windows-Containers-Wcifs%4Operational	0	16	0	17	12
Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc	0	14	0	0	0
Microsoft-Windows-Crypto-DPAPI%4Operational	0	231	0	8	8
Microsoft-Windows-Crypto-NCrypt%4Operational	0	81	0	0	0
Microsoft-Windows-DNSServer%4Audit	0	82	0	0	0
Microsoft-Windows-DataIntegrityScan%4Admin	0	40	0	6	6
Microsoft-Windows-DateTimeControlPanel%4Operational	0	0	0	1	0
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Admin	0	1	0	71	99
Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Operational	0	2	0	0	0
Microsoft-Windows-DeviceSetupManager%4Admin	0	475	0	686	212
Microsoft-Windows-DeviceSetupManager%4Operational	0	304	0	152	55
Microsoft-Windows-Dhcp-Client%4Admin	8	4	7	1	1
Microsoft-Windows-Dhcpv6-Client%4Admin	29	0	31	0	0
Microsoft-Windows-Diagnosis-DPS%4Operational	198	44	218	33	56
Microsoft-Windows-Diagnosis-PCW%4Operational	0	2199	0	2273	1416
Microsoft-Windows-Diagnosis-PLA%4Operational	66	110	86	1580	60
Microsoft-Windows-Diagnosis-Scheduled%4Operational	0	21	7	14	0
Microsoft-Windows-Diagnosis-Scripted%4Admin	1	5	3	2	0
Microsoft-Windows-Diagnosis-Scripted%4Operational	4	20	12	8	0
Microsoft-Windows-Diagnostics-Networking%4Operational	0	8	2	0	0
Microsoft-Windows-Diagnostics-Performance%4Operational	62	0	63	0	0
Microsoft-Windows-DirectoryServices-Deployment%4Operational	0	185	0	0	0
Microsoft-Windows-DiskDiagnosticDataCollector%4Operational	0	0	0	0	0
Microsoft-Windows-DriverFrameworks-UserMode%4Operational	2	0	2	0	0
Microsoft-Windows-Fault-Tolerant-Heap%4Operational	2	0	2	6	2
Microsoft-Windows-FileServices-ServerManager-EventProvider%4Operational	0	52	0	3	0
Microsoft-Windows-Forwarding%4Operational	0	13	0	2	2
Microsoft-Windows-GroupPolicy%4Operational	1633	6786	2344	8023	7917
Microsoft-Windows-HelloForBusiness%4Operational	0	48	0	0	0
Microsoft-Windows-Hyper-V-Guest-Drivers%4Admin	0	32	0	0	0
Microsoft-Windows-International%4Operational	1	0	1	1	2
Microsoft-Windows-Kernel-Boot%4Operational	0	64	0	0	0
Microsoft-Windows-Kernel-Cache%4Operational	0	827	0	0	0
Microsoft-Windows-Kernel-EventTracing%4Admin	23	1	43	36	7
Microsoft-Windows-Kernel-IO%4Operational	0	2528	0	1536	1246
Microsoft-Windows-Kernel-PnP%4Configuration	0	501	0	381	237
Microsoft-Windows-Kernel-PnP%4Device Management	0	211	0	0	0
Microsoft-Windows-Kernel-Power%4Thermal-Operational	0	0	0	0	0
Microsoft-Windows-Kernel-ShimEngine%4Operational	0	3	0	3	2
Microsoft-Windows-Kernel-StoreMgr%4Operational	0	0	0	0	0
Microsoft-Windows-Kernel-WDI%4Operational	0	0	0	0	0
Microsoft-Windows-Kernel-WHEA%4Errors	0	0	0	0	0
Microsoft-Windows-Kernel-WHEA%4Operational	42	68	43	17	12
Microsoft-Windows-Known Folders API Service	206	1738	184	1896	619
Microsoft-Windows-LanguagePackSetup%4Operational	28	18	32	10	8
Microsoft-Windows-LiveId%4Operational	0	537	0	588	350
Microsoft-Windows-MUI%4Admin	0	0	0	0	0
Microsoft-Windows-MUI%4Operational	108	12	108	13	13
Microsoft-Windows-NCSI%4Operational	0	38	0	28	14
Microsoft-Windows-NetworkAccessProtection%4Operational	0	0	0	0	0
Microsoft-Windows-NetworkAccessProtection%4WHC	0	0	0	0	0
Microsoft-Windows-NetworkLocationWizard%4Operational	1	0	1	0	0
Microsoft-Windows-NetworkProfile%4Operational	254	157	259	115	80
Microsoft-Windows-Ntfs%4Operational	0	4382	0	820	786
Microsoft-Windows-Ntfs%4WHC	0	16	0	17	12
Microsoft-Windows-OfflineFiles%4Operational	83	0	84	0	0
Microsoft-Windows-Partition%4Diagnostic	0	29	0	0	0
Microsoft-Windows-PowerShell%4Operational	0	770	0	24954	200
Microsoft-Windows-PrintService%4Admin	43	19	42	2	2
Microsoft-Windows-Privacy-Auditing%4Operational	0	118	0	0	0
Microsoft-Windows-PushNotification-Platform%4Operational	0	317	0	1753	1703
Microsoft-Windows-ReadyBoost%4Operational	90	0	96	0	0
Microsoft-Windows-ReliabilityAnalysisComponent%4Operational	23	0	23	0	0
Microsoft-Windows-RemoteAssistance%4Admin	0	0	0	0	0
Microsoft-Windows-RemoteAssistance%4Operational	4	0	4	0	0
Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational	0	1942	0	1960	1110
Microsoft-Windows-RemoteDesktopServices-SessionServices%4Operational	0	0	0	0	5
Microsoft-Windows-Resource-Exhaustion-Detector%4Operational	70	32	76	30	21
Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational	12	6	16	8	15
Microsoft-Windows-RestartManager%4Operational	0	0	0	0	0
Microsoft-Windows-SMBServer%4Audit	0	0	0	0	0
Microsoft-Windows-SMBServer%4Connectivity	0	0	0	1	0
Microsoft-Windows-SMBServer%4Operational	0	41	0	74	53
Microsoft-Windows-SMBServer%4Security	0	37	0	1	1
Microsoft-Windows-Security-Mitigations%4KernelMode	0	6	0	0	0
Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter	0	6	0	6	6
Microsoft-Windows-ServerManager-DeploymentProvider%4Operational	0	509	0	352	162
Microsoft-Windows-ServerManager-MgmtProvider%4Operational	0	1925	0	428	441
Microsoft-Windows-ServerManager-MultiMachine%4Operational	0	1987	0	1979	878

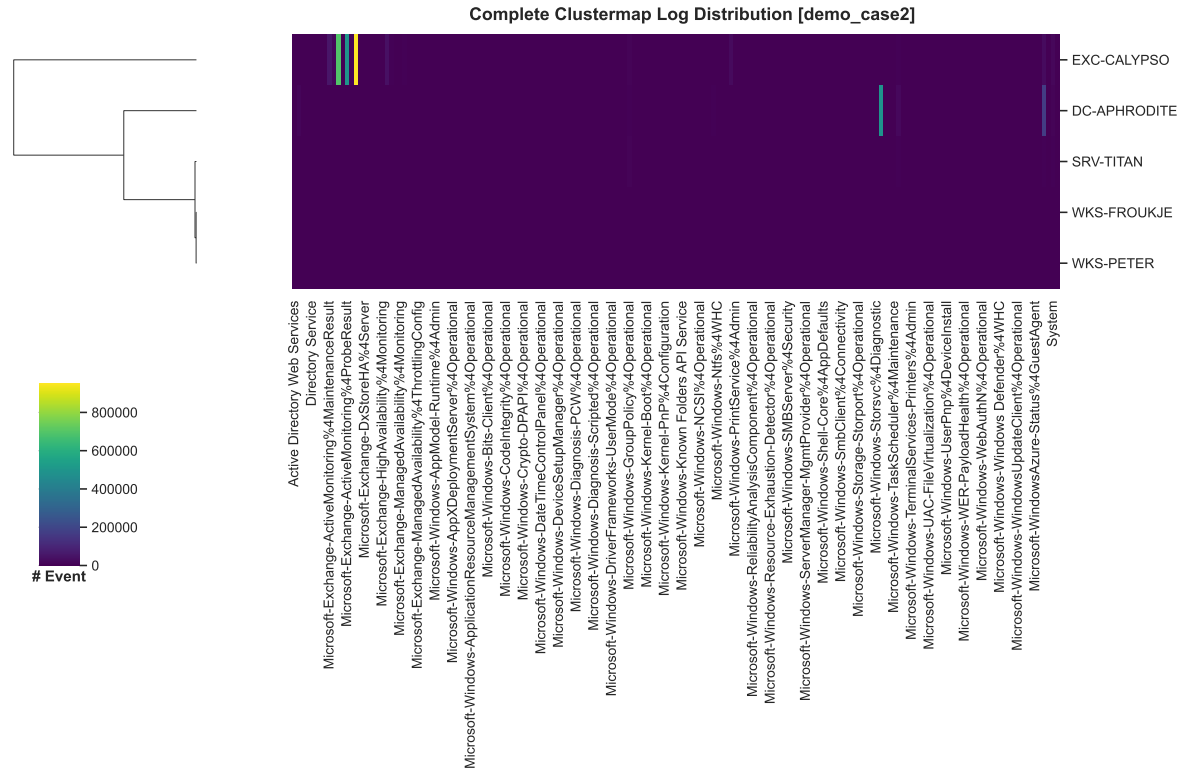
Table B.4: Absolute Size per Log Type for each Host in Demo-Case 4 (cont.)

Log Event Counts per Host (Part 2) [demo_case4]

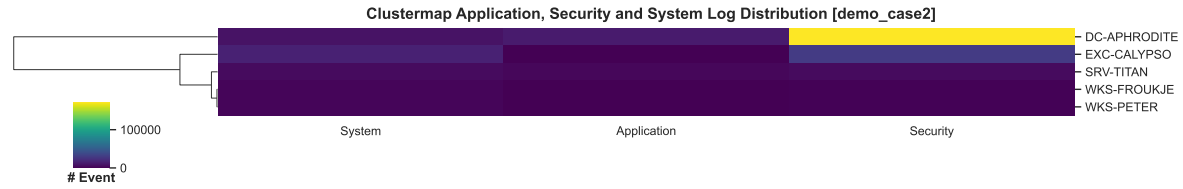
EventLog	WKS-FROUKJE	DC-APHRODITE	WKS-PETER	EXC-CALYPSO	SRV-TITAN
Microsoft-Windows-SettingSync%4Debug	0	0	0	1104	1141
Microsoft-Windows-SettingSync%4Operational	0	0	0	9	9
Microsoft-Windows-Shell-Core%4AppDefaults	0	680	0	0	0
Microsoft-Windows-Shell-Core%4Operational	0	1806	0	1387	1174
Microsoft-Windows-ShellCommon-StartLayoutPopulation%4Operational	0	142	0	0	0
Microsoft-Windows-SmartCard-DeviceEnum%4Operational	0	0	0	107	42
Microsoft-Windows-SmbClient%4Connectivity	0	110	0	171	95
Microsoft-Windows-SmbClient%4Security	0	0	0	3	0
Microsoft-Windows-StateRepository%4Operational	0	722	0	65	52
Microsoft-Windows-Storage-ClassPnP%4Operational	0	102	0	73	55
Microsoft-Windows-Storage-Storport%4Operational	0	863	0	619	620
Microsoft-Windows-StorageManagement%4Operational	0	67	0	0	0
Microsoft-Windows-StorageSpaces-Driver%4Operational	0	29	0	31	21
Microsoft-Windows-Store%4Operational	0	449	0	1120	775
Microsoft-Windows-Storsvc%4Diagnostic	0	47	0	0	0
Microsoft-Windows-SystemDataArchiver%4Diagnostic	0	489613	0	0	0
Microsoft-Windows-TWinUI%4Operational	0	3	0	78	85
Microsoft-Windows-TZSync%4Operational	0	8	0	12	12
Microsoft-Windows-TaskScheduler%4Maintenance	0	174	0	52	36
Microsoft-Windows-TaskScheduler%4Operational	1089	19567	2211	6147	6445
Microsoft-Windows-TerminalServices-LocalSessionManager%4Admin	0	0	0	0	0
Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational	124	203	134	205	123
Microsoft-Windows-TerminalServices-PnPDevices%4Admin	1	0	0	0	0
Microsoft-Windows-TerminalServices-PnPDevices%4Operational	0	0	0	0	0
Microsoft-Windows-TerminalServices-Printers%4Admin	0	34	0	38	16
Microsoft-Windows-TerminalServices-RDPClient%4Operational	0	14	0	0	0
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin	0	8	0	15	7
Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational	81	300	85	119	61
Microsoft-Windows-Time-Service%4Operational	0	411	0	0	0
Microsoft-Windows-UAC%4Operational	0	0	0	0	0
Microsoft-Windows-UAC-FileVirtualization%4Operational	0	0	0	1198	0
Microsoft-Windows-UniversalTelemetryClient%4Operational	0	750	0	102	67
Microsoft-Windows-User Device Registration%4Admin	0	372	0	16	0
Microsoft-Windows-User Profile Service%4Operational	192	79	224	81	63
Microsoft-Windows-UserPnp%4DeviceInstall	0	12	0	7	3
Microsoft-Windows-VHDMP-Operational	0	0	0	2	0
Microsoft-Windows-VolumeSnapshot-Driver%4Operational	0	162	0	172	120
Microsoft-Windows-WER-Diag%4Operational	1	0	1	0	0
Microsoft-Windows-WER-PayloadHealth%4Operational	0	8	0	0	0
Microsoft-Windows-WFP%4Operational	0	2	0	3	0
Microsoft-Windows-WMI-Activity%4Operational	0	1471	0	1716	1624
Microsoft-Windows-Wcmsvc%4Operational	0	179	0	224	137
Microsoft-Windows-WebAuthN%4Operational	0	29	0	0	0
Microsoft-Windows-WinINet-Config%4ProxyConfigChanged	0	3	0	2	2
Microsoft-Windows-WinRM%4Operational	0	430	0	2440	258
Microsoft-Windows-Windows Defender%4Operational	2	882	2	70	55
Microsoft-Windows-Windows Defender%4WHC	87	0	92	62	43
Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	368	590	395	459	319
Microsoft-Windows-WindowsBackup%4ActionCenter	12	0	16	0	0
Microsoft-Windows-WindowsSystemAssessmentTool%4Operational	39	0	39	0	0
Microsoft-Windows-WindowsUpdateClient%4Operational	1018	173	1026	55	42
Microsoft-Windows-Winlogon%4Operational	0	902	0	0	0
Microsoft-WindowsAzure-Diagnostics%4Bootstrapper	0	0	0	0	0
Microsoft-WindowsAzure-Diagnostics%4GuestAgent	1957	1488	1924	1709	1767
Microsoft-WindowsAzure-Diagnostics%4Heartbeat	1616	1592	1614	1537	1616
Microsoft-WindowsAzure-Diagnostics%4Runtime	0	0	0	0	0
Microsoft-WindowsAzure-Status%4GuestAgent	124	1392	213	1302	0
Microsoft-WindowsAzure-Status%4Plugins	8	1258	10	14	10
Security	778	156589	993	29177	4384
Setup	0	29	0	107	0
System	1513	8154	1864	19808	3762
Windows PowerShell	57	585	93	1044	148

B.4. Clustermap based on the Log Distribution

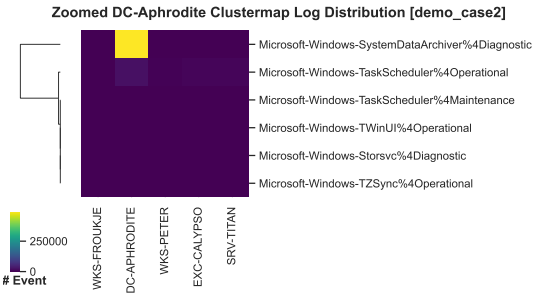
Figures B.5 through B.7 depict the clustermaps resulting from clustering analysis based on the log distributions of the hosts observed in Demo-Case 2 through 4, respectively. In each figure, subfigure 'a' represents the full map, while subfigure 'b' displays a zoomed-in clustermap focusing solely on Security, Application, and System logs



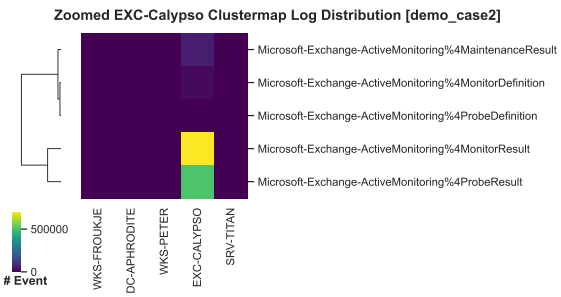
(a) Complete clustermap encompassing all log types



(b) Clustermap focused on application, system, and security logs

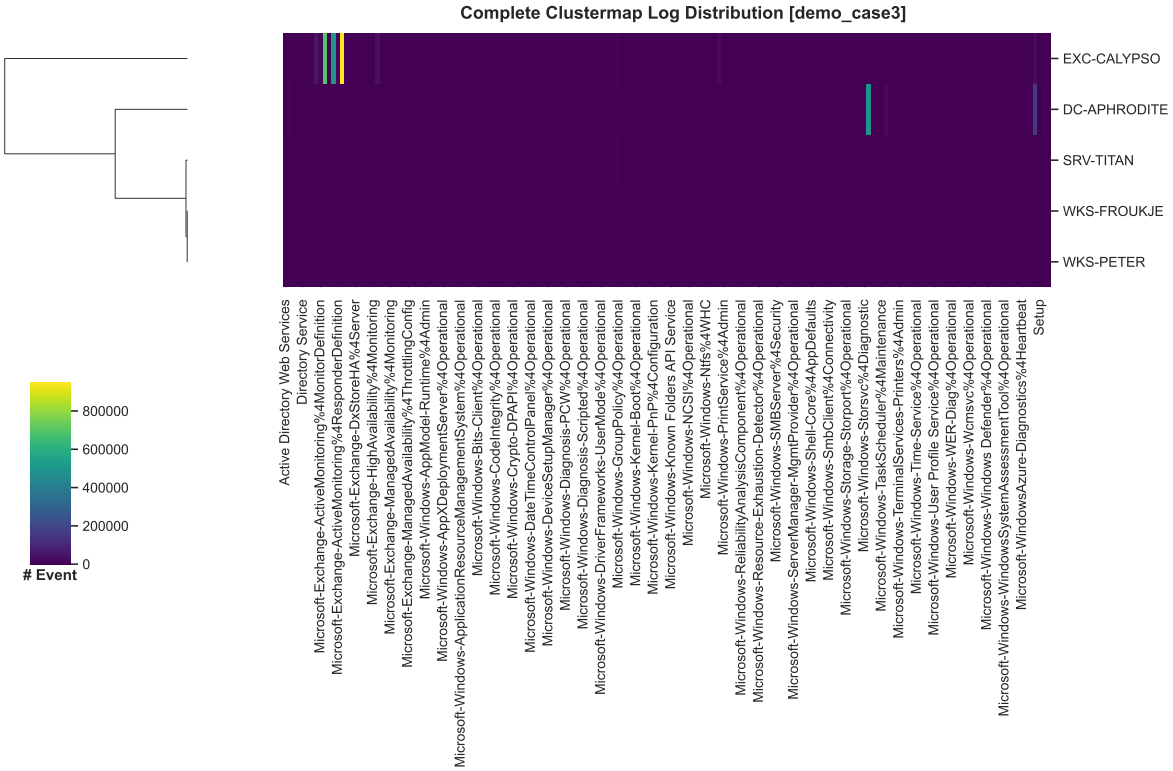


(c) Clustermap focused on outlier logs for DC-Aphrodite

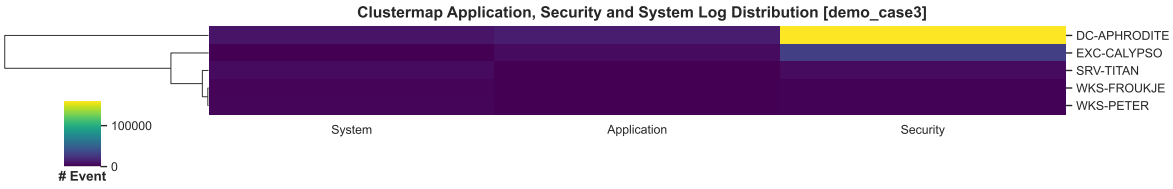


(d) Clustermap focused on outlier logs for EXC-Calypso

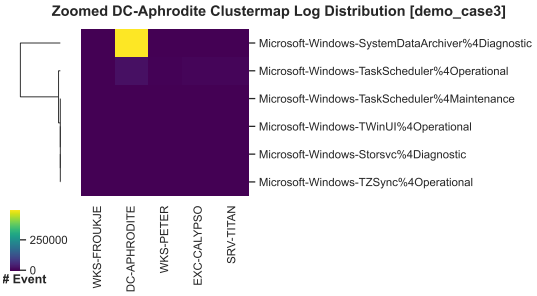
Figure B.5: Clustermap based on the log distribution for Demo-Case 2



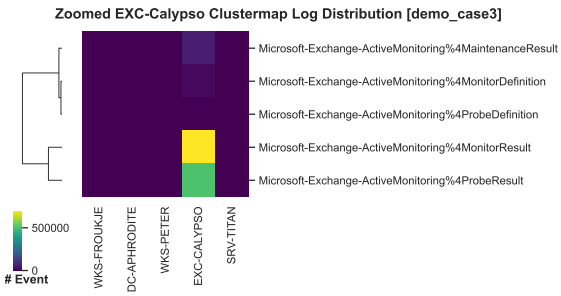
(a) Complete clustermap encompassing all log types



