Poster: Using Machine Learning to Infer Network Structure from Security Metadata

Asfa Khalid, Seán Óg Murphy, Cormac J. Sreenan, and Utz Roedig

School of Computer Science and Information Technology (CSIT),
University College Cork, Cork, Ireland
123111938@umail.ucc.ie, {seanogmurphy, cormac.sreenan, u.roedig}@ucc.ie
https://ucc.ie/nascresearch/

Abstract. In distributed cloud-edge environments, data-driven decision-making is essential for enhancing operational efficiency and maintaining a competitive advantage. Achieving this requires strong guarantees of data integrity and authenticity, as any compromise can lead to inaccurate insights, loss of trust, and financial damage. To address the cybersecurity risks posed by data transmission across complex, heterogeneous networks, Data Confidence Fabrics have been introduced. These enhance data security by generating metadata at each stage of transmission and storing it using distributed ledgers, which ensures the immutability and verifiability of this metadata. However, despite these benefits, the public accessibility of ledgers introduces significant privacy concerns. While previous research has focused on hostname obfuscation to protect network structure, timestamps often remain exposed, creating an exploitable vulnerability. We demonstrate that one can use K-means clustering on exposed timestamp patterns to reconstruct the obfuscated network structure, even when hostnames are fully obfuscated. Our findings reveal a critical gap in existing metadata protection mechanisms and highlight the need for defense against timestamp-based inference attacks.

Keywords: Distributed Systems, Data Confidence Fabrics, Machine Learning, K-means Clustering, Timestamp Analysis, Metadata Privacy, Network Structure Inference

1 Introduction

Research has shown that metadata, even when seemingly innocuous, can pose serious privacy threats. Studies across social networks, ISPs, and financial systems demonstrate how elements like timestamps, activity logs, and network structures can be exploited to identify users, infer behaviors, and reveal organizational patterns. For networks, techniques such as DNS encryption [3], network topology disguising [4], and web data hiding [10] aim to shield sensitive structures and activities.

Data Confidence Fabrics (DCFs) have been introduced to enhance data security for cloud-edge systems by generating metadata (annotations) at each stage of transmission, recording details about network nodes [1], and storing this information in distributed ledgers. This is illustrated in Figure 1, which depicts the Linux Foundation's Alvarium Data Confidence Fabric (DCF). Since ledgers ensure immutability, metadata remains secure and verifiable. However, despite these advantages, the public accessibility of metadata on ledgers presents significant privacy concerns: attackers can analyze metadata to infer network structure. By analyzing network metadata, attackers can uncover structural details, identify critical nodes, and assess vulnerabilities, enabling them to plan precise attacks. In particular, stored metadata includes a timestamp field and host identifiers, allowing adversaries to infer the relationships between network nodes. Therefore, safeguarding the confidentiality of annotations has practical significance.

To address the challenge of protecting network structure information, encryption-based obfuscation offers stronger security by preventing meaningful pattern extraction. Prior work by us and others used encryption to effectively concealing hostname information [6] [7], but timestamp data remains a critical vulnerability. We show in this short paper that adversaries can use timestamp clustering techniques, such as K-means clustering, to identify communication patterns, reconstruct network layers, and infer host relationships.

This research explores how the timing of network communications can reveal relationships between hosts using machine learning techniques such as K-means clustering, even when hostnames are encrypted. Using a dataset of network annotations with encrypted hostnames, we analyzed time intervals between nodes and applied K-means clustering to group hosts with similar timing patterns. Our method accurately maps encrypted hostnames to their real counterparts, effectively reconstructing much of the original network structure. Experimental results demonstrate that timing-based clustering remains highly effective for inferring network structure, even without access to hostname information. These findings highlight a significant vulnerability in current metadata protection strategies and underscore the need for greater consideration of timestamp privacy in the design of secure distributed systems.