(b) Clustermap focused on application, system, and security logs

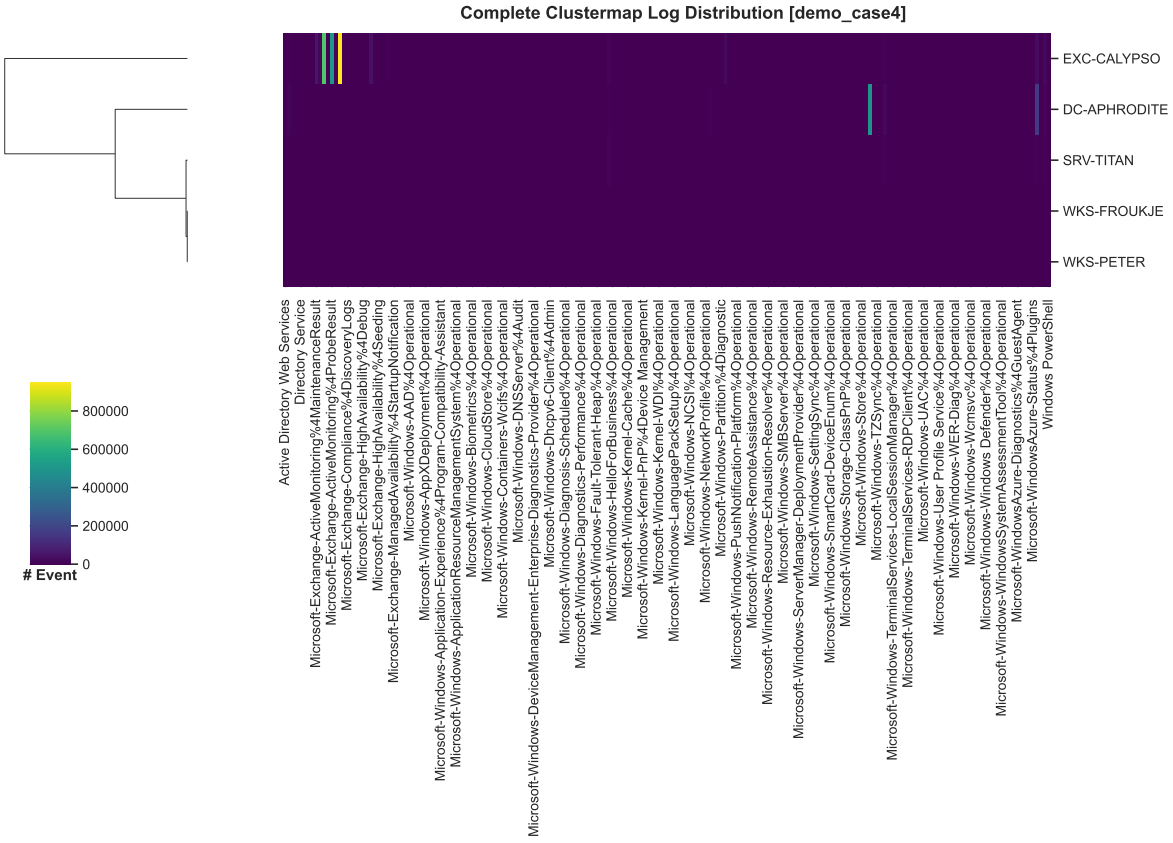


(c) Clustermap focused on outlier logs for DC-Aphrodite

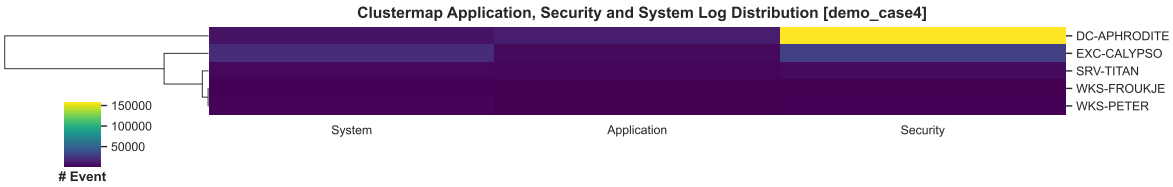


(d) Clustermap focused on outlier logs for EXC-Calypso

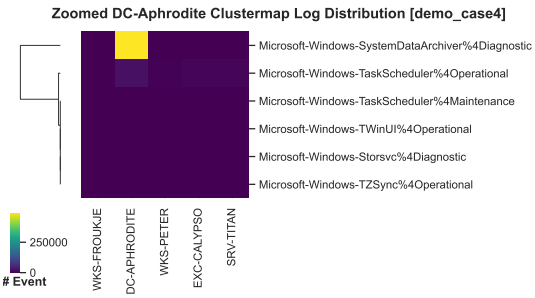
Figure B.6: Clustermap based on the log distribution for Demo-Case 3



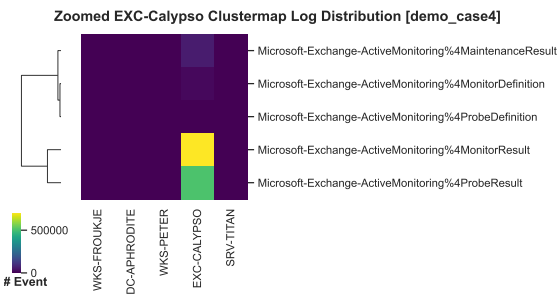
(a) Complete clustermap encompassing all log types



(b) Clustermap focused on application, system, and security logs



(c) Clustermap focused on outlier logs for DC-Aphrodite



(d) Clustermap focused on outlier logs for EXC-Calypso

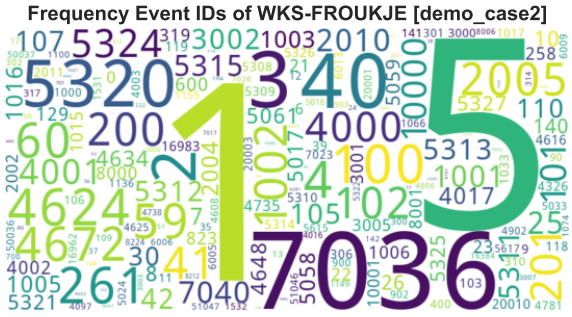
Figure B.7: Clustermap based on the log distribution for Demo-Case 4

C Event IDs Frequencies

Here, section C.1 contains the frequencies of the event IDs of the hosts in Demo-Case 2, 3 and 4. Then, section C.2 contains the clusterings based on these event IDs frequencies of the hosts in Demo-Case 2, 3 and 4.

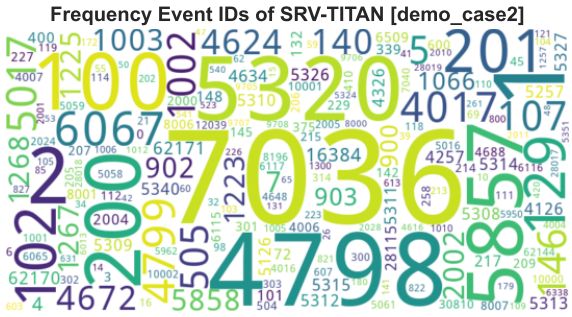
C.1. Event IDs Frequency of the Hosts

Figures B.1, B.2, and B.3 present the log distribution of the hosts observed in Demo-case 2, 3, and 4, respectively.

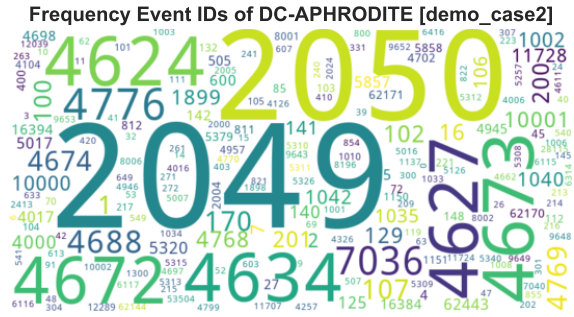


(a) Log Distribution WKS-Froukje

(b) Log Distribution WKS-Peter

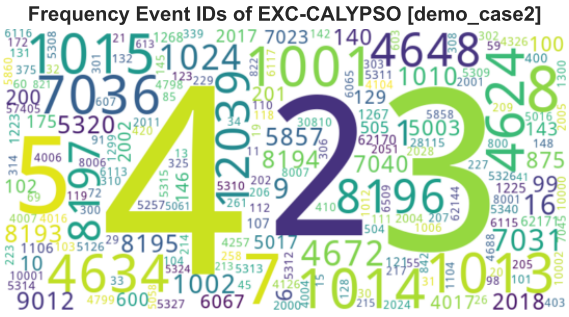


(c) Log Distribution SRV-Titan



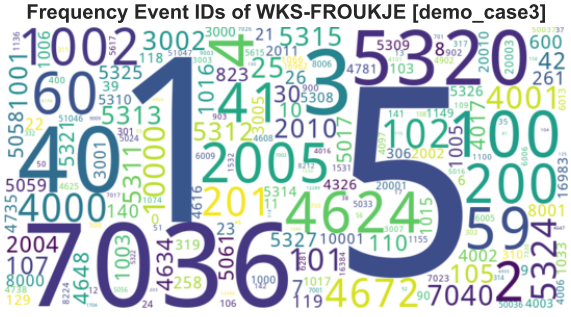
(d) Log Distribution DC-Aphrodite

Figure C.1: Event IDs Frequency of Demo-Case 2



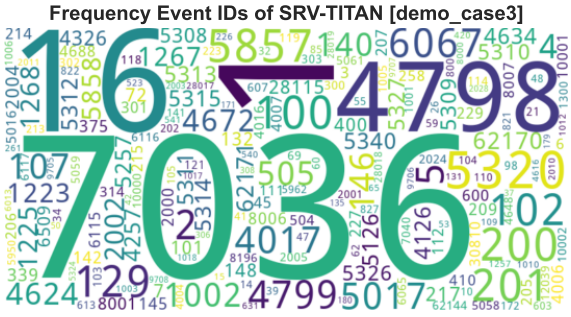
(e) Log Distribution EXC-Calypso

Figure C.1: Event IDs Frequency of Demo-Case 2 (cont.)

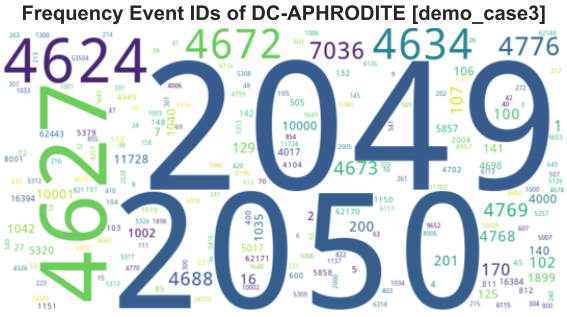


(a) Log Distribution WKS-Froukje

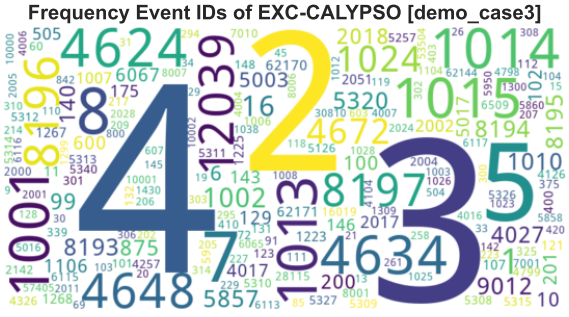
(b) Log Distribution WKS-Peter



(c) Log Distribution SRV-Titan

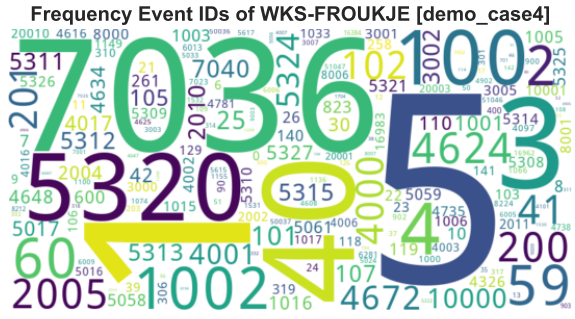


(d) Log Distribution DC-Aphrodite



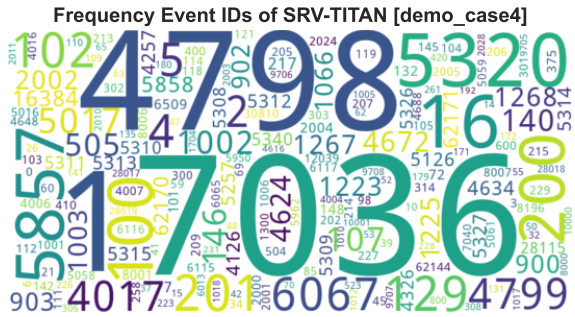
(e) Log Distribution EXC-Calypso

Figure C.2: Event IDs Frequency of Demo-Case 3

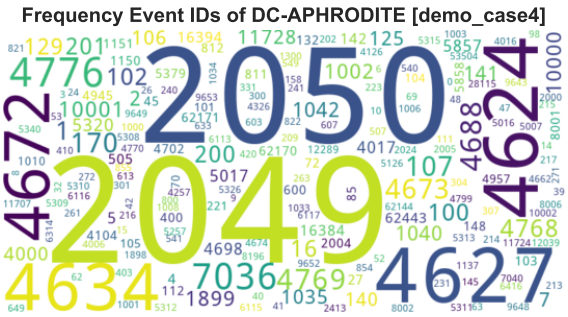


(a) Log Distribution WKS-Froukje

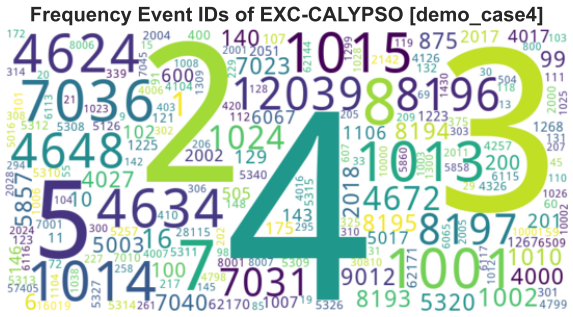
(b) Log Distribution WKS-Peter



(c) Log Distribution SRV-Titan



(d) Log Distribution DC-Aphrodite

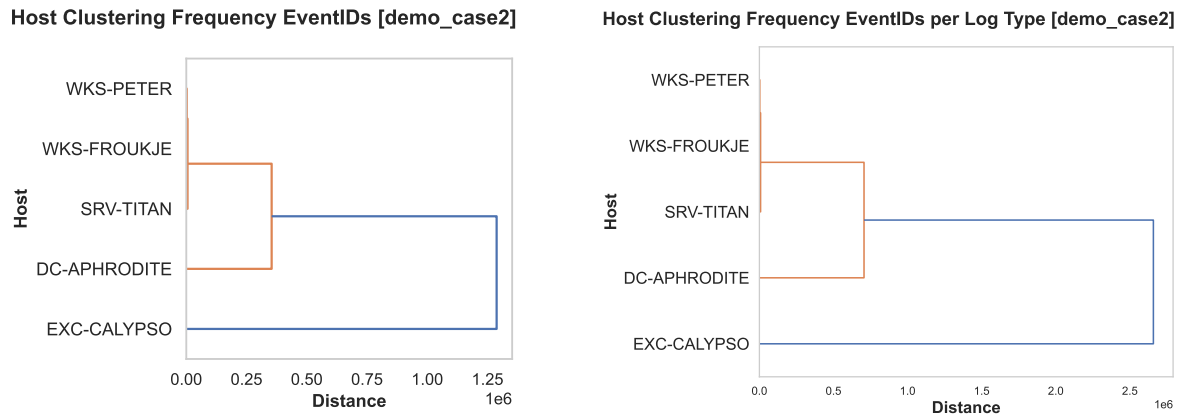


(e) Log Distribution EXC-Calypso

Figure C.3: Event IDs Frequency of Demo-Case 4

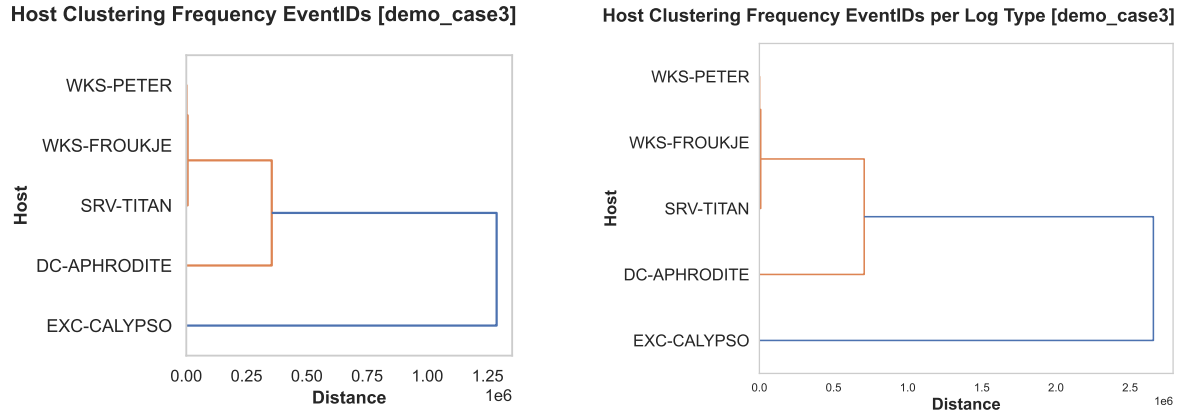
C.2. Event ID Frequency Clustering of the Hosts

Figures C.4, C.5, and C.6 illustrate the clustering based on the event ID frequecies across all logs and per log of the hosts observed in Demo-case 2, 3, and 4, respectively.



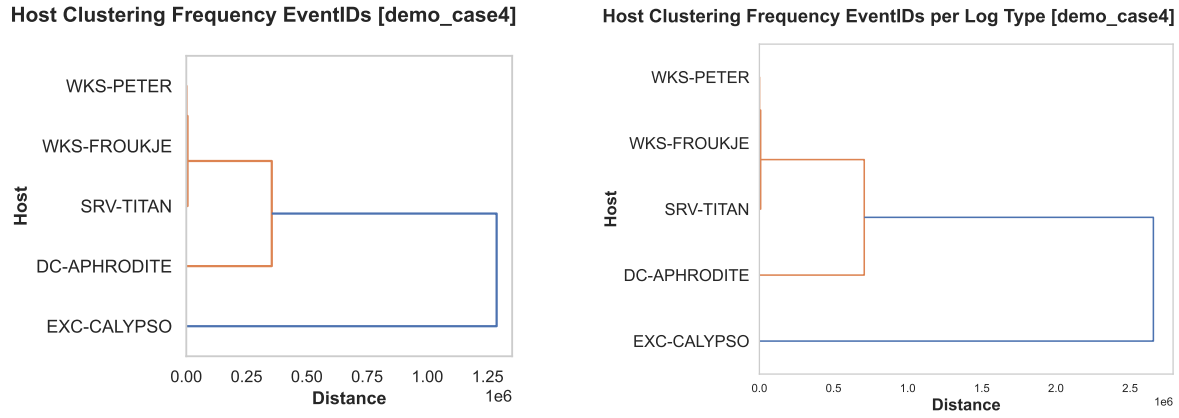
(a) All Log Types **(b)** Per Log Type

Figure C.4: Clustering based on frequency of the Event IDs in Demo-Case 2



(a) All Log Types **(b)** Per Log Type

Figure C.5: Clustering based on frequency of the Event IDs in Demo-Case 3



(a) All Log Types **(b)** Per Log Type

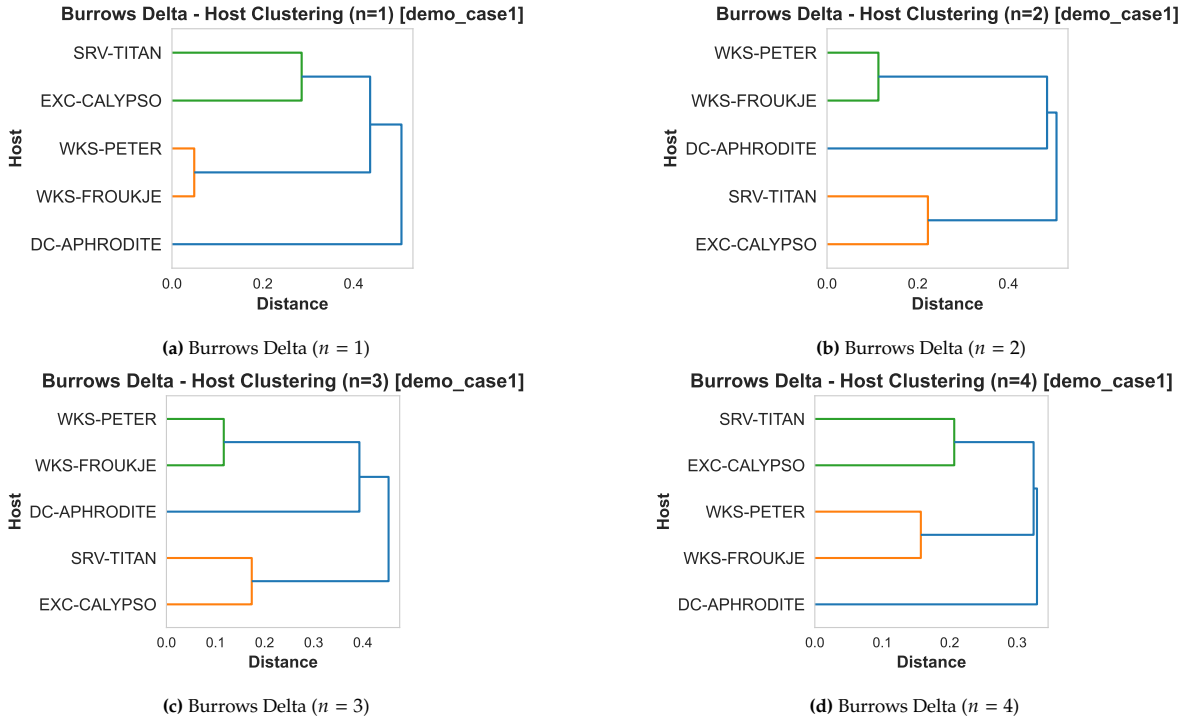
Figure C.6: Clustering based on frequency of the Event IDs in Demo-Case 4

D

Delta Method Clustering

D.1. Burrows Delta

The Burrows Delta (BD) method has been applied to the log messages to cluster the hosts. Figure D.1 presents the resulting clusters for uni- up to 5-grams. A notable trend is observed: the distances between clusters decrease as the n-gram size increases. Additionally, consistent patterns emerge across different n-gram sizes. Specifically, the exchange server and server consistently form a cluster, as do the workstations. However, the clustering behaviour of the domain controller exhibits variability. It alternates between clustering with the workstations (for bi- and tri-grams), grouping with the server cluster (for 5-grams), and appearing as an independent cluster (for uni- and 4-grams).



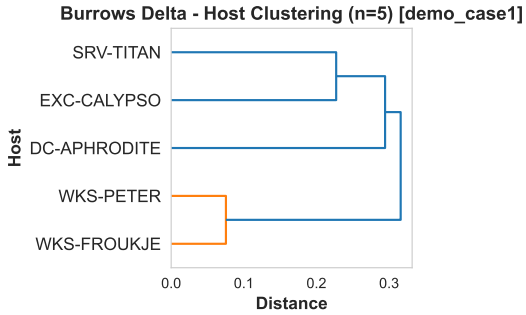
(a) Burrows Delta ($n = 1$)

(b) Burrows Delta ($n = 2$)

(c) Burrows Delta ($n = 3$)

(d) Burrows Delta ($n = 4$)

Figure D.1: Clustering of Demo-Case 1 based on Burrows Delta

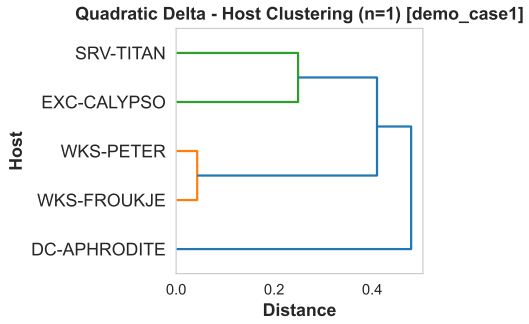


(e) Burrows Delta ($n = 5$)

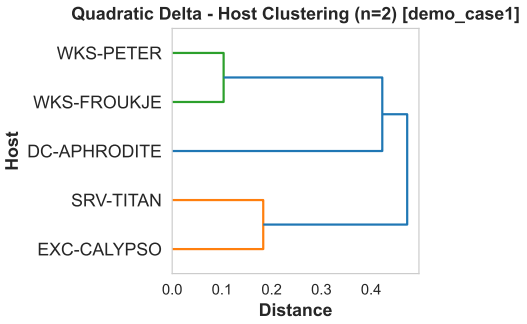
Figure D.1: Clustering of Demo-Case 1 based on Burrows Delta (cont.)

D.2. Quadratic Delta

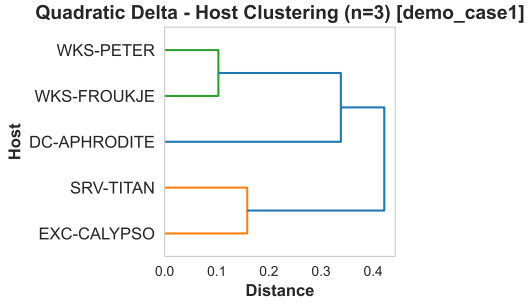
The Quadratic Delta (QD) method has been applied to the log messages to cluster the hosts. Figure D.2 presents the resulting clusters for uni- up to 5-grams. Like BD, a consistent pattern emerges across different n-gram sizes for the workstations and servers; only the domain controller exhibits variability. It alternates between forming an independent cluster (for uni-, 4-, and 5-grams) and grouping with the workstations (for bi- and tri-grams). Notably, the clusters for the unigrams and bigrams of BD and QD are identical, including their distances. However, a difference arises with 5-grams: while BD clusters the domain controller with the servers, QD treats the domain controller as an independent group.



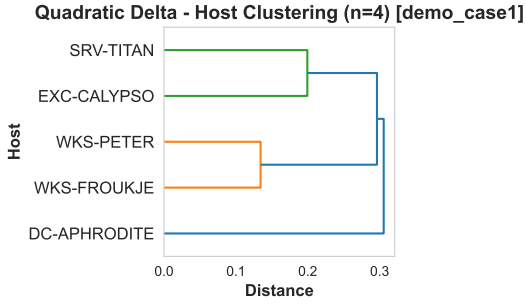
(a) Quadratic Delta ($n = 1$)



(b) Quadratic Delta ($n = 2$)

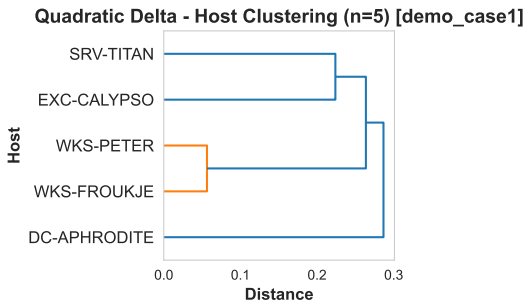


(c) Quadratic Delta ($n = 3$)



(d) Quadratic Delta ($n = 4$)

Figure D.2: Clustering of Demo-Case 1 based on Quadratic Delta



(e) Quadratic Delta ($n = 5$)

Figure D.2: Clustering of Demo-Case 1 based on Quadratic Delta (cont.)

D.3. Linear Delta

The Linear Delta (LD) method standardises using the parameters of the Laplace distribution. Defining what to consider as the document collection is essential to determine these parameters accurately. In this context, four distinct definitions of the document collection have been examined:

1. **The two logs under consideration**: This definition involves using only the two specific logs currently being analysed. This approach ensures that the parameters are directly relevant to the logs but may lack broader contextual data.
2. **All the logs of the two hosts under consideration**: The collection includes all logs generated by the two hosts being examined. This definition provides a broader dataset that reflects the behaviour and characteristics of these hosts over time, potentially leading to more robust parameter estimation.
3. **All the logs of the log type under consideration**: This definition expands the collection to include all logs of the same type as the ones under consideration. By focusing on the log type, this approach aims to capture the general characteristics and patterns associated with that particular log type, regardless of the specific hosts.
4. **All logs**: The broadest definition encompasses all available logs in the dataset. This comprehensive approach leverages the maximum amount of data, potentially enhancing the robustness of the parameter estimation but at the risk of introducing noise from unrelated logs.

For illustrative purposes, Figure D.3 displays the definition of the document collection when computing the linear delta for log type 2 between the workstations WKS-Froukje and WKS-Peter. Each definition provides a distinct scope for determining the Laplace parameters, balancing specificity and generalisation to various extents. The choice of definition can significantly influence the parameter estimation and, consequently, the clustering outcomes. This section delves into the impact of the choice of definition. By exploring these influences, a comprehensive understanding is provided of how the definition of document collection impacts the overall clustering.

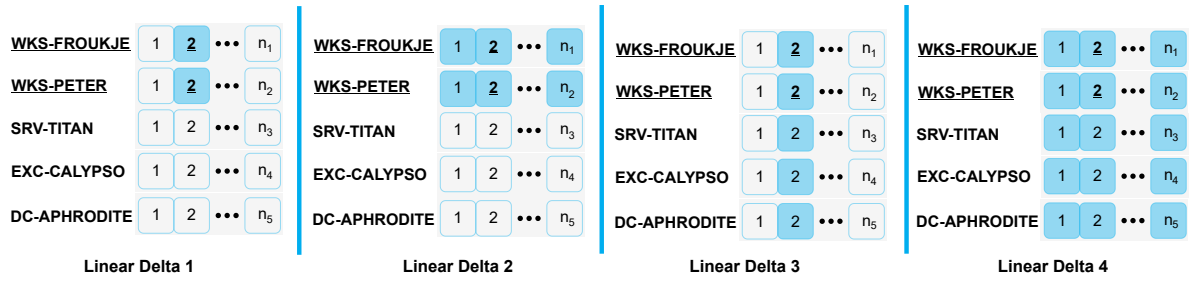
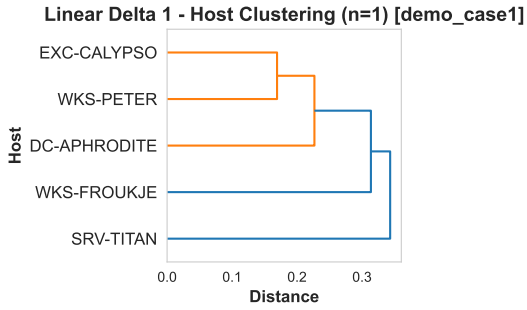


Figure D.3: Definition of the document collection for linear deltas between the workstations WKS-Froukje and WKS-Peter of log type 2

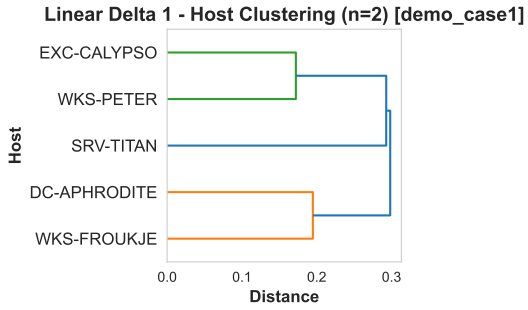
D.3.1. Linear Delta 1

The Linear Delta 1 (LD1) method has been applied to the log messages to cluster the hosts. Here, the document collection is defined as the two logs under consideration. Figure D.4 presents the resulting clusters for uni- up to 5-grams. These clusterings differ significantly from those obtained using BD and QD. Specifically, LD1 consistently groups EXC-Calypso and WKS-Peter together, while BD and QD consistently group the two workstations together and the two servers together.

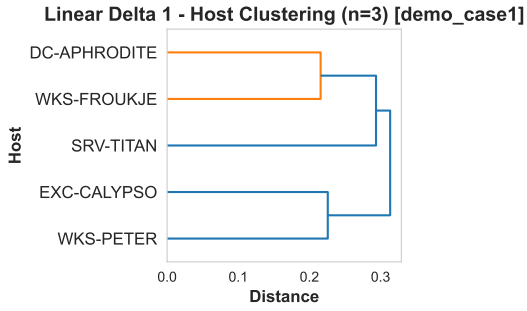
Additionally, unlike BD and QD, where only the domain controller exhibited variance, LD1 also shows significant variation in the clustering of WKS-Froukje and SRV-Titan. Depending on the n-gram, these hosts are grouped differently: for unigrams, each of these hosts is treated as an independent group; for bigrams and trigrams, WKS-Froukje and DC-Aphrodite cluster together, followed by SRV-Titan joining this group; and for 4-grams and 5-grams, WKS-Froukje and SRV-Titan cluster together, while DC-Aphrodite joins the group of WKS-Peter and EXC-Calypso.



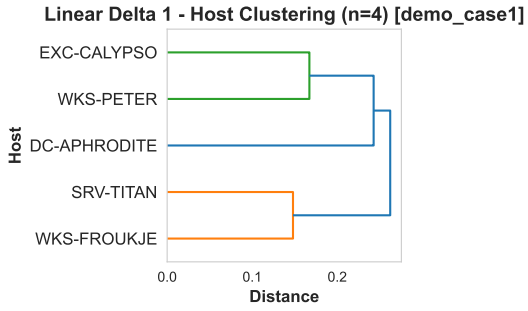
(a) Linear Delta 1 ($n = 1$)



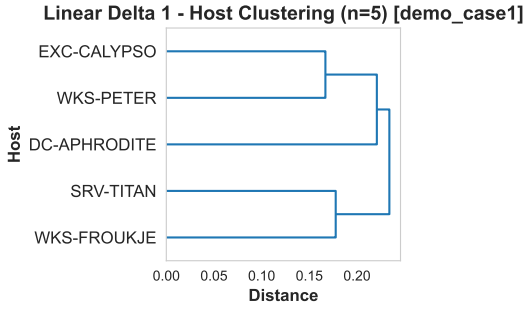
(b) Linear Delta 1 ($n = 2$)



(c) Linear Delta 1 ($n = 3$)



(d) Linear Delta 1 ($n = 4$)



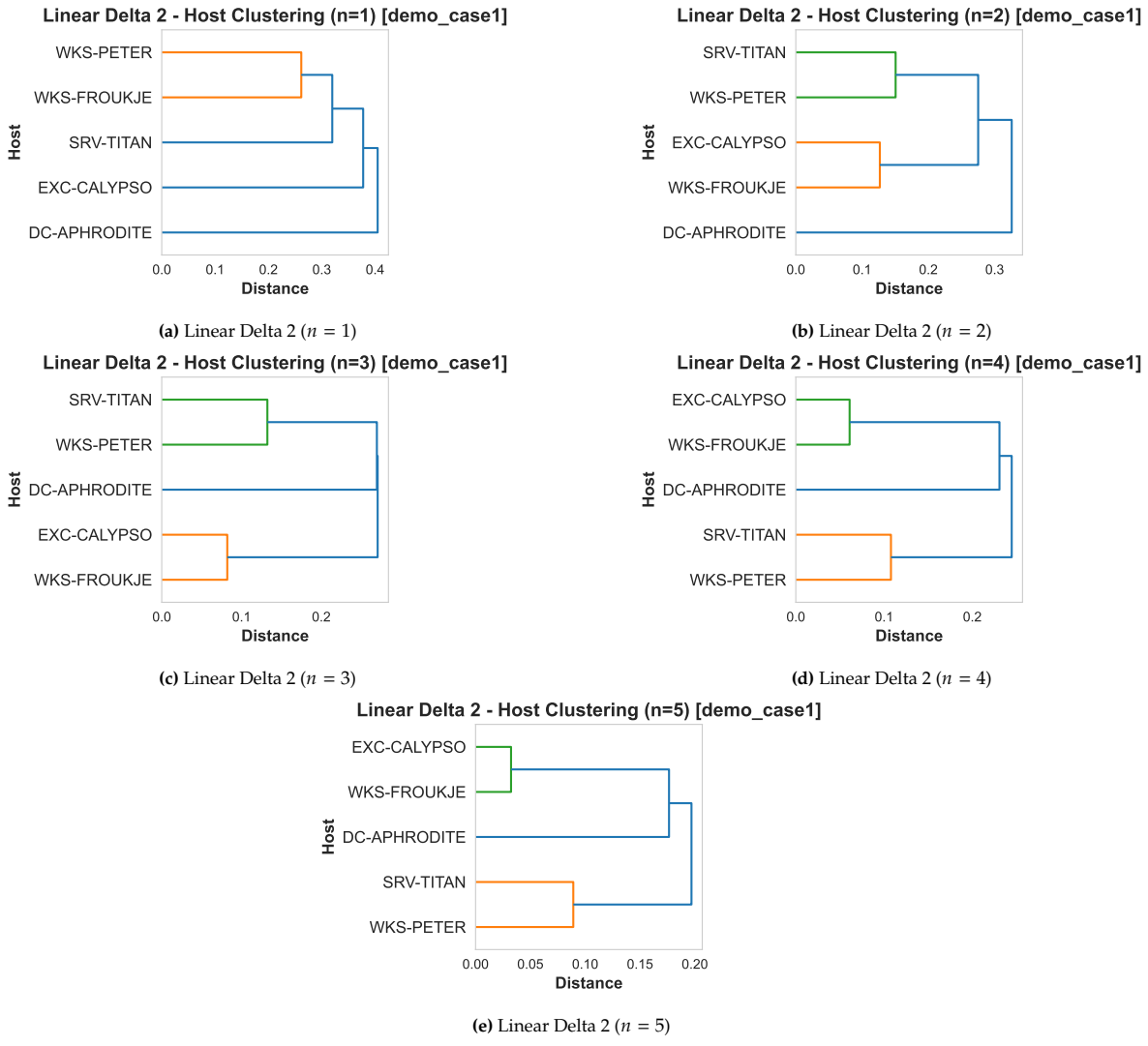
(e) Linear Delta 1 ($n = 5$)

Figure D.4: Clustering of Demo-Case 1 based on Linear Delta 1

D.3.2. Linear Delta 2

The Linear Delta 2 (LD2) method has been applied to the log messages to cluster the hosts. Here, the document collection is defined as all the logs of the two hosts under consideration. Figure D.5 presents the resulting clusters for uni-, up to 5-grams. These clusterings differ significantly not only from those obtained using BD and QD but also from those derived using LD1. Notably, there is no overlap in the clustering outcomes across these methods.

While LD2 does not exhibit any consistent groupings overall, EXC-Calypso and WKS-Froukje are consistently grouped for n-grams greater than one, as are SRV-Titan and WKS-Peter. These clusters vary based on the inclusion of DC-Aphrodite, which clusters as an independent group with bigrams, joins the cluster containing the server with trigrams, or joins the cluster containing the exchange server with 4-grams and 5-grams. For unigrams, the workstations are clustered together, with the servers and domain controller following as independent groups.



(a) Linear Delta 2 ($n = 1$)

(b) Linear Delta 2 ($n = 2$)

(c) Linear Delta 2 ($n = 3$)

(d) Linear Delta 2 ($n = 4$)

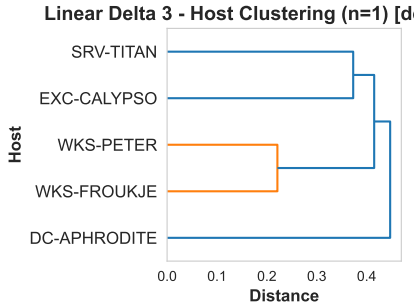
(e) Linear Delta 2 ($n = 5$)

Figure D.5: Clustering of Demo-Case 1 based on Linear Delta 2

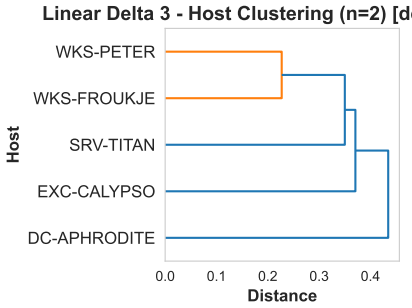
D.3.3. Linear Delta 3

The Linear Delta 3 (LD3) method has been applied to the log messages to cluster the hosts. Here, the document collection is defined as all the logs of the log type under consideration. Figure D.6 displays the resulting clusters for uni- up to 5-grams. The clustering with unigrams is consistent with the results obtained using BD and QD unigrams. Likewise, as with BD and QD, the workstations consistently group together initially. However, regardless of the n-gram size, the inter-distance between the workstations is significantly greater under LD3 - approximately 0.2 - compared to BD and QD, which range from 0.05 to 0.1. Additionally, unlike BD and QD, the servers do not consistently form a cluster.

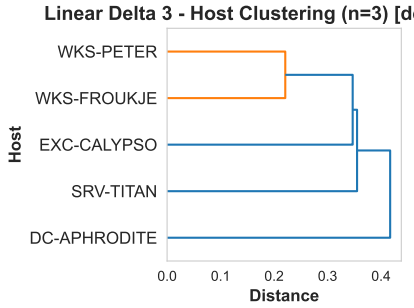
Instead, the domain controller and servers emerge as independent clusters, with slight variations in the order of clustering. Specifically, the joining order of servers in bigram clustering mirrors that of LD2 unigram clustering. The alternative order observed in tri-, 4-, and 5-gram clustering does not occur in BD, QD, LD1, or LD2.



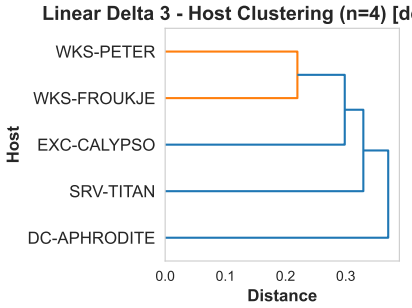
(a) Linear Delta 3 ($n = 1$)



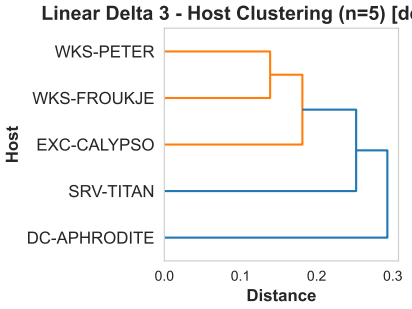
(b) Linear Delta 3 ($n = 2$)



(c) Linear Delta 3 ($n = 3$)



(d) Linear Delta 3 ($n = 4$)

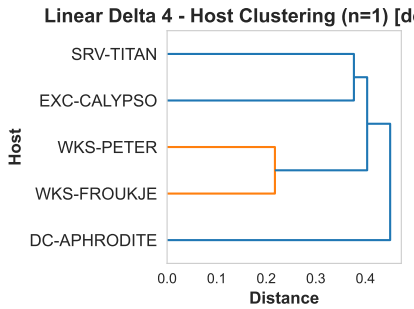


(e) Linear Delta 3 ($n = 5$)

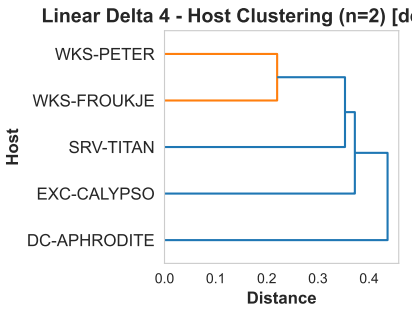
Figure D.6: Clustering of Demo-Case 1 based on Linear Delta 3

D.3.4. Linear Delta 4

The Linear Delta 4 (LD4) method has been applied to the log messages to cluster the hosts. Here, the document collection is defined as all the available logs. Figure D.7 illustrates the resulting clusters for uni- up to 5-grams. Regardless of the n-gram size, the clusters obtained with LD4 are identical, concerning grouping and inter-distances, to those obtained using LD3.

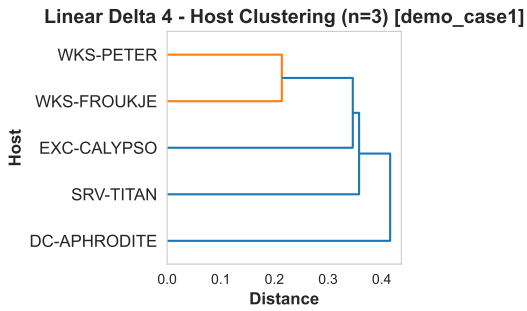


(a) Linear Delta 4 ($n = 1$)

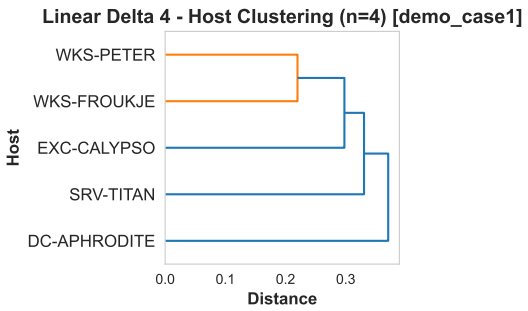


(b) Linear Delta 4 ($n = 2$)

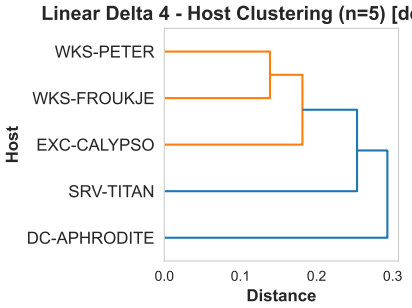
Figure D.7: Clustering of Demo-Case 1 based on Linear Delta 4



(c) Linear Delta 4 ($n = 3$)



(d) Linear Delta 4 ($n = 4$)



(e) Linear Delta 4 ($n = 5$)

Figure D.7: Clustering of Demo-Case 1 based on Linear Delta 4 (cont.)

D.3.5. Impact Definition Document Collection

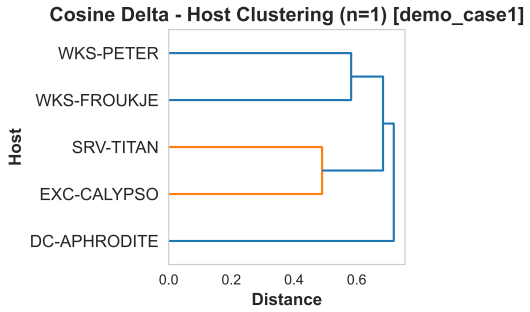
Limiting the document collection to logs of a specific log type - LD3 -while including all logs from all hosts - LD4 - leads to the same clustering results despite the broader scope. This similitude suggests that using the full range of logs - LD4 - is redundant, and the document collection can be effectively confined to logs of the specific log type under consideration - LD3.