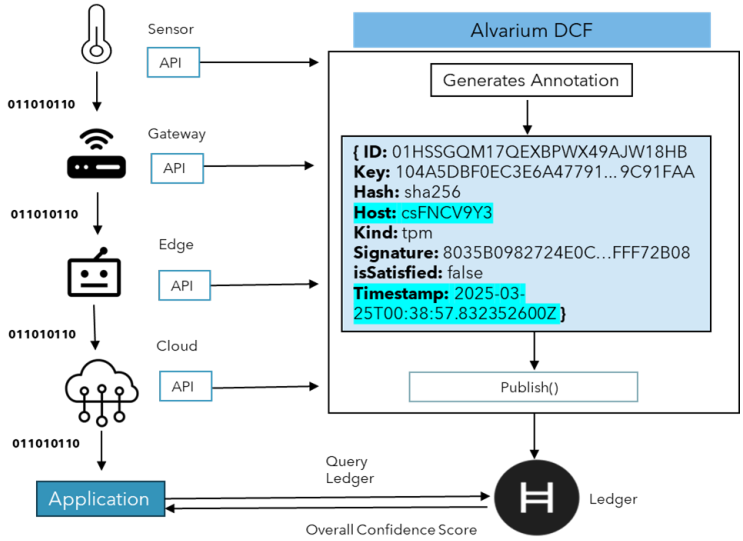


Fig. 1: Alvarium DCF publishes the generated annotations to the ledger, revealing host and timestamp information that attackers could exploit to infer the network structure.

2 Methodology

Our approach uses K-means clustering on timestamp metadata to successfully infer the original network hosts 2. By carefully finding the exact K-means value and accurately clustering hosts based on their timestamps, we reconstructed most of the network structure.

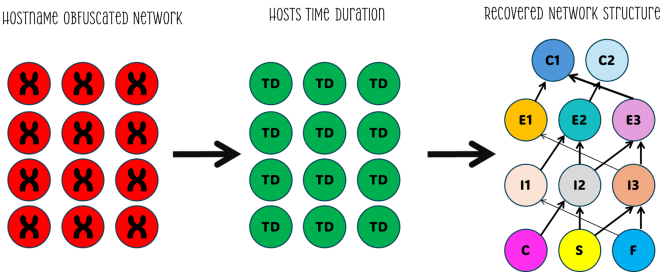


Fig. 2: Inferring obfuscated network based on timestamp clustering

K-means clustering [11], widely applied in pattern analysis, has been adapted for spatio-temporal and large-scale datasets. Rather than enhancing clustering accuracy, we highlight how K-means can be used maliciously to extract network insights from timestamp metadata. This perspective reframes clustering not just as an analytical tool but also a potential vector for privacy attacks, reinforcing the need for metadata obfuscation within DCFs.

The first step in our approach involves organizing the network into distinct layers by analyzing annotation keys in combination with their corresponding timestamps. Each annotation key, generated through a hashing process, is unique to a specific data point at a particular node. This uniqueness allows us to consistently trace and identify the location of each host within the network. By examining when and where these keys appear, we can assign each host to a specific layer within the system. In this context, a layer represents a logical level within the network, used to categorize hosts based on the timing and sequence of their communications. As a result, a hierarchical model of the network emerges, clearly outlining whether a host operates in the first, second, or subsequent layer based on its interaction patterns.

The second step involves preparing the annotation dataset for clustering analysis. We restructure the data to focus on temporal differences between hosts, allowing us to analyze the timing relationships among them. To identify the optimal number of clusters (denoted as K), we employ an iterative K-means clustering approach. Starting with a small value for K, we incrementally increase it, assessing the quality and cohesion of the resulting clusters at each step. The goal is to find a value of K that leads to meaningful and interpretable groupings of hosts rather than arbitrary separations. Through this process, we identify that $K = 9$ produces the most accurate and consistent clustering, aligning well with known host behavior and network structure.

Following the clustering process, we assign each host a unique, structured identifier based on its original layer and its cluster number, which groups hosts by the time gaps in their communication patterns. The merged value containing host's original layer with cluster number form a unique identifier that reflects both temporal and structural characteristics. For example, a host in Layer 1

assigned to Cluster 1 would be labeled as "H1-1." Similarly, a host in Layer 4 grouped into Cluster 9 becomes "H4-9." These new identifiers replace the original encrypted hostnames in the dataset.

With hosts now labeled according to these structured identifiers, we proceed to reconstruct the network topology. This involves mapping relationships between hosts based on their cluster groupings.

3 Evaluation

To evaluate the selection of the K value and the effectiveness of our clustering approach in grouping original hosts to recover network structure based on them, we use two methods. First, we compare the results of different cluster evaluation metrics (Silhouette Score, Davies-Bouldin Index, Dunn Index, and the Elbow Method) to find the optimal K value. Second, we use the original hostnames as a reference dataset and compare them with our clustering results. We base our analysis on a dataset of 4,000 annotated timestamped communication events, where each original host within an annotation is encrypted. These annotations were generated using the Alvarium SDK [2], simulating network behavior, including timing durations and host communication patterns.