However, when the document collection is restrained to solely the two logs being compared - LD1 - or all logs from only the hosts being examined - LD2 -, the resulting clusters differ significantly from each other and those produced by LD3 and LD4. Specifically, LD1 and LD2 produce distinct clustering patterns, indicating that the restriction to the logs of only the hosts under consideration substantially impacts the clustering outcomes. Only LD2 under unigrams shows some resemblance to LD3 and LD4 under bigrams.

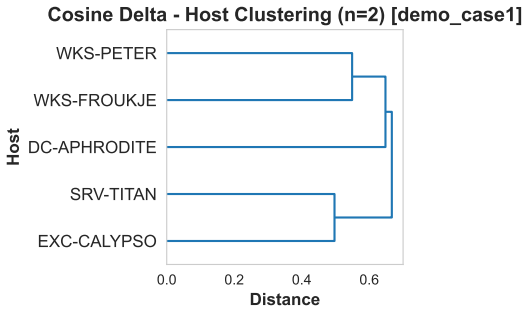
Hence, this divergence in clustering outcomes suggests that the clustering behaviour changes noticeably when the document collection is restrained to specific hosts. The specificity of the logs considered in LD1 and LD2 introduces variations in the clusters that are not observed when a broader log collection is used in LD3 and LD4. Therefore, while LD3 and LD4 demonstrate that a specific log type's logs are sufficient for consistent clustering, reducing the scope to individual hosts or specific logs significantly alters the results. This highlights the importance of the definition of the document collection to the clustering.

D.4. Cosine Delta

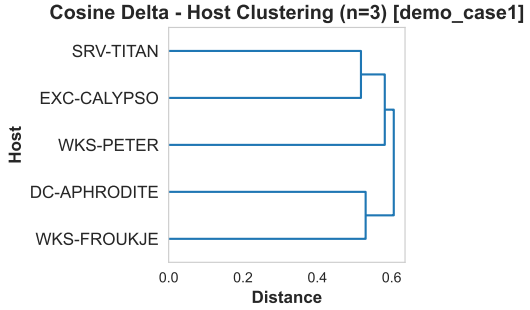
The Cosine Delta (CD) method has been applied to the log messages to cluster the hosts. Figure D.8 illustrates the resulting clusters for uni- up to 5-grams. Like the BD and QD methods, the workstations and servers form initial clusters, except when using CD under trigrams. Specifically, CD produces clusters identical to BD and QD with unigrams, bigrams and 4-grams, respectively. However, CD generates more pronounced inter-distances between the workstations and between the servers. Moreover, unlike BD and QD, the inter-distance between workstations is larger than between servers. Notably, the clustering obtained under trigrams shows that while the servers are clustered, WKS-Froukje clusters with DC-Aphrodite upon which WKS-Peter joins them.



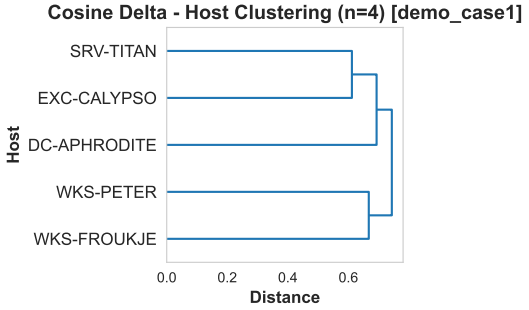
(a) Cosine Delta ($n = 1$)



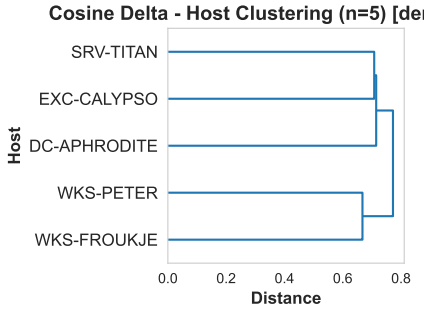
(b) Cosine Delta ($n = 2$)



(c) Cosine Delta ($n = 3$)



(d) Cosine Delta ($n = 4$)

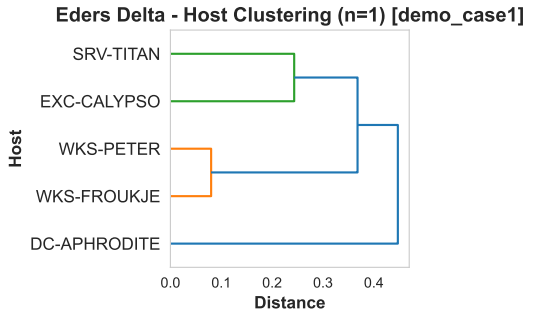


(e) Cosine Delta ($n = 5$)

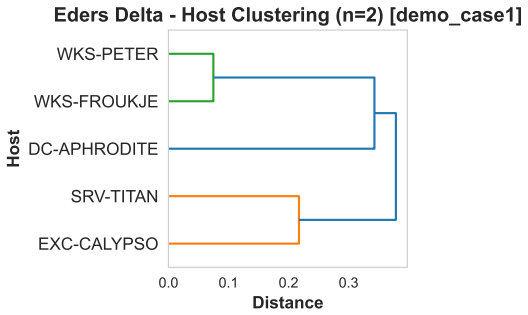
Figure D.8: Clustering of Demo-Case 1 based on Cosine Delta

D.5. Eders Delta

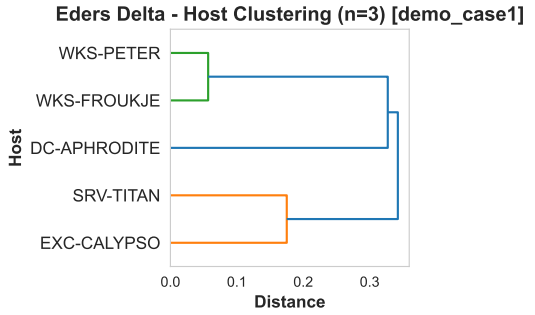
The Eders Delta (ED) method has been applied to the log messages to cluster the hosts. Figure D.9 illustrates the resulting clusters for uni- up to 5-grams. Similar to the BD and QD methods, the workstations and servers form initial clusters, with only the domain controller showing variation. Up to trigrams, ED produces clusters identical to BD and QD, with slight variations in inter-distances. The clusters generated by ED with 4-grams and 5-grams are swapped compared to those obtained by BD, again with minor differences in inter-distances. However, across varying n-gram sizes, ED results in relatively smaller inter-distances among workstations than among servers.



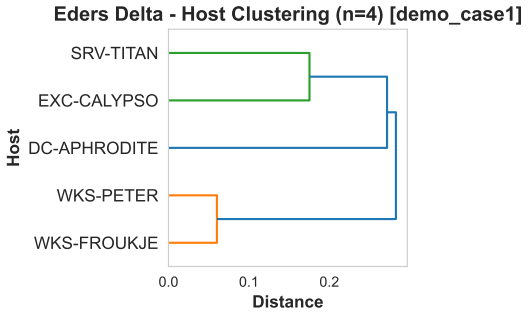
(a) Eders Delta ($n = 1$)



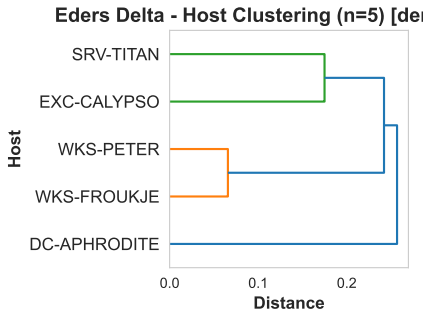
(b) Eders Delta ($n = 2$)



(c) Eders Delta ($n = 3$)



(d) Eders Delta ($n = 4$)

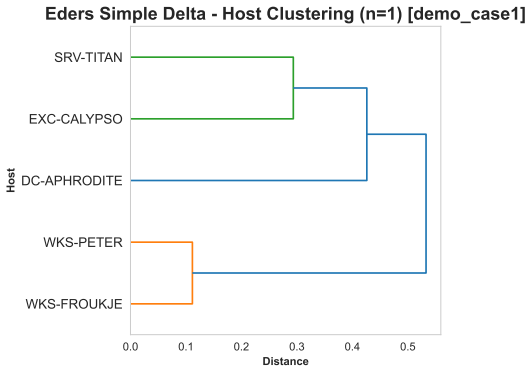


(e) Eders Delta ($n = 5$)

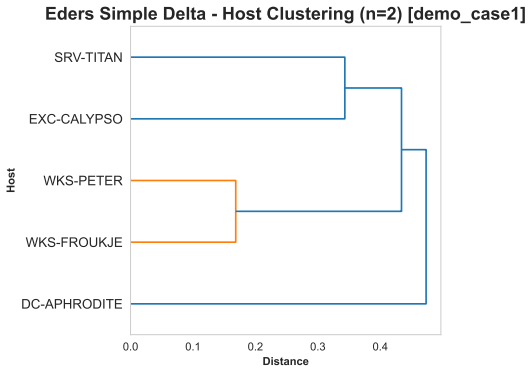
Figure D.9: Clustering of Demo-Case 1 based on Eders Delta

D.6. Eders Simple Delta

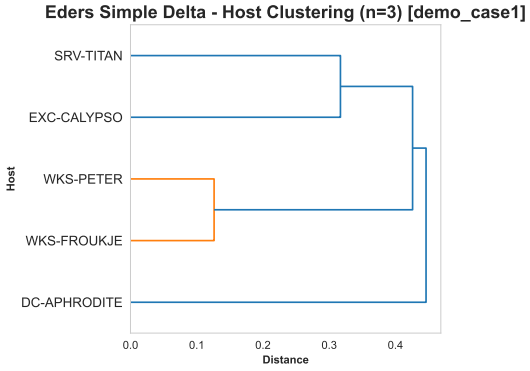
The Eders Simple Delta (ESD) method has been applied to the log messages to cluster the hosts. Figure D.9 illustrates the resulting clusters for uni- up to 5-grams. Similar to the BD, QD, and ED methods under specific gram sizes, the workstations and servers form initial clusters, with the domain controller showing variation in clustering either with the servers (for unigrams, 4-grams, and 5-grams) or at the end (for bigrams and trigrams). These clusters are identical to those obtained using the BD, CD, and ED methods, though they differ based on the specific n-gram size. Additionally, ESD shows greater distances between the hosts within these overlapping clusters than BD and ED but smaller distances than CD.



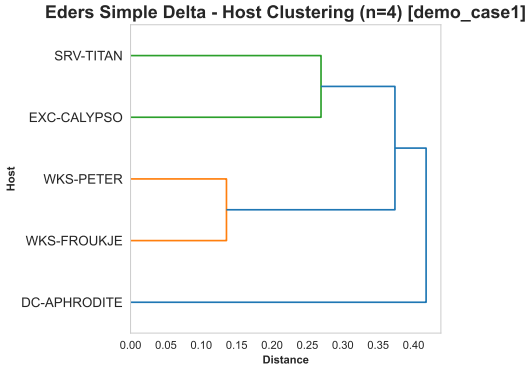
(a) Eders Simple Delta ($n = 1$)



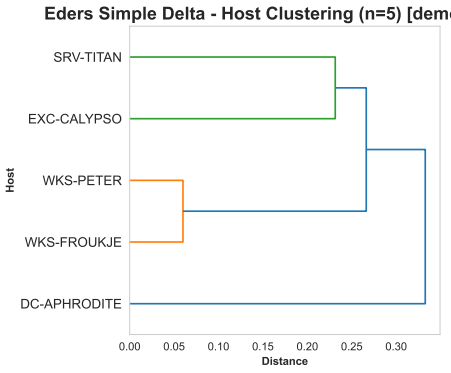
(b) Eders Simple Delta ($n = 2$)



(c) Eders Simple Delta ($n = 3$)



(d) Eders Simple Delta ($n = 4$)



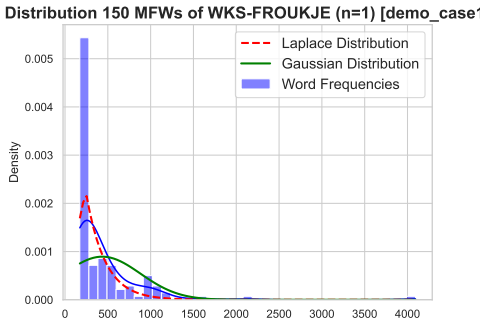
(e) Eders Simple Delta ($n = 5$)

Figure D.10: Clustering of Demo-Case 1 based on Eders Simple Delta

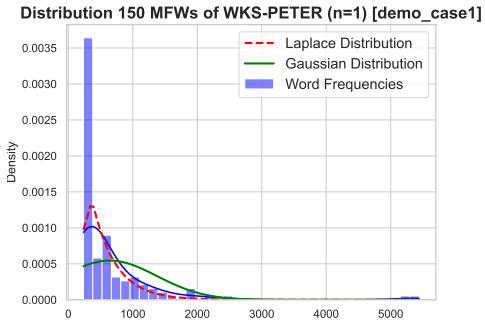
E

N-Gram Distribution Demo-Case 1

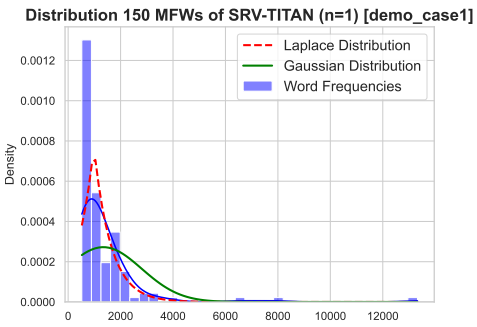
E.1. Distribution Unigrams ($n = 1$)



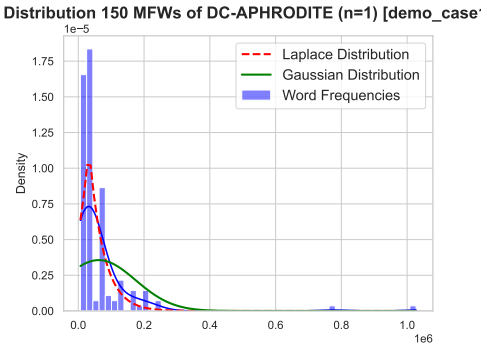
(a) Distribution WKS-Froukje



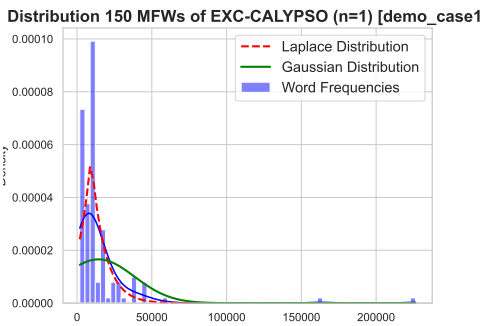
(b) Distribution WKS-Peter



(c) Distribution SRV-Titan



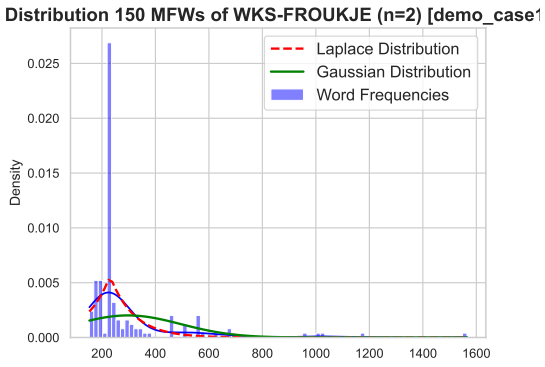
(d) Distribution DC-Aphrodite



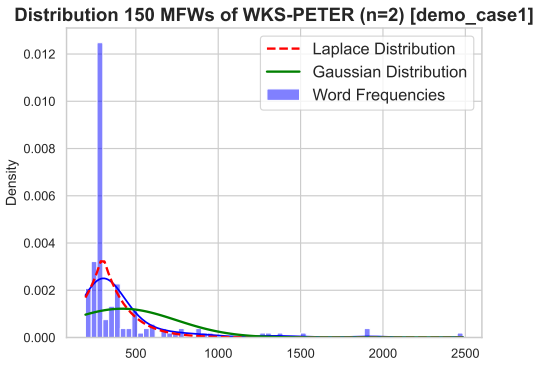
(e) Distribution EXC-Calypso

Figure E.1: Unigrams ($n = 1$) Distribution of the hosts in Demo-Case 1

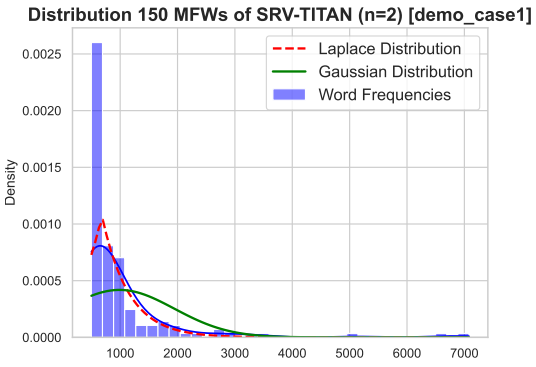
E.2. Distribution Bigrams ($n = 2$)



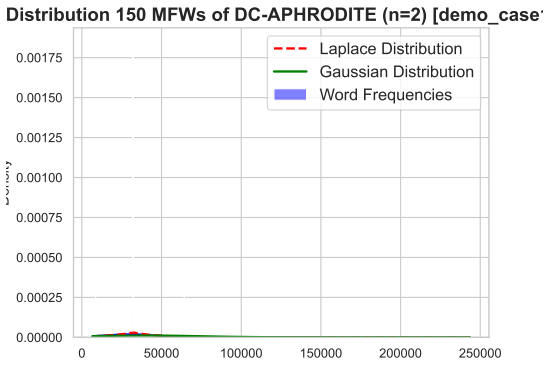
(a) Distribution WKS-Froukje



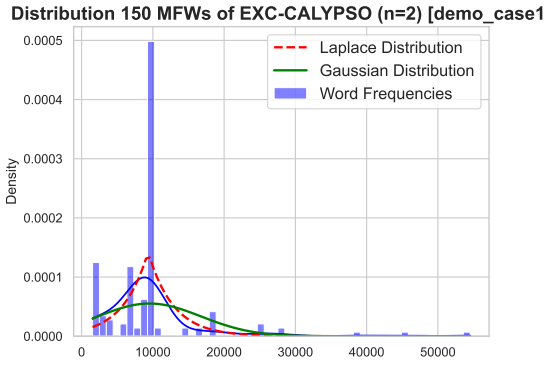
(b) Distribution WKS-Peter



(c) Distribution SRV-Titan



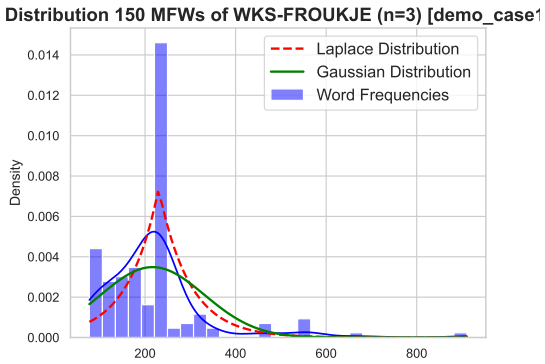
(d) Distribution DC-Aphrodite



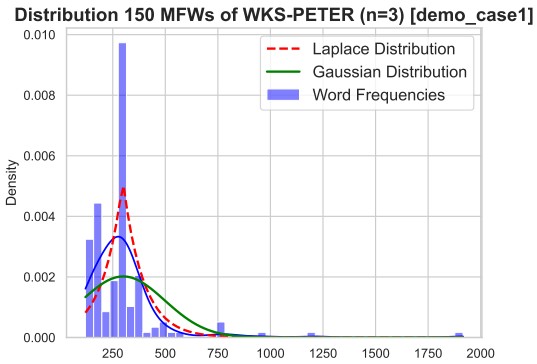
(e) Distribution EXC-Calypso

Figure E.2: Bigrams ($n = 2$) Distribution of the hosts in Demo-Case 1

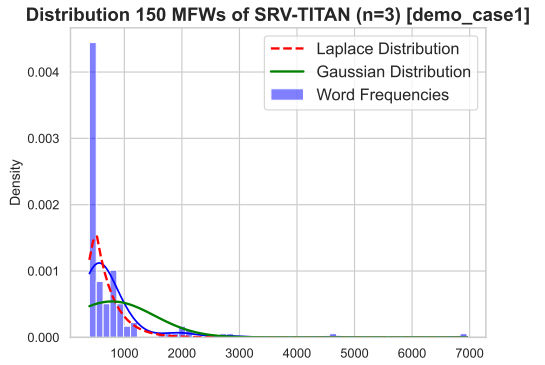
E.3. Distribution Trigrams ($n = 3$)



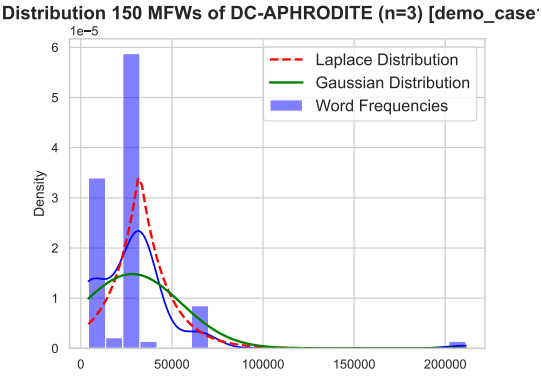
(a) Distribution WKS-Froukje



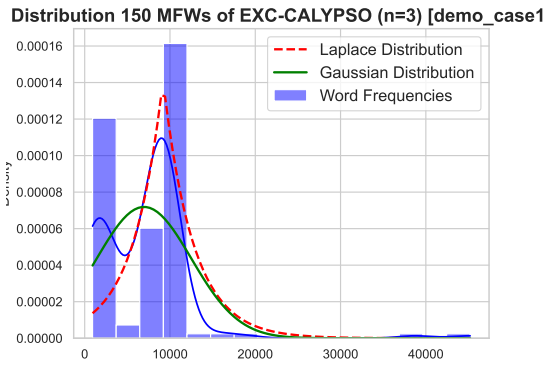
(b) Distribution WKS-Peter



(c) Distribution SRV-Titan



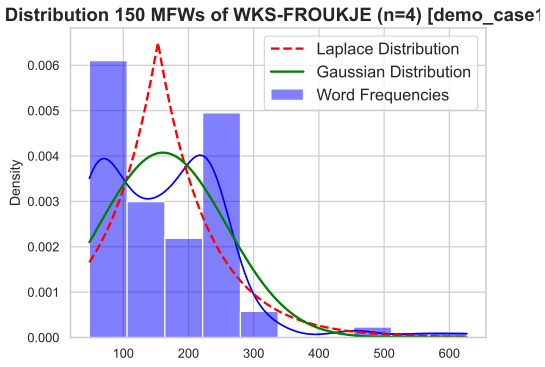
(d) Distribution DC-Aphrodite



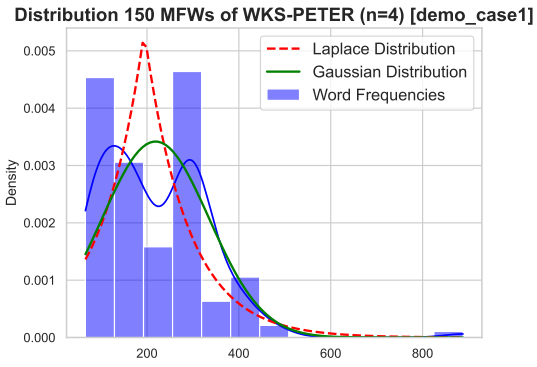
(e) Distribution EXC-Calypso

Figure E.3: Trigrams ($n = 3$) Distribution of the hosts in Demo-Case 1

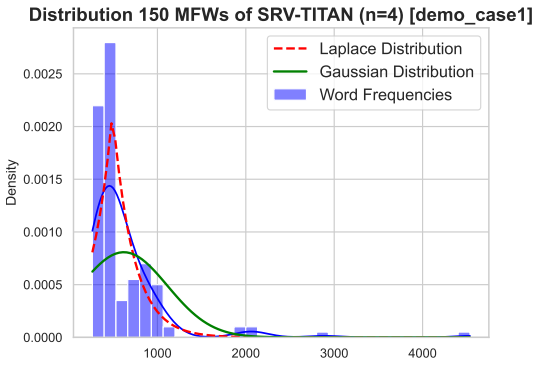
E.4. Distribution 4-grams ($n = 4$)



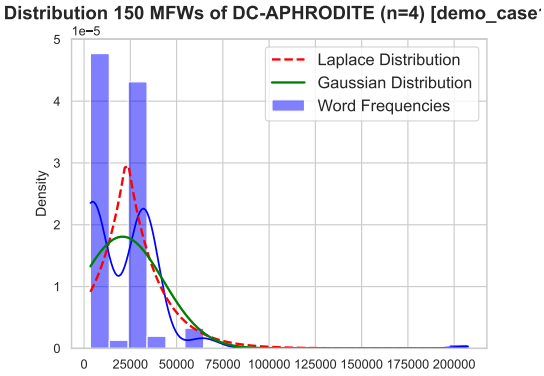
(a) Distribution WKS-Froukje



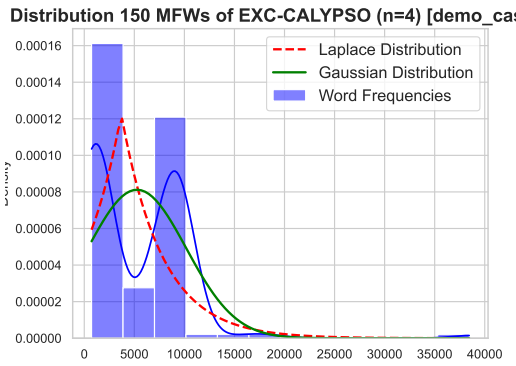
(b) Distribution WKS-Peter



(c) Distribution SRV-Titan



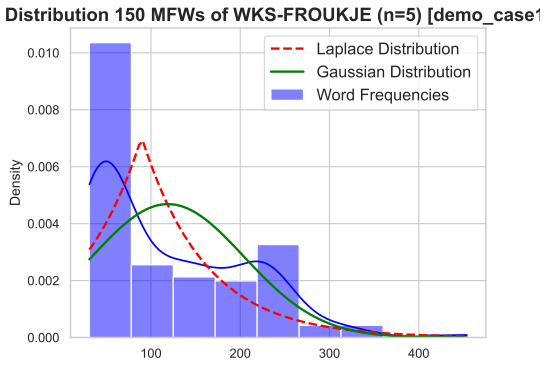
(d) Distribution DC-Aphrodite



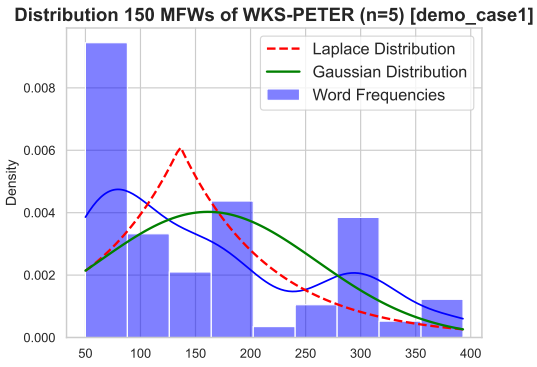
(e) Distribution EXC-Calypso

Figure E.4: 4-grams ($n = 4$) Distribution of the hosts in Demo-Case 1

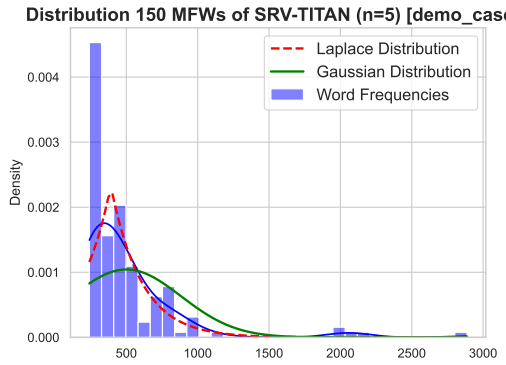
E.5. Distribution 5-grams ($n = 5$)



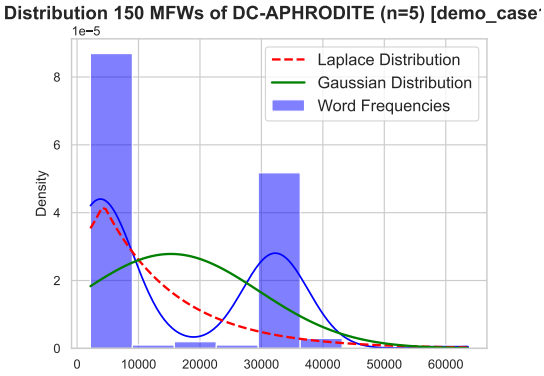
(a) Distribution WKS-Froukje



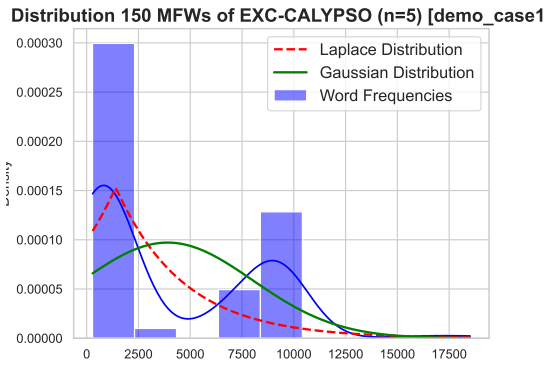
(b) Distribution WKS-Peter



(c) Distribution SRV-Titan



(d) Distribution DC-Aphrodite



(e) Distribution EXC-Calypso

Figure E.5: 5-grams ($n = 5$) Distribution of the hosts in Demo-Case 1

F

Contributing Logs

Here, for each behaviour type, namely common (), distinctive() and unique, an comparative analysis is conducted for the log types. Then, for each behaviour type comparative analysis the specific log types that are uniquely identified for each method are examined. This helps to understand the reliability and sensitivity of each method in detecting subtle differences. Thus, it will shed light on the effectiveness of capturing behaviour, thereby aiding in the selection of the most appropriate method for further investigations.

F.1. Common Behaviour

F.1.1. Overview Log Comparison

Here, the log types are presented between WKS-Froukje and WKS-Peter that are uniquely identified by the the delta method as completely similar. The methods compared are listed along the columns and rows of the table. Here, each cell in the table indicates the log types that have a zero score in the method corresponding to the row but do not have a zero score in the method corresponding to the column. In other words, if considering the cell located in Table F.1 at the intersection of column BD and row QD, the log type `Microsoft-Windows-TaskScheduler%4Operational` in this cell represents the log type that exhibits a zero QD score but not a zero BD score. This is done for the common behaviour for $n = 1$, $n = 4$ and $n = 5$ in Table F.1, F.2, and F.3, respectively.

Table F.1: Unique complete similar log types between the common behaviour of WKS-Froukje and WKS-Peter under unigrams ($n = 1$)

n=1	BD	QD	ED	ESD	LD3
BD				Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-Bits-Client%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Application Microsoft-Windows-Diagnosis-Scripted%4Admin System Microsoft-Windows-Bits-Client%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational
QD	Microsoft-Windows-TaskScheduler%4Operational		Microsoft-Windows-TaskScheduler%4Operational	Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-Bits-Client%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Application Microsoft-Windows-Diagnosis-Scripted%4Admin System Microsoft-Windows-Bits-Client%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational
ED	Microsoft-Windows-Application-Experience%4Program-Telemetry Windows PowerShell	Microsoft-Windows-Application-Experience%4Program-Telemetry Windows PowerShell		Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry 'Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-Bits-Client%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-LanguagePackSetup%4Operational Security Application Microsoft-Windows-Diagnosis-Scripted%4Admin System Microsoft-Windows-Bits-Client%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational
ESD	Windows PowerShell	Windows PowerShell			Application System
LD3	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-GroupPolicy%4Operational Windows PowerShell	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-GroupPolicy%4Operational Windows PowerShell	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-GroupPolicy%4Operational	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Windows Defender%4WHC Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	

Table F.2: Unique complete similar log types between the common behaviour of WKS-Froukje and WKS-Peter under 4-grams ($n = 4$)

n=4	BD	QD	ED	ESD	LD3
BD				Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Microsoft-Windows-NetworkProfile%4Operational
QD	Microsoft-Windows-Diagnosis-PLA%4Operational System		Microsoft-Windows-Diagnosis-PLA%4Operational	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational 'Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational 'Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Microsoft-Windows-NetworkProfile%4Operational System
ED	Microsoft-Windows-PrintService%4Admin Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational System	Microsoft-Windows-PrintService%4Admin Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational		Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-PrintService%4Admin Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Resource-Exhaustion-Detector%4Operational Microsoft-Windows-PrintService%4Admin Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-LanguagePackSetup%4Operational Microsoft-Windows-NetworkProfile%4Operational Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational System
ESD	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Kernel-EventTracing%4Admin System Microsoft-Windows-WindowsUpdateClient%4Operational	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-WindowsUpdateClient%4Operational	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational		Microsoft-Windows-NetworkProfile%4Operational System Microsoft-Windows-WindowsUpdateClient%4Operational
LD3	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Kernel-EventTracing%4Admin	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Kernel-EventTracing%4Admin	Microsoft-Windows-GroupPolicy%4Operational	Security Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	

Table F.3: Unique complete similar log types between the common behaviour of WKS-Froukje and WKS-Peter under 5-grams ($n = 5$)

n=5	BD	QD	ED	ESD	LD3
BD			Microsoft-Windows-Diagnosis-Scripted%4Operational	Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Microsoft-Windows-NetworkProfile%4Operational System
QD	Microsoft-Windows-Diagnosis-DPS%4Operational		Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Diagnosis-DPS%4Operational	Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-Diagnosis-DPS%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Microsoft-Windows-NetworkProfile%4Operational Microsoft-Windows-Diagnosis-DPS%4Operational System
ED	Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-Dhcp-Client%4Admin	Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-Dhcp-Client%4Admin		Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Security Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4WHC	Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-LanguagePackSetup%4Operational Microsoft-Windows-NetworkProfile%4Operational System
ESD	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Dhcp-Client%4Admin Microsoft-Windows-WindowsUpdateClient%4Operational	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Dhcp-Client%4Admin Microsoft-Windows-WindowsUpdateClient%4Operational	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational		Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-NetworkProfile%4Operational System Microsoft-Windows-WindowsUpdateClient%4Operational
LD3	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Dhcp-Client%4Admin	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-Dhcp-Client%4Admin	Microsoft-Windows-GroupPolicy%4Operational	Microsoft-Windows-GroupPolicy%4Operational	

F.1.2. In-Depth Log Comparison

Quadratic Delta

Regardless of the n-gram size, BD and QD exhibit identical ranking orders, indicating that both measures are consistent in pinpointing n-grams with minimal differences. However, BD assigns higher scores than QD, suggesting that BD perceives less similarity between the workstations than QD. This discrepancy stems from the inherent nature of the distance measures used.
BD employs the Euclidean distance, which is more sensitive to larger discrepancies due to its use of squared differences. This sensitivity amplifies the impact of any significant frequency differences between n-grams. Conversely, QD uses the Manhattan distance, which aggregates differences linearly through absolute differences, resulting in a more stable and moderated assessment. Consequently, the Euclidean distance often yields higher scores, especially in the presence of notable frequency variations, whereas the Manhattan distance provides a more balanced evaluation of similarity.
The following log types exhibit no deviation according to QD but demonstrate slight deviations when assessed using the BD:

- **Microsoft-Windows-TaskScheduler%4Operational**: This log type specifically tracks the tasks executed on the workstation. This log type exhibits no discrepancy between the workstations under QD unigrams.

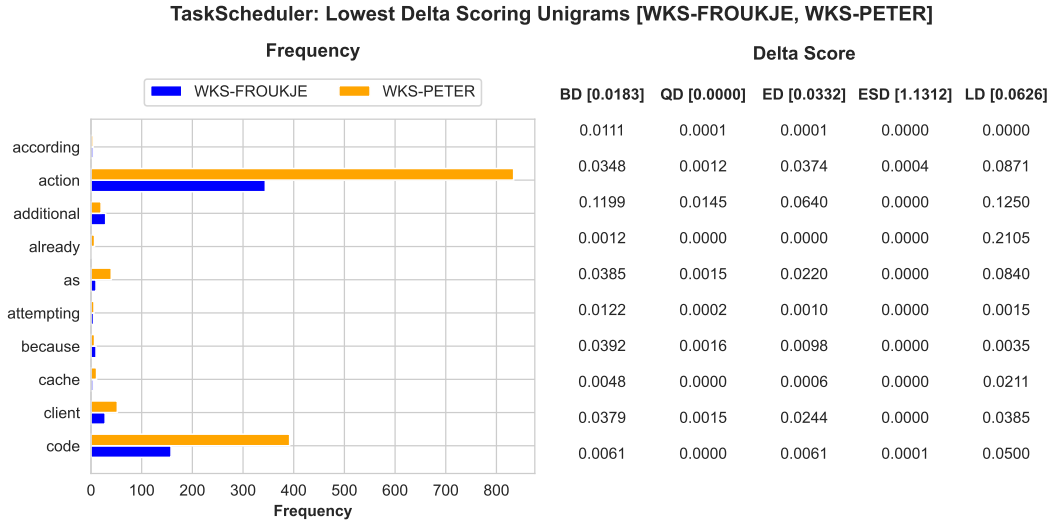


Figure F.1: The frequency of the 10 unigrams exhibiting the lowest delta score within the logtype `Microsoft-Windows-TaskScheduler%4Operational` of WKS-Froukje and WKS-Peter under BD-unigrams and QD-unigrams

Figure F.1 compares the top 10 unigrams that exhibit the lowest delta scores. Within this log type, 97% of the unigrams have a BD score below 0.3, and 100% of the unigrams have a QD score below 0.2. However, it is important to note that unigrams consider words in isolation without their surrounding context, which limits the understanding of the precise nature of common behaviour. For example, while *action* may appear frequently, the specific actions being performed and their context—such as the triggers or outcomes—are not fully elucidated by unigrams alone.

- **Microsoft-Windows-Diagnosis-PLA%4Operational**: This log type is part of Windows' Performance Logs and Alerts (PLA) service. This service allows for the collection of performance data and the generation of alerts based on thresholds. Specifically, the Operational log for this service records events related to the operation of diagnostic tasks and performance logging activities. This log type exhibits no discrepancy between the workstations under QD 4-grams, whereas, under unigrams, LD3 is the only delta method indicating dissimilarity between the workstations based on this log type.

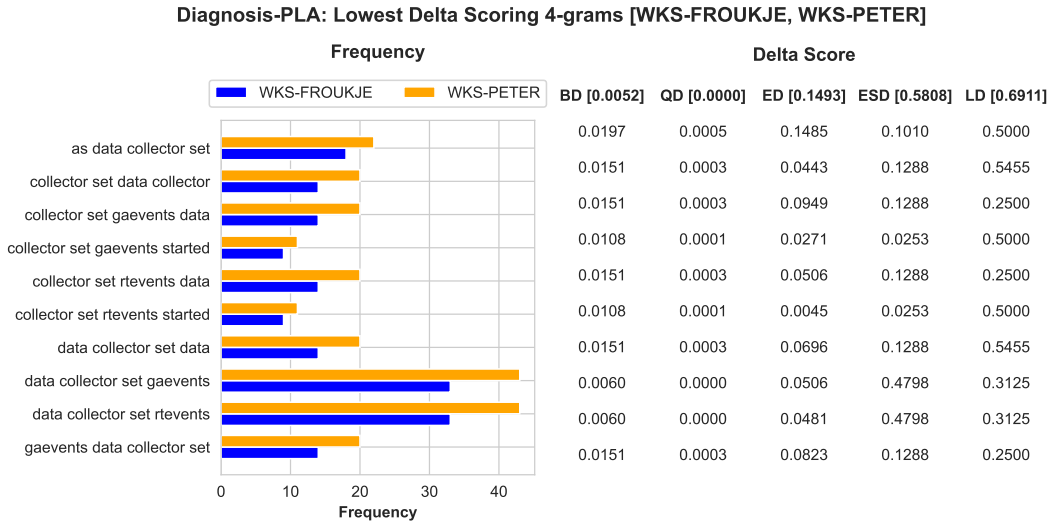


Figure F.2: The frequency of the 10 4-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-Diagnosis-PLA%4Operational` of WKS-Froukje and WKS-Peter under BD and QD 4-grams

Figure F.2 compares the top 10 4-grams that exhibit the lowest delta scores. Within this log type, all 4-grams have both BD and QD scores below 0.1. Remarkably, the recurring phrase *data collector set* alternating between *gaevents* or *rtevents* and either *started* or *data* suggests the presence of two distinct types of events being monitored or collected: *gaevents* relating to the system's general availability, and *rtevents* pertaining to real-time performance data. The terms *started* and *data* indicate the initiation and ongoing activity of these data collection processes, respectively. The consistent occurrence of these phrases in both workstations implies that both workstations are configured to collect data at similar frequencies and types related to diagnostic and performance monitoring. This consistency suggests a uniformity in the setup of diagnostic and performance monitoring features across both systems.

- **Microsoft-Windows-Diagnosis-DPS%4Operational**: This log type is associated with the Windows Diagnostic Policy Service (DPS) service. This service is responsible for detecting and troubleshooting system issues, with the Operational log specifically recording events related to the diagnostic processes and their outcomes. Notably, this log type shows no discrepancy between the workstations under QD 5-grams.

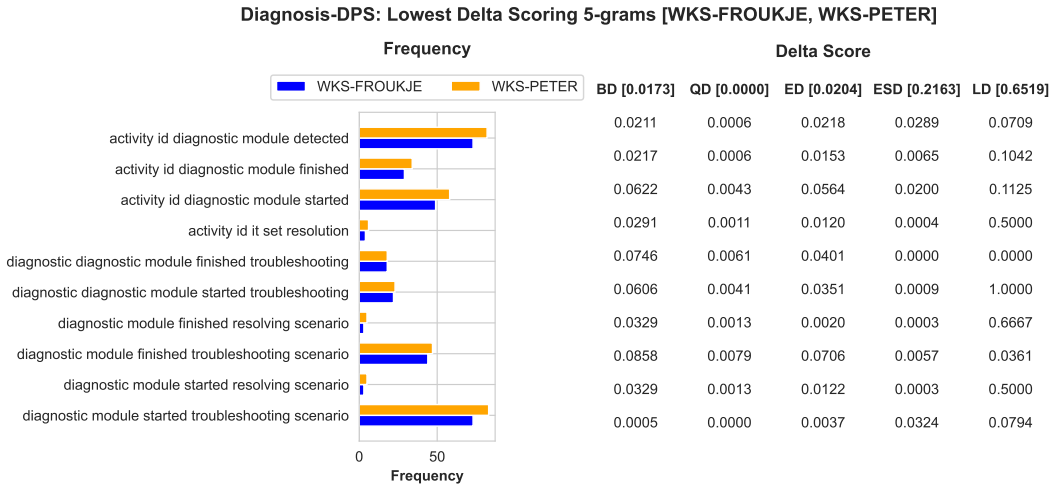


Figure F.3: The frequency of the 10 4-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-Diagnosis-DPS%4Operational` of WKS-Froukje and WKS-Peter under BD and QD 4-grams

Figure F.3 compares the top 10 5-grams that exhibit the lowest delta scores. Within this log type, 100% of the 5-grams have a BD score below 0.2, and 100% of have a QD score below 0.1. Notably, the 5-grams, 'activity id diagnostic module detected' and 'diagnostic module started troubleshooting' appear at comparable frequencies within both workstations. This indicates that both workstations detect specific activities and initiate troubleshooting scenarios to address the detected issues. These phrases in relatively equal frequencies suggest that both workstations are actively and consistently responding to diagnostic events. This similitude is more strongly reflected under QD than BD, as indicated by the lower delta scores for these phrases under QD.

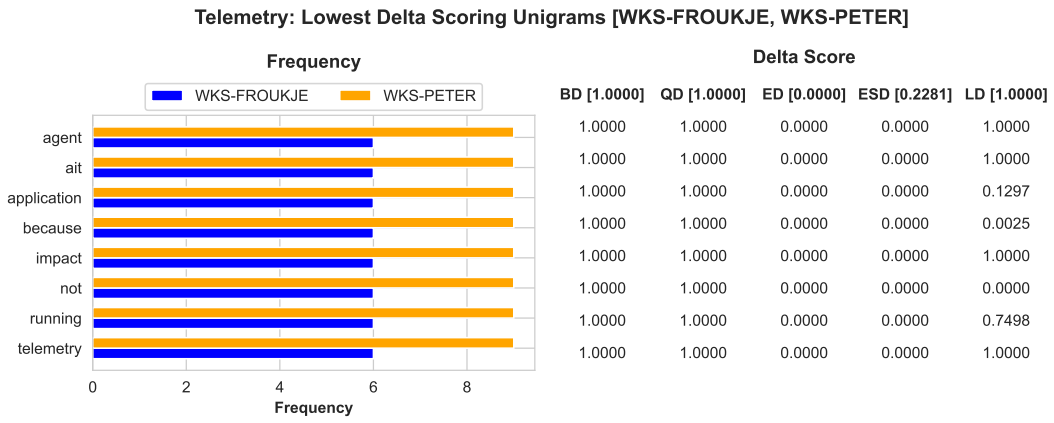
Regardless of the n-gram size, within the deviating log types, there is a high proportion of low delta scores under both BD and QD methods. Additionally, the inter-occurrence ratios - the relative frequency of one word to another within the same host - are identical. This consistency strongly supports QD's indication of no significant discrepancy between the workstations based on this log type.

Even though BD indicates minor discrepancies with delta scores around 0.018, which suggests subtle differences, QD's performance provides a more reliable picture. While present, the minor discrepancies highlighted by BD are not substantial enough to undermine the overall similarity between the workstations. Therefore, QD, by showing a more uniform absence of discrepancies, is better suited for this analysis because it offers a more stable and consistent evaluation.

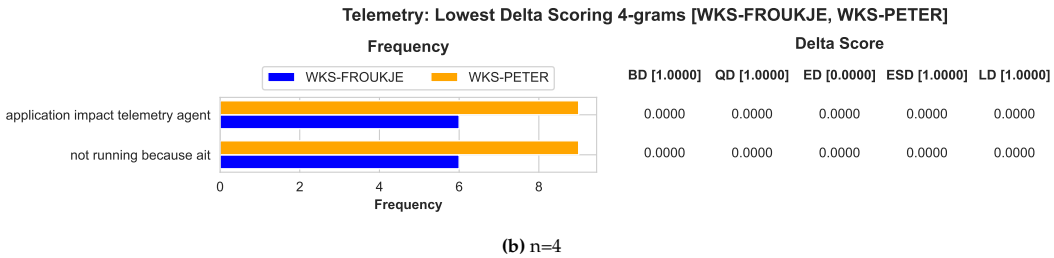
Eders Delta

Figures 5.14 to F.8 compare the uni- and 4-grams within the log types that uniquely exhibit no discrepancy between the workstations under ED.

- **Microsoft-Windows-Application-Experience%4Program-Telemetry**: This log type is part of the Microsoft Customer Improvement Program, which aims to improve system performance and functionality by collecting data on application operations. The Telemetry log type is primarily concerned with compatibility issues.



(a) n=1



(b) n=4

Figure F.4: The frequency of the 10 n-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-Application-Experience%4Program-Telemetry`of WKS-Froukje and WKS-Peter under BD and ED

- **Microsoft-Windows-PrintService%4Admin**: This log type is responsible for tracking printer usage on the system. This log type helps in identifying local issues with printing, user-specific problems, and ensuring that the print configurations are correctly set up.

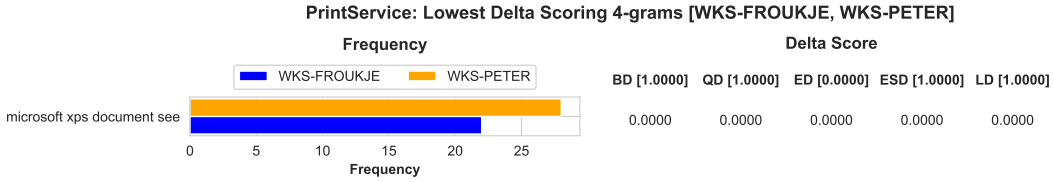


Figure F.5: The frequency of the 10 4-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-PrintService%4Admin` of WKS-Froukje and WKS-Peter under BD and QD 4-grams

- **Microsoft-Windows-Diagnosis-Scripted%4Admin**: This log type is part of the broader set of diagnostic logs in Windows. These logs specifically capture events generated by diagnostic scripts and tools that run on the system. These logs are valuable for understanding the effectiveness of automated diagnostics and for pinpointing the root causes of system problems.

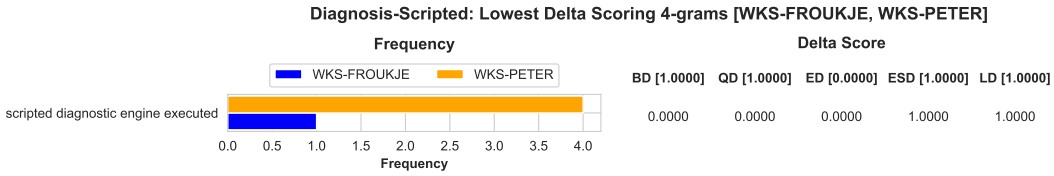


Figure F.6: The frequency of the 10 4-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-Diagnosis-Scripted%4Admin` of WKS-Froukje and WKS-Peter under BD and QD 4-grams

- **Microsoft-Windows-CodeIntegrity%4Operational**: This log type captures events related to the integrity and trustworthiness of code running on Windows machines. Code Integrity (CI) is a security feature that ensures that only trusted code runs in the Windows operating system. These logs are essential for maintaining system integrity, complying with security policies, and identifying potential security breaches or attempted compromises on workstations.

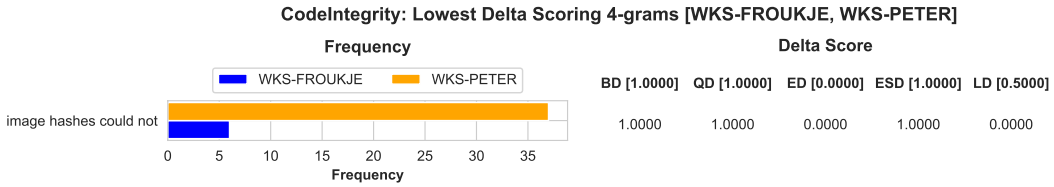


Figure F.7: The frequency of the 10 4-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-CodeIntegrity%4Operational` of WKS-Froukje and WKS-Peter under BD and QD 4-grams

- **Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational**: This log type focuses on events related to the detection and resolution of resource exhaustion issues on Windows systems. These logs allow to pinpoint the root causes of resource-related problems, take corrective actions, and implement strategies to prevent similar issues in the future.

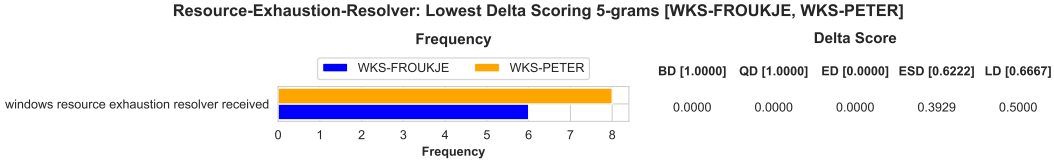


Figure F.8: The frequency of the 10 4-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational` of WKS-Froukje and WKS-Peter under BD and QD 4-grams

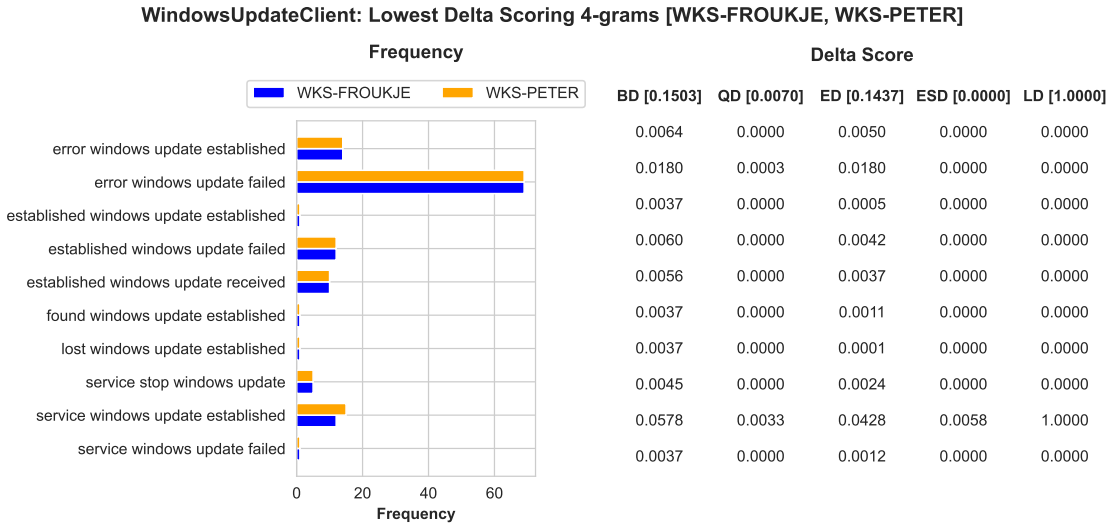
Within these log types, 100% of the unigrams under ED have a score of 0, while under BD, they have a score of 1. Conversely, 100% of the 4- or 5-grams within each of these log types have a score of 0 for both ED and BD. The consistent 0 scores are expected, as in log types with more than one unigram or 4-gram, each unigram and 4-gram frequency is constant across workstations, indicating no dissimilarity for this log type. Additionally, in other log types where only a single phrase appears for both workstations, this also indicates no dissimilarity since no other phrases are present. The absence of variation in these cases reinforces the conclusion that there is no dissimilarity between the workstations for this log type. This lack of dissimilarity is accurately reflected in the overall ED score of the log type but not in the overall BD score.

The discrepancy observed within the scores, where BD yields a score of 1 while ED yields a score of 0, arises from the normalisation process. Normalising considering all host pairs in the network ensures consistent comparison across different pairs. However, this can lead to an inflated BD score because BD, lacking the additional factor present in ED, is more susceptible to variations introduced by the normalisation across diverse hosts. In this case, the BD score of 1 is a result of the normalisation process, which amplifies minor differences when scaled against the entire network. In contrast, ED's additional adjustment mitigates this effect, maintaining a score of 0 to reflect true similarity accurately.

Hence, although BD and ED both employ the Manhattan distance and z-transformation, ED includes an additional ranking factor, allowing it to capture similarities more precisely. In contrast, the BD score is influenced by its sensitivity to variations, resulting in a score that does not accurately represent the specific log type's similarity between the workstations. This additional ranking factor in ED makes it a more reliable measure in cases where normalisation might otherwise introduce discrepancies, ensuring a correct reflection in the scores of the actual similarity.

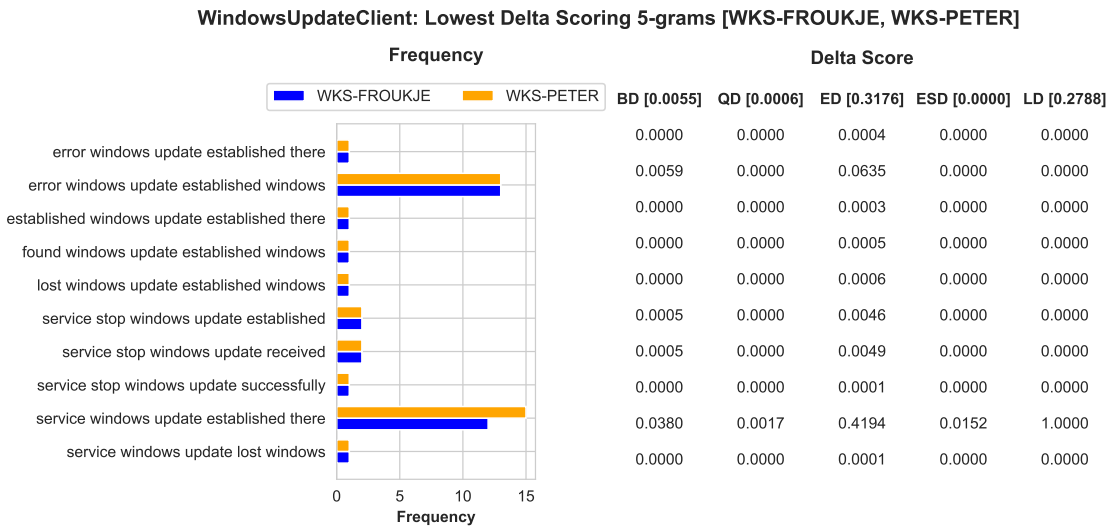
Eders Simple Delta

Figure F.10 compares the 4- and 5-grams within the logtype `Microsoft-Windows-WindowsUpdate Client%40perational`, which uniquely exhibits zero dissimilarity under ESD alone.



(a) n=4

Figure F.9: The frequency of the 10 4- and 5-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-WindowsUpdateClient%40perational` of WKS-Froukje and WKS-Peter



(a) n=5

Figure F.10: The frequency of the 10 4- and 5-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-WindowsUpdateClient%40Operational` of WKS-Froukje and WKS-Peter (cont.)

Within this log, 100% of the 4- and 5-grams under both BD and ESD score below 0.1, indicating a minor level of dissimilarity. Among the 4- and 5-grams, the 5-grams overlap significantly with the lowest identified 4-grams. The additional words in the 5-grams, such as *there*, *windows*, and *service*, do not provide essential information. Notably, the analysis of the 4-grams reveals that both workstations experience a similar number of Windows update failures. This pattern is not apparent in the 5-grams, as no phrases explicitly indicate the failures. This observation is crucial, as update failures are often associated with underlying issues. However, knowing that both workstations exhibit a similar number of failures provides reassurance that these failures occur at a regular rate.

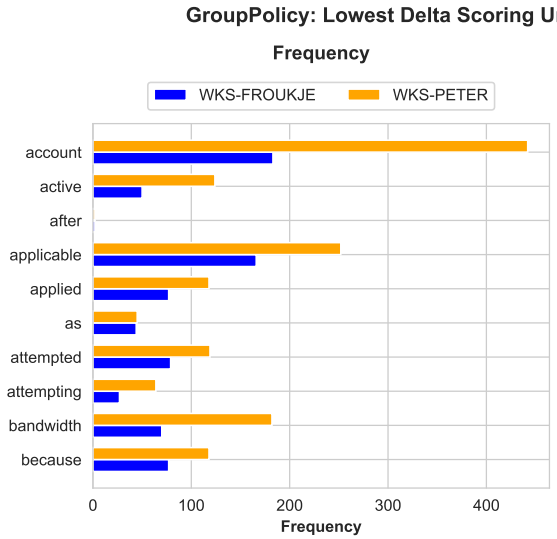
Within this log, the lowest-scoring n-grams exhibit identical frequencies for both workstations, except the phrase *service windows update established (there)*. Both BD and ESD rank this phrase as the most differing. However, BD assigns minor delta scores to the equally occurring phrases, which does not seem appropriate. These phrases appear in equal amounts, yet the BD scores show slight variations, inaccurately reflecting their identical frequencies.

Both BD and ESD employ the Manhattan distance, making both methods equally sensitive to the same deviations in the log. This leads to identical ranking orders of 4- and 5-grams when assessing minimal differences. However, the key difference lies in their standardisation techniques: BD uses z-transformation, while ESD employs square root normalisation. While both standardisations ensure that each n-gram contributes equally to the overall score, the standardisation choice significantly impacts the scores.
The z-transformation is sensitive to the log's statistical properties, such as mean and standard deviation. Even when n-grams have equal frequencies, the slight variations in these properties can result in different z-scores, leading to non-zero delta scores. Conversely, ESD directly compares the absolute frequencies. When frequencies are identical, the absolute difference is zero, and the square root of zero remains zero. This direct approach avoids the minor numerical errors introduced by statistical transformations.

Hence, ESD is more appropriate than BD because its use of square root normalisation ensures that identical n-gram frequencies yield a zero score, accurately reflecting their similarity. In contrast, BD's reliance on z-transformation introduces slight variations due to its sensitivity to the log's statistical properties, which can inaccurately depict identical frequencies differently. Thus, based on this log type, ESD provides a more precise and reliable measure of similarity between workstations.

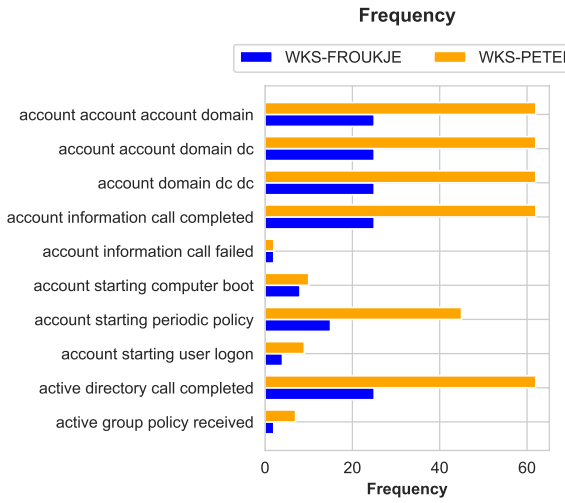
Linear Delta 3

Figure F.11 compares the uni-, 4-, and 5-grams within the `Microsoft-Windows-GroupPolicy%4Operational` log type. LD3 uniquely shows zero dissimilarity, except for 4-grams where ESD also indicates zero dissimilarity for this log type. The `GroupPolicy` log type relates to the Group Policy feature, which provides centralised management and configuration of operating systems and applications. This capability enables administrators to streamline operations, enhance security, and maintain a consistent and efficient computing environment across the organisation.



Delta Score				
BD [0.3371]	QD [0.1318]	ED [0.3342]	ESD [0.0239]	LD [0.0000]
0.3089	0.0955	0.2968	0.0145	0.0858
0.0747	0.0056	0.0401	0.0012	0.0858
0.0234	0.0006	0.0002	0.0000	0.0000
0.0076	0.0001	0.0071	0.0032	0.0469
0.0129	0.0002	0.0098	0.0007	0.1262
0.0597	0.0036	0.0200	0.0000	0.0078
0.0167	0.0003	0.0128	0.0007	0.1227
0.0246	0.0006	0.0110	0.0003	0.0858
0.1295	0.0168	0.1035	0.0025	0.1429
0.0129	0.0002	0.0094	0.0007	0.1262

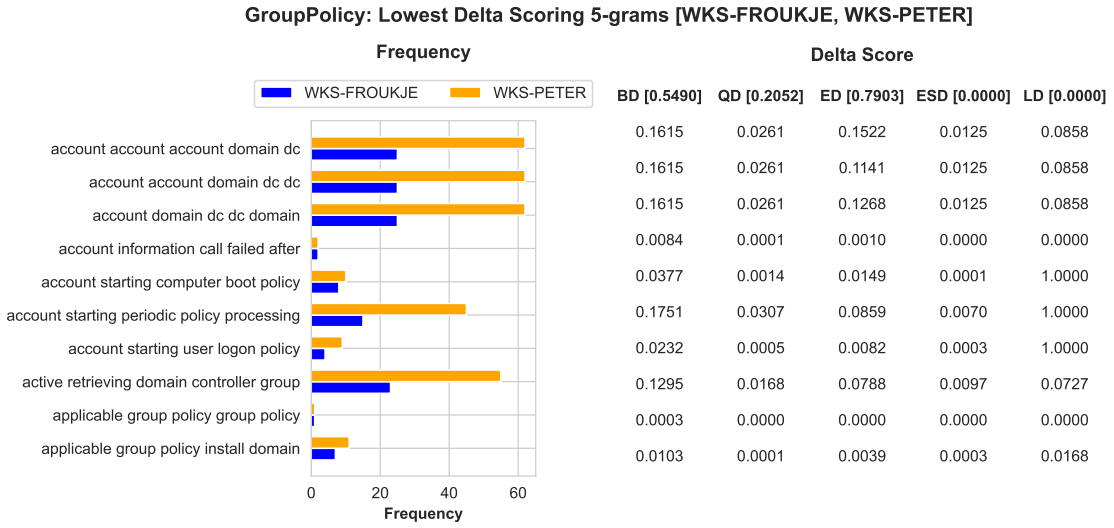
(a) n=1



Delta Score				
BD [0.4138]	QD [0.1558]	ED [0.5095]	ESD [0.0000]	LD [0.0000]
0.1583	0.0251	0.1262	0.0119	0.0858
0.1583	0.0251	0.1175	0.0119	0.0858
0.1583	0.0251	0.1110	0.0119	0.0858
0.1583	0.0251	0.1240	0.0119	0.0858
0.0342	0.0012	0.0047	0.0000	0.0000
0.0434	0.0019	0.0149	0.0001	1.0000
0.1451	0.0211	0.0718	0.0067	1.0000
0.0036	0.0000	0.0011	0.0002	1.0000
0.1583	0.0251	0.1414	0.0119	0.0858
0.0044	0.0000	0.0010	0.0002	1.0000

(b) n=4

Figure F.11: The frequency of the 10 n-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-GroupPolicy%40perational`of WKS-Froukje and WKS-Peter under BD and LD3 (cont.)



(c) n=5

Figure F.11: The frequency of the 10 n-grams exhibiting the lowest delta score within the logtype `Microsoft-Windows-GroupPolicy%4Operational`of WKS-Froukje and WKS-Peter under BD and LD3

Among the 4- and 5-grams, the 5-grams overlap significantly with the lowest identified 4-grams. The additional words in the 5-grams, such as *dc*, *domain*, and *policy*, do not provide essential information. Notably, the analysis of the 4-grams reveals slight differences between the workstations in phrases like *active directory call completed* and *active group policy received*, which were not apparent in the 5-grams. Thus, the 4-grams provide not only the same information as the 5-grams but also more detailed insights.

Within this log, the ranking order of the lowest-scoring n-grams differs between BD and LD3. However, regardless of the n-gram size, the inter-occurrence ratios between the workstations remain identical. BD more effectively identifies the similarity property of identical inter-occurrence ratios because its z-transformation normalisation minimises small deviations, resulting in minor dissimilarity scores for n-grams with similar frequencies. In contrast, LD3's Laplace normalisation emphasises differences more sharply, leading to higher dissimilarity scores even for slight deviations. This difference in normalisation affects the ranking order of the lowest-scoring n-grams.

Hence, LD3 proves least effective in identifying subtle similarities between workstations with identical inter-occurrence ratios. While both BD and LD3 employ the Manhattan distance, BD's z-transformation normalisation minimises slight deviations, resulting in minor dissimilarity scores for n-grams with similar inter-occurrences. In contrast, LD3's Laplace normalisation tends to emphasise differences more sharply, occasionally assigning a dissimilarity score of 1 to particular 4- and 5-grams. This exaggerated differentiation by LD3 is less appropriate given the identical inter-occurrence ratios.

F.2. Distinctive Behaviour

F.2.1. Overview Log Comparison

Here, Table F.4 presents for $n = 1, n = 4$ and $n = 5$ the log types that are identified as completely dissimilar within the distinctive behaviour in the considered delta method.

Table F.4: Completely dissimilar log types between the distinctive behaviour of WKS-Froukje and DC-Aphrodite under each delta method and n-gram size

	$n = 1$	$n = 4$	$n = 5$
BD	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Dhcp-Client%4Admin Microsoft-Windows-MUI%4Operational Microsoft-Windows-Diagnosis-DPS%4Operational Microsoft-Windows-CodeIntegrity%4Operational	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-Bits-Client%4Operational	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4Operational
QD	Microsoft-Windows-Kernel-WHEA%4Operational Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Dhcp-Client%4Admin Microsoft-Windows-MUI%4Operational Microsoft-Windows-CodeIntegrity%4Operational Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	Microsoft-Windows-GroupPolicy%4Operational Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-Bits-Client%4Operational	Microsoft-Windows-GroupPolicy%4Operational 'Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational Microsoft-Windows-WindowsUpdateClient%4Operational Microsoft-Windows-User Profile Service%4Operational Microsoft-Windows-Windows Defender%4Operational
ED	Microsoft-Windows-MUI%4Operational Microsoft-Windows-Diagnosis-DPS%4Operational Microsoft-Windows-CodeIntegrity%4Operational	Microsoft-Windows-User Profile Service%4Operational	
ESD	Microsoft-Windows-Diagnosis-Scripted%4Operational Security Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TaskScheduler%4Operational Microsoft-Windows-Diagnosis-Scripted%4Admin	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TaskScheduler%4Operational
LD3	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Diagnosis-Scripted%4Admin	Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Diagnosis-Scripted%4Admin	Microsoft-Windows-Diagnosis-PLA%4Operational Microsoft-Windows-Diagnosis-Scripted%4Operational

F.2.2. In-Depth Log Comparison

Quadratic Delta

Regardless of the n-gram size, BD and QD exhibit remarkable similarity in the log types that demonstrate complete dissimilarity, with only for each method a single log differing under unigrams:

- `Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall`: This log type captures information about firewall activities, such as allowed or blocked network connections, configuration changes, and security policy enforcement, providing insights into the firewall's operational status and security events.

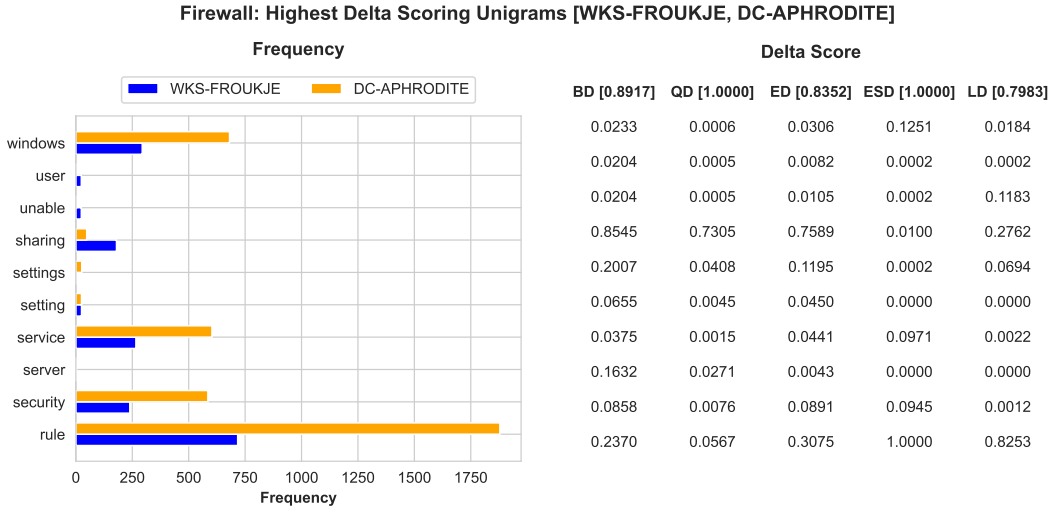


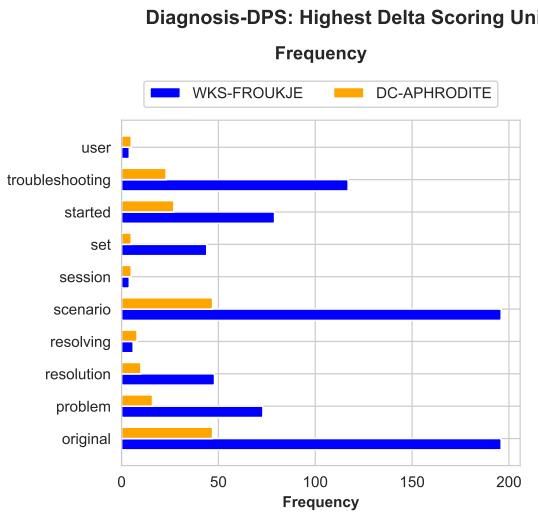
Figure F.12: The frequency of the 10 unigrams exhibiting the highest delta score within the logtype `Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall` of WKS-Froukje and DC-Aphrodite

Figure F.12 compares the highest differing unigrams within this log type. Here, BD, QD and ED exhibit identical ranking orders, all accurately identifying the unigram *sharing* as the most deviating term within this log. This unigram has a significantly higher frequency in WKS-Froukje than DC-Aphrodite, whereas overall, the unigrams show a higher frequency in DC-Aphrodite. This discrepancy likely arises due to functional operational differences between the two workstations.

Although QD indicates complete dissimilarity based on this log, BD and ED also signal substantial dissimilarity. Therefore, while these methods highlight the same deviating term and show significant dissimilarity, the divergence in their absolute dissimilarity scores suggests that no single method can be definitively chosen over the others based solely on this observation of this log type.

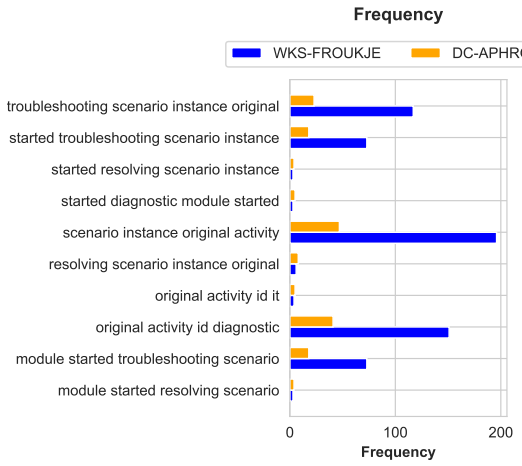
However, the unigrams *user* and *unable* exhibit slightly higher occurrences in WKS-Froukje than in DC-Aphrodite. BD and ED more strongly detect this difference than QD, which assigns a negligibly small delta score to these unigrams. This indicates that BD and ED are more sensitive to minor variations in unigram frequencies, making them better suited for identifying subtle distinctions between the two workstations.

- `Microsoft-Windows-Diagnosis-DPS%4Operational`: Recall that the similitude between the workstations for the log type `Diagnosis-DPS` is more strongly indicated by QD than BD. However, here, BD diverges from QD by showing complete dissimilarity between WKS-Froukje and DC-Aphrodite for this log type. ED also reflects this complete dissimilarity under unigrams. Nonetheless, it is important to note that the similitude is observed under 4-grams, whereas the dissimilarity is observed under unigrams.



Delta Score				
BD [1.0000]	QD [0.9759]	ED [1.0000]	ESD [0.7541]	LD [0.6553]
0.2338	0.0564	0.0484	0.0001	0.0000
0.6836	0.4694	0.4571	0.1933	0.5997
1.0000	1.0000	0.6160	0.0810	0.0130
0.9603	0.9225	0.3450	0.0281	0.0429
0.2338	0.0564	0.0606	0.0001	0.0000
0.2272	0.0534	0.2132	0.5318	0.5878
0.5859	0.3456	0.1808	0.0004	0.3810
0.3935	0.1572	0.1829	0.0324	0.5961
0.3213	0.1054	0.1664	0.0745	0.5984
0.2272	0.0534	0.2251	0.5318	0.5878

(a) n=1



Delta Score				
BD [0.8973]	QD [0.8359]	ED [0.9019]	ESD [0.7407]	LD [0.6165]
0.7947	0.6318	0.8079	0.2731	0.4498
0.1073	0.0117	0.1042	0.1038	0.4365
0.1808	0.0330	0.0435	0.0001	0.2500
0.3138	0.0988	0.1062	0.0003	0.5000
0.0314	0.0010	0.0371	0.7513	0.4408
0.6205	0.3854	0.3303	0.0006	0.2857
0.2831	0.0805	0.1370	0.0002	0.2500
0.5521	0.3052	0.5881	0.4382	0.4314
0.1073	0.0117	0.0990	0.1038	0.4365
0.1808	0.0330	0.0172	0.0001	0.2500

(b) n=4

Figure F.13: The frequency of the 10 n-grams exhibiting the highest delta score within the logtype `Microsoft-Windows-Diagnosis-DPS%40perational` of WKS-Froukje and DC-Aphrodite

Figure F.13 compares the highest differing uni- and 4-grams within this log type. BD, QD and ED exhibit identical ranking orders, all showing significantly high delta scores for most n-grams due to dissimilar inter-occurrences. All these methods indicate significant dissimilarity with a minor divergence in their absolute dissimilarity scores.

The n-grams within this log show variations in phrases where WKS-Froukje has a much higher frequency than DC-Aphrodite and phrases that appear slightly more in DC-Aphrodite. Given their operational differences, which of the two phenomena should be more heavily penalised is unclear. As BD, ED, and QD assign similar delta scores to the words and the logs, no definitive conclusion can be drawn based on this log type.

Eders Delta

ED is the only method showing complete dissimilarity between WKS-FROUKJE and DC-Aphrodite based on the log type `Microsoft-Windows-User Profile Service%40perational` under 4-grams. This log type captures operational events, such as successful or failed profile load and unload operations, errors, and other activities associated with managing user profiles. Figure 5.20 displays the highest differing 4-grams within this log.

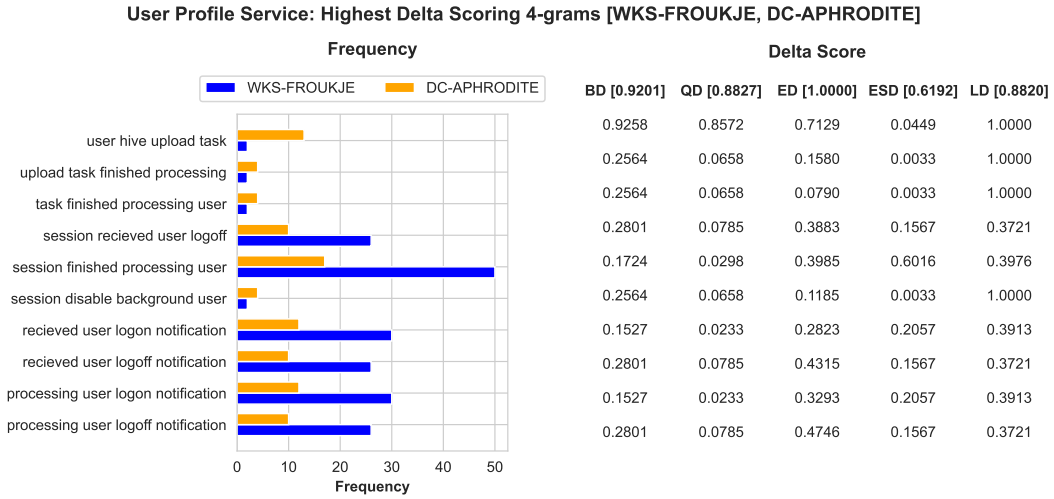


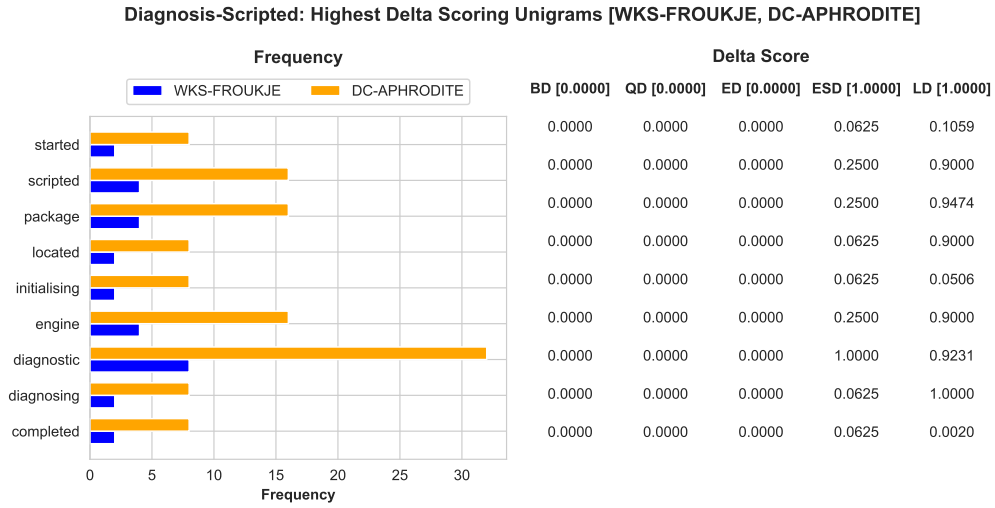
Figure F.14: The frequency of the 10 4-grams exhibiting the highest delta score within the logtype `Microsoft-Windows-User Profile Service%40perational` of WKS-Froukje and DC-Aphrodite

BD, QD, and ED exhibit identical ranking orders for the 4-grams, all identifying *user hive upload task* as the most differing activity between the hosts. In DC-Aphrodite, this 4-gram is more likely to occur frequently compared to the workstation WKS-Froukje due to the domain controller's role in managing multiple user profiles across a network, while workstations only manage profiles of users who log into that specific machine. This significant difference is highlighted by all the delta methods, except ESD.

The 4-gram *session disable background user* and *task finished processing user* appear to occur at equal rates for both hosts. BD, QD and LD3 assign the same scores to these phrases, while ED shows a slight difference in the delta score. ED's ability to detect these slight differences suggests it is more sensitive to nuanced variations. This sensitivity can be crucial when minor differences are significant. Therefore, ED's slight differentiation makes it preferable for analyses requiring detailed scrutiny of subtle differences.

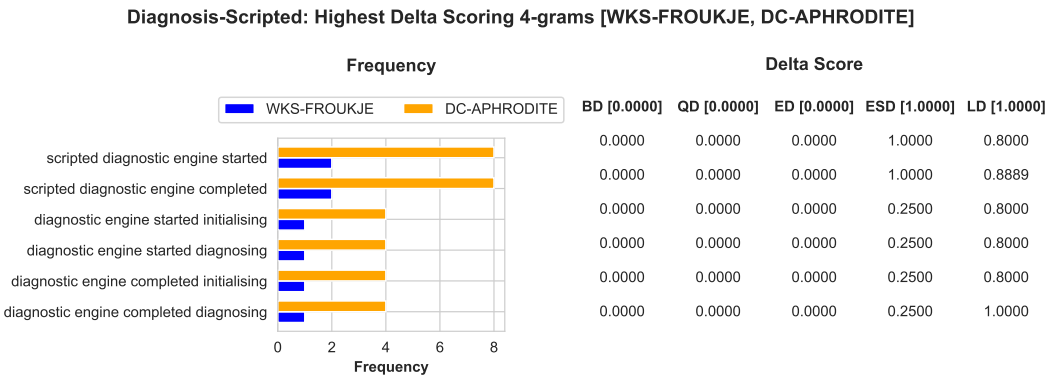
Eders Simple Delta & Linear Delta 3

ESD and LD3 both show, regardless of the n-gram, uniquely full dissimilarity between WKS-Froukje and DC-Aphrodite based on logtype `Microsoft-Windows-Diagnosis-Scripted%40perational`. Figure F.16 reveals the uni-, 4- and 5-grams within this log.

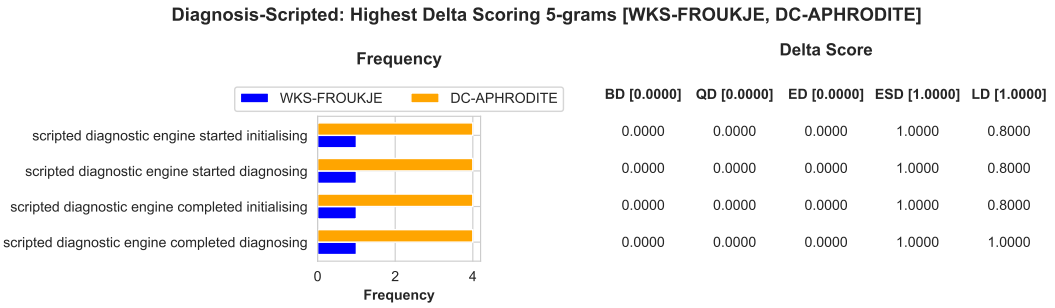


(a) n=1

Figure F.15: The frequency of the 10 n-grams exhibiting the highest delta score within the logtype `Microsoft-Windows-Diagnosis-Scripted%40perational` of WKS-Froukje and DC-Aphrodite (cont.)



(a) n=4



(b) n=5

Figure F.16: The frequency of the 10 n-grams exhibiting the highest delta score within the logtype `Microsoft-Windows-Diagnosis-Scripted%4Operational` of WKS-Froukje and DC-Aphrodite

ESD and LD3 consistently indicate complete dissimilarity across the n-grams, while QD, ED, and BD consistently indicate complete similarity. The uniform occurrence of each 5-gram suggests a high degree of similarity in the usage patterns between the hosts. Additionally, the uni-grams and 4-grams exhibit constant inter-occurrence ratios between the hosts, reinforcing the observation of complete similarity for these n-grams as well.

Hence, these findings imply that QD, ED, and BD are more sensitive to uniform frequency distributions, recognising identical patterns across hosts, while ESD and LD3 might be more attuned to capturing variations that result in a perception of dissimilarity. This indicates that, in contexts where uniformity in n-gram occurrences is evident, QD, ED, and BD provide a more accurate reflection of similarity between hosts.

F.3. Unique Behaviour

F.3.1. Overview Log Comparison

Here, Table F.5 presents for $n = 1, n = 4$ and $n = 5$ the log types that are identified as completely dissimilar within the unique behaviour in the considered delta method.

Table F.5: Completely dissimilar log types between the common behaviour of WKS-Froukje and WKS-Peter under each delta method and n-gram size

	$n = 1$	$n = 4$	$n = 5$
BD	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry	Microsoft-Windows-PrintService%4Admin Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational	Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Dhcp-Client%4Admin
QD	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry	Microsoft-Windows-PrintService%4Admin Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational	Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational Microsoft-Windows-Dhcp-Client%4Admin
ED	Microsoft-Windows-Kernel-EventTracing%4Admin Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-Dhcp-Client%4Admin	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational	Microsoft-Windows-Diagnosis-Scripted%4Operational Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational
ESD	Microsoft-Windows-Dhcp-Client%4Admin Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational	Microsoft-Windows-PrintService%4Admin Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational	
LD3	Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Dhcp-Client%4Admin Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-CodeIntegrity%4Operational	Microsoft-Windows-PrintService%4Admin Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Application-Experience%4Program-Telemetry Microsoft-Windows-Diagnosis-Scripted%4Admin Microsoft-Windows-WindowsUpdateClient%4Operational	Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational Microsoft-Windows-Bits-Client%4Operational

F.3.2. In-Depth Log Comparison

Quadratic Delta & Eders Simple Delta

QD and ESD , regardless of the n-gram, uniquely show no complete dissimilarity between WKS-Froukje and WKS-Peter based on logtype `Microsoft-Windows-TerminalServices-RemoteConnectionManager%40perational`. Figure F.17 reveals the 4-grams within this log since under 4-grams all remaining delta methods consistently show full dissimilarity based on this log.

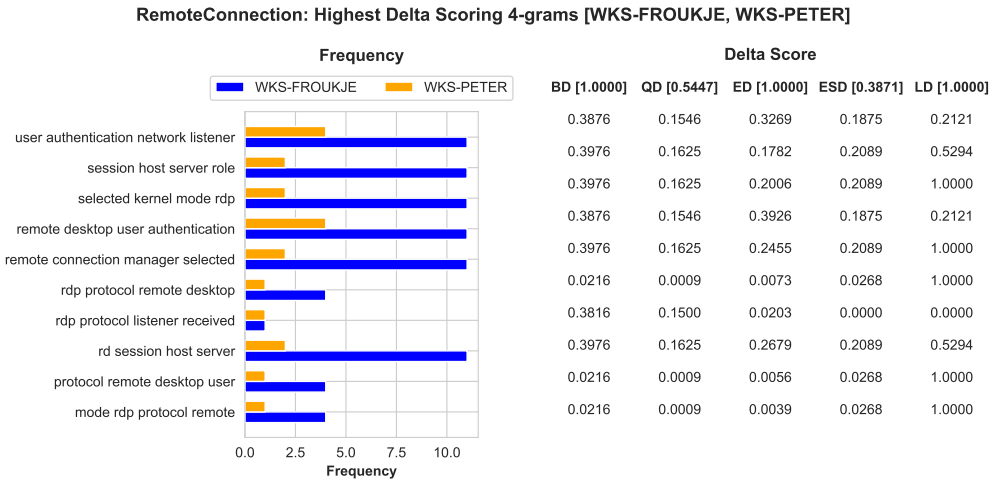


Figure F.17: The frequency of the 10 4-grams exhibiting the highest delta score within the logtype `Microsoft-Windows-TerminalServices-RemoteConnectionManager%40perational` of WKS-Froukje and WKS-Peter

Three distinct frequency scores are observed in the highest delta scoring 4-grams of WKS-Froukje. This pattern is also seen in WKS-Peter. However, the 4-grams are not assigned equivalent levels in WKS-Froukje and WKS-Peter. Additionally, the distribution of levels does not match between the workstations: WKS-Froukje assigns the highest level to the majority of the 4-grams, whereas WKS-Peter distributes them more evenly.

The 4-grams with mismatching levels are assigned the highest delta score under BD, QD, and ESD. Yet, the matching levels are assigned a negligibly smaller score. In contrast the matching levels are assigned a higher score under ED, while the mismatching ones have a significantly lower score. This is counterintuitive, as it would be expected that matching levels receive smaller scores than mismatching ones to indicate a greater level of similarity.

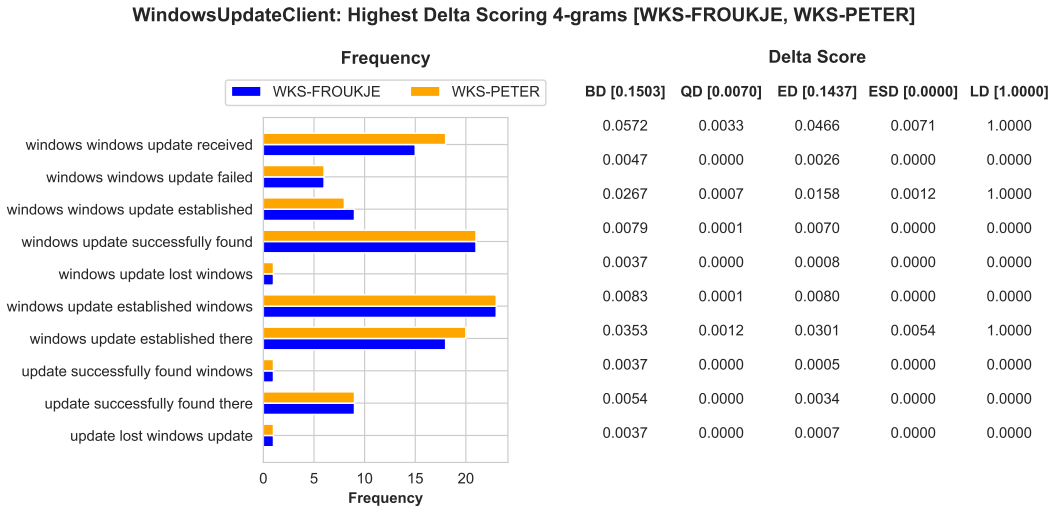
The majority of the 4-grams, precisely 66%, have mismatched levels. With an overall delta score of 0.54, QD better captures this disparity compared to ED, which assigns a score of 0.39. The full dissimilarity claimed by BD could be appropriate given the significant amount of mismatching levels and the misaligned distribution of the levels. However, it overlooks the fact that there is some overlap and that both workstations exhibit three frequency levels. Hence, based on this log type, QD seems most suitable for assessing the uniqueness between hosts.

Eders Delta

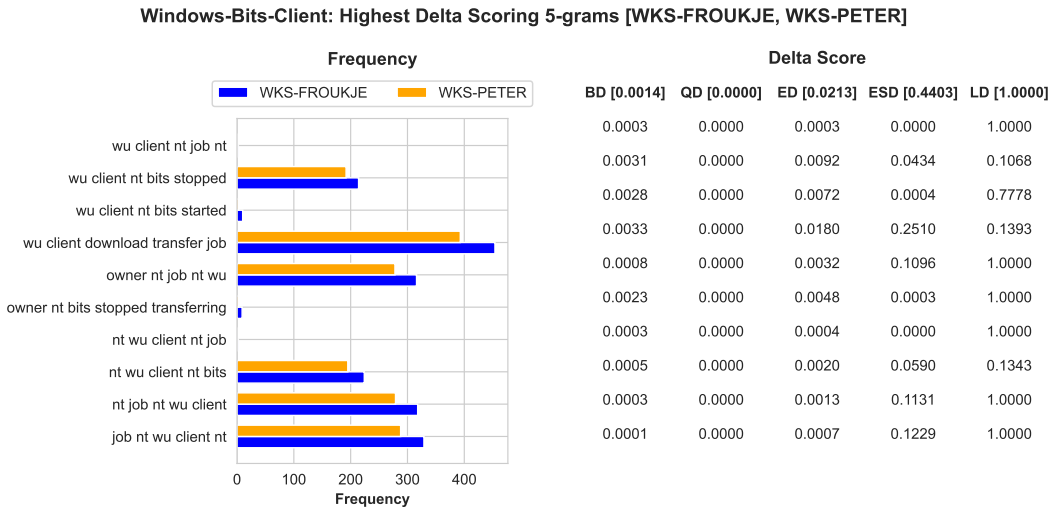
ED is the only method showing no complete dissimilarity between WKS-Froukje and WKS-Peter based on logtype `Microsoft-Windows-Application-Experience%4Program-Telemetry` under 4-grams. Figure F.4b reveals the 4-grams within this log. As discussed in subsection 5.2.1 Common Behaviour - Eders Delta, the frequency of both 4-grams appears at a constant rate for both hosts, indicating no dissimilarity between the hosts based on this log type. Hence, ED is the only method that accurately reflects this.

Linear Delta 3

Linear Delta 3 is the only method showing complete dissimilarity between WKS-Froukje and WKS-Peter based on logtype `Microsoft-Windows-WindowsUpdateClient%40perational` under 4-grams and `Microsoft-Windows-Bits-Client%40perational` under 5-grams. Figure F.18 reveals the 4- and 5-grams within the logs, respectively.



(a) n=4



(b) n=5

Figure F.18: The frequency of the 10 n-grams exhibiting the highest delta score within the logtype `Microsoft-Windows-Bits-Client%40perational` of WKS-Froukje and WKS-Peter

Within the top 10 n-grams of both log types, the ranking order from highest to lowest frequency is identical for both hosts. However, the relative frequency differences between these n-grams are inconsistent across the hosts. Specifically, if the frequency of an n-gram for one host decreases by a certain factor, the corresponding n-gram for the other host does not necessarily decrease by the same factor.

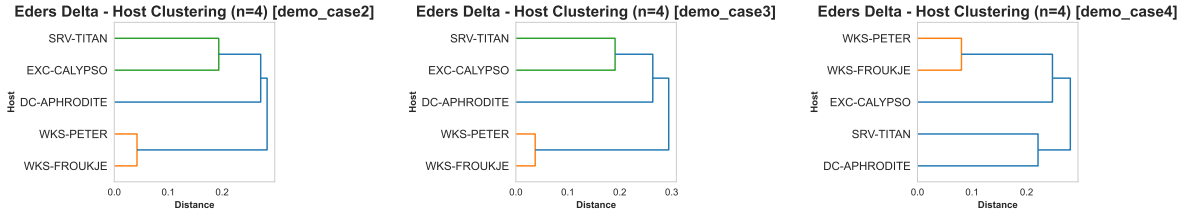
Hence, the assertion of complete dissimilarity between the hosts based on LD3 renders incorrect, given the identical frequency order of n-grams. Conversely, the assertion of complete similarity as indicated by ESD under 4-grams or QD under 5-grams yields also inaccurate due to the observed relative inconsistency in frequencies. A slight presence of dissimilarity, as indicated by ED, is expected. The dissimilarity indicated by BD appears negligible under 5-grams.

G

Clustering and Top 10 Behaviour

Eders Delta method using n-gram size four has been applied to the Demo-Case 2, 3 and 4. section G.1 provides the obtained clusterings of the Demo-Cases. Then, section G.2 presents the top 10 behaviour of each host within the Demo-Case. The behaviour is classified into three types, namely, common, distinctive and unique. Moreover, the behaviour is expressed in terms of log types and 4-grams.

G.1. Clustering



(a) Demo-Case 2 **(b)** Demo-Case 3 **(c)** Demo-Case 4

Figure G.1: Clustering of Demo-Cases 2, 3, and 4 based on Eders Delta ($n = 4$)

G.2. Top 10 Behaviour

G.2.1. Common Behaviour

Figure G.2: Top 10 Common Behaviour in terms of logs and words for Demo-Case 2

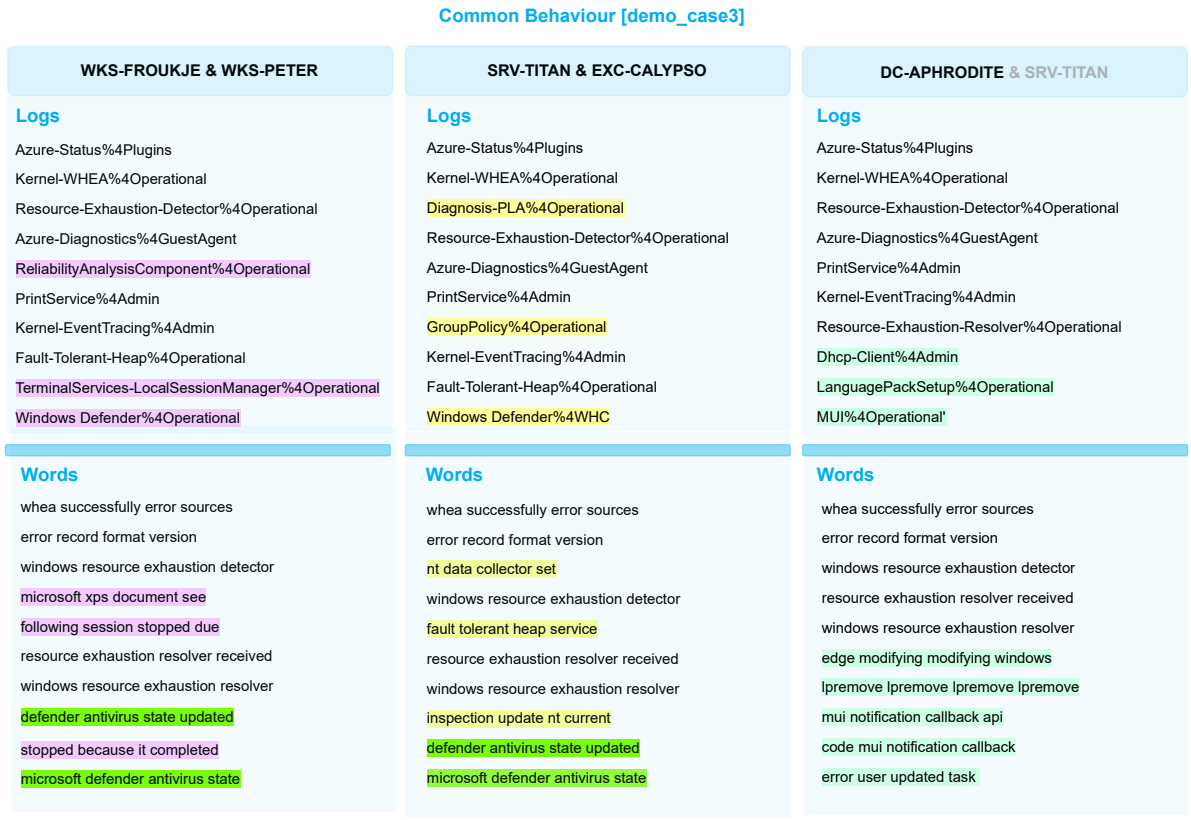


Figure G.3: Top 10 Common Behaviour in terms of logs and words for Demo-Case 3

Figure G.4: Top 10 Common Behaviour in terms of logs and words for Demo-Case 4

G.2.2. Distinctive Behaviour

Figure G.5: Top 10 Distinctive Behaviour in terms of logs and words for Demo-Case 2

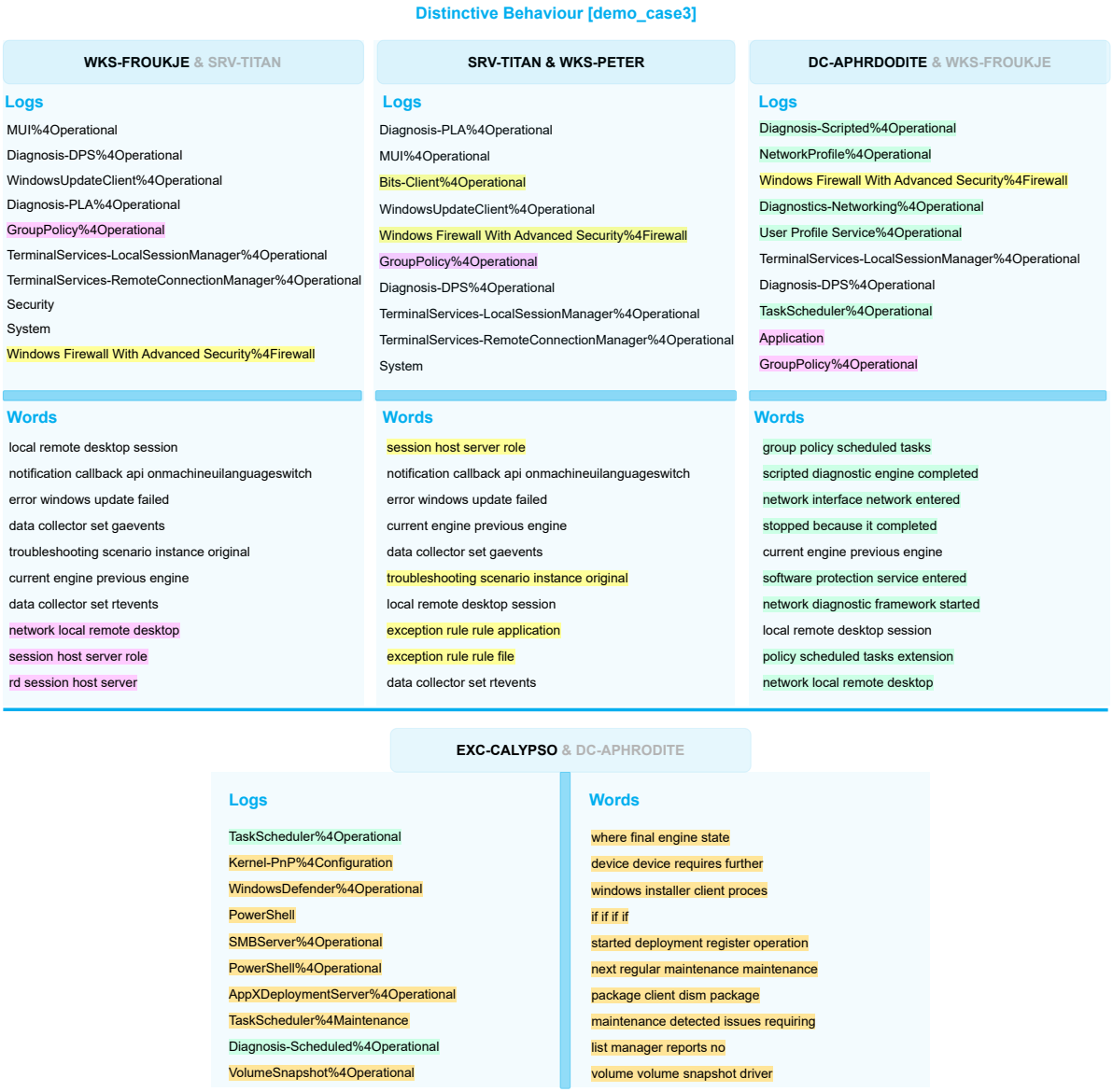


Figure G.6: Top 10 Distinctive Behaviour in terms of logs and words for Demo-Case 3

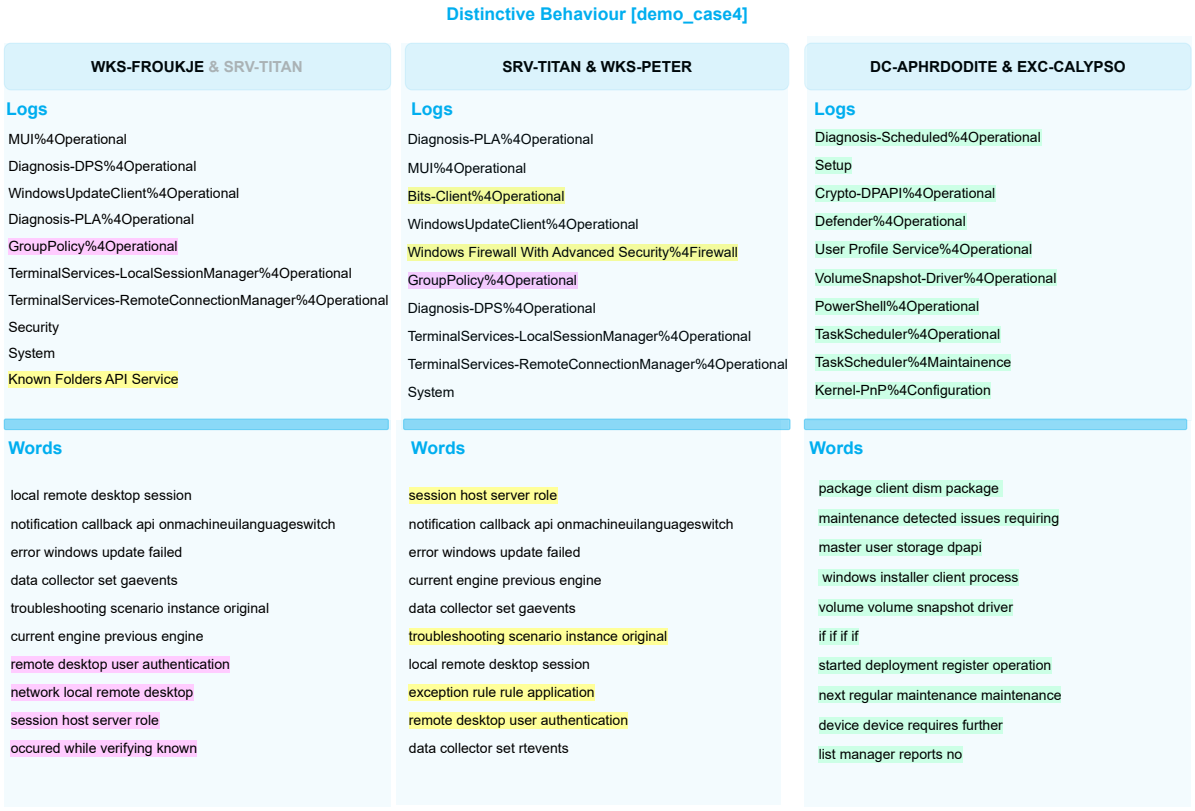


Figure G.7: Top 10 Distinctive Behaviour in terms of logs and words for Demo-Case 4

G.2.3. Unique Behaviour

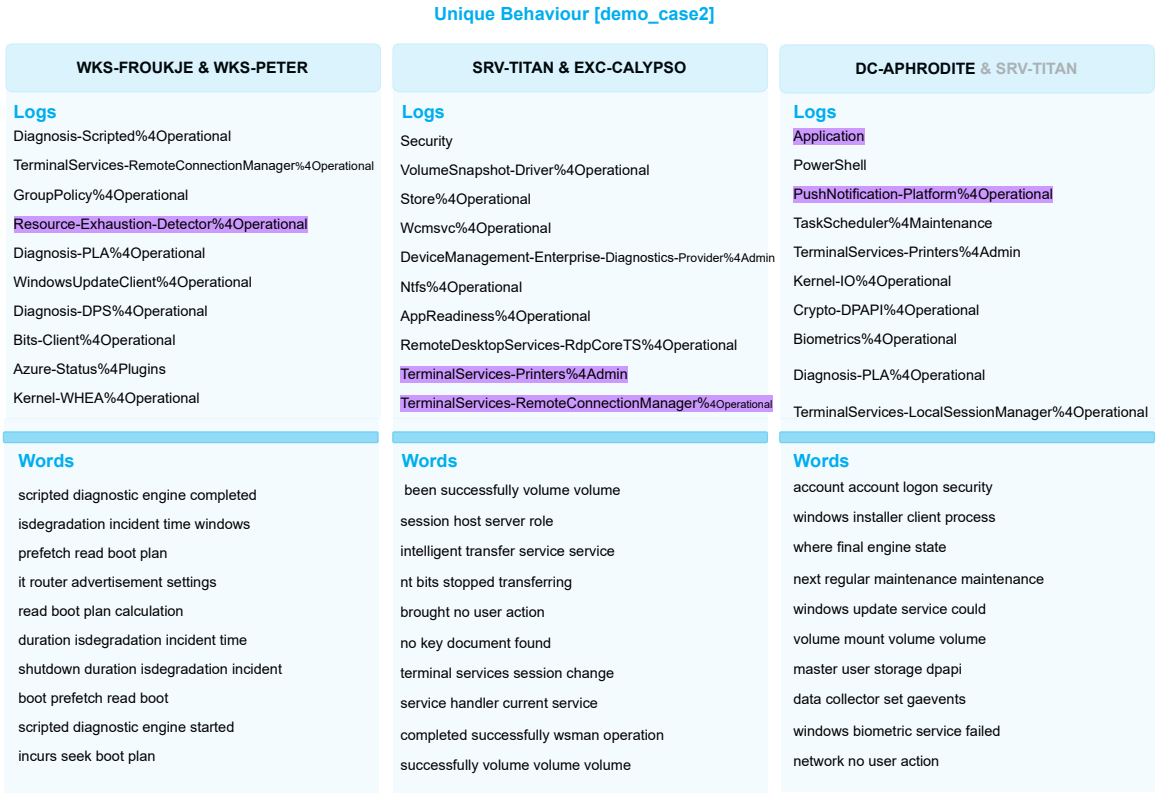


Figure G.8: Top 10 Unique Behaviour in terms of logs and words for Demo-Case 2

Figure G.9: Top 10 Unique Behaviour in terms of logs and words for Demo-Case 3

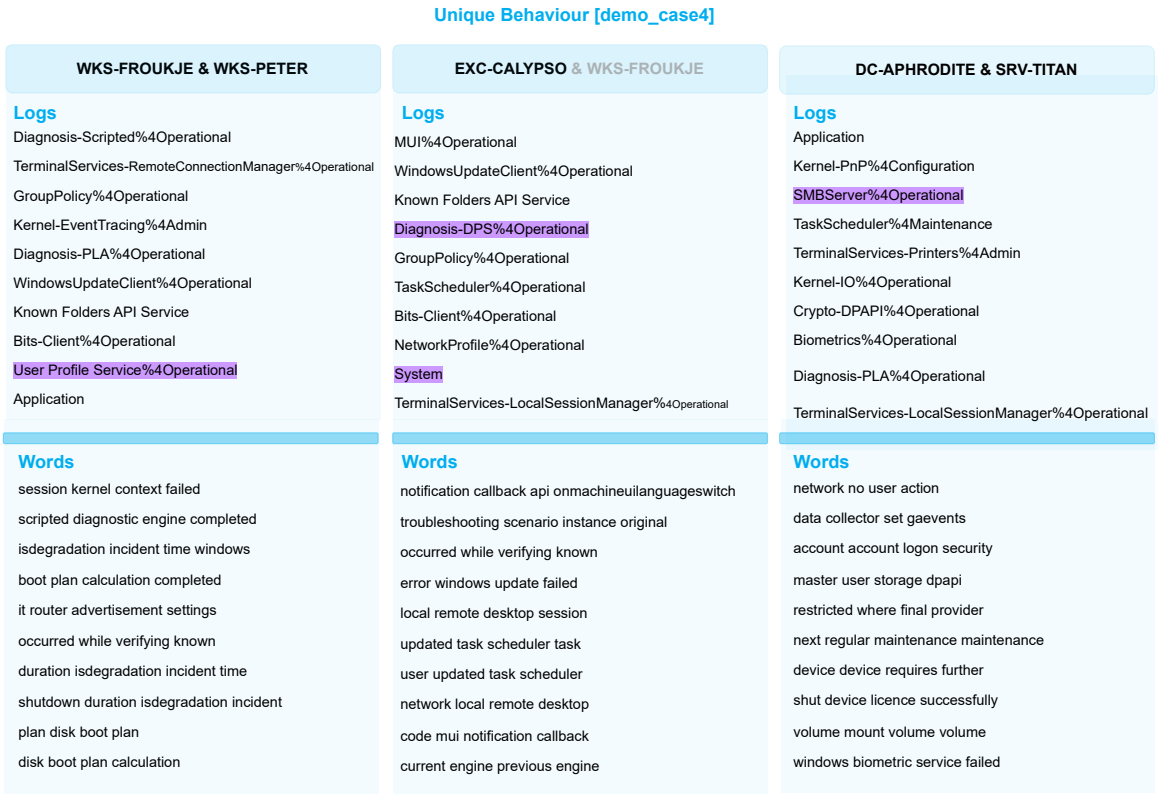


Figure G.10: Top 10 Unique Behaviour in terms of logs and words for Demo-Case 4