Evaluation of Optimal K for Clustering To determine the optimal number of clusters for grouping similar communication patterns, we used several clustering metrics across K values ranging from 5 to 15. Specifically, we used the Silhouette Score [8], Davies-Bouldin Index (DBI) [12], Dunn Index [9], and the Elbow Method [5], each offering a different perspective on clustering quality. These metrics assess factors such as cohesion within clusters, separation between clusters, and the balance between compactness and spread. By analyzing these scores collectively, we aimed to identify the most meaningful and stable value of K for our dataset.

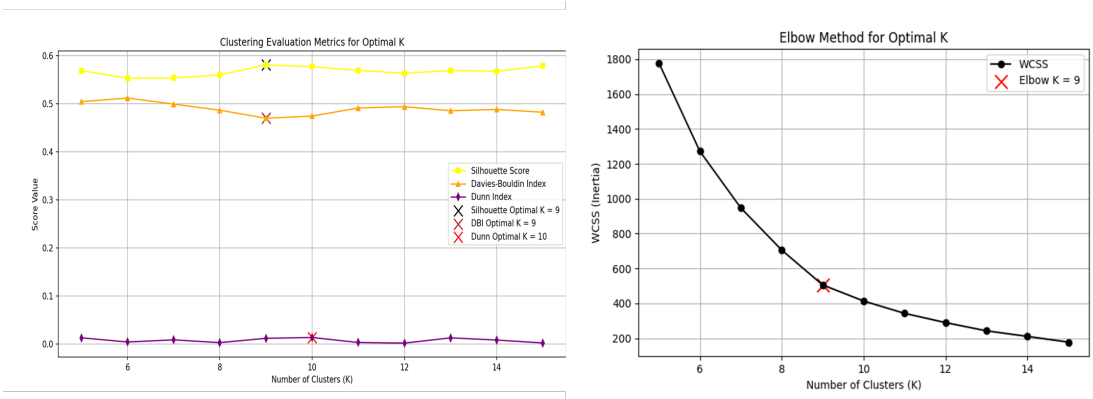
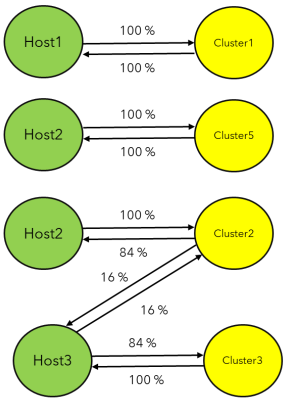


Fig. 3: Scoring result of cluster evaluation metrics for optimal K

Based on this multi-metric analysis (Figure 3), both $K = 9$ and $K = 10$ emerge as optimal choices. The Dunn Index suggests $K = 10$ as a strong alternative, with slightly better inter-cluster separation. However, $K = 9$ is the most consistently optimal choice, as reflected by the Silhouette Score, DBI, and Elbow Method, all of which indicate well-separated and compact clusters.

Evaluation of Clustering Accuracy through Bidirectional Validation To further evaluate the accuracy of our clustering approach, we employed a bidirectional validation method that compares the clustered hostnames against the real hostnames. This evaluation involved two complementary perspectives. The first, referred to as forward mapping, examined how instances of each real hostname were distributed across the resulting clusters. The second, reverse mapping, assessed how well each cluster corresponded to a specific original hostname.



Real Host	Dominant cluster percentage	Subordinate cluster percentage	Std Dev	Number of clusters
asFNC00Y4	100		0	1
asFNC00Y7	100		0	1
bsANC89Y8	100		0	1
bsFNC89Y4	100		0	1
csANC89Y9	69.23	30.77	27.2	2
csANCV8Y6	100		0	1
csBNCV8Y5	100		0	1
csBNCV9Z3	83.33	16.67	47.14	2
csFNC89Y10	100		0	1
csFNCV8Y3	84	16	48.08	2

Cluster	Dominant real host percentage	Subordinate real host percentage	Std Dev	Number of hosts
H1-0	100		0	1
H1-1	83.33	16.67	47.14	2
H1-2	57.14	42.86	10.1	2
H1-3	100		0	1
H1-4	100		0	1
H1-5	80	20	42.43	2
H1-6	100		0	1
H1-7	100		0	1
H1-8	83.33	16.67	47.14	2

Fig. 4: Bidirectional validation of clustering accuracy.

We conducted a series of experiments using different values of K and analyzed the resulting clustering performance. The most accurate results were observed at $K = 9$, as depicted in Figure 4. Showing strong alignment between the real and assigned cluster hostnames. In the forward mapping, we observed that most clustered hosts mapped exclusively to a single real host, achieving 100% accuracy with zero standard deviation. This outcome indicates excellent cluster purity. In other cases, although some hostnames were split across multiple clusters, the split was often limited and containing high standard deviation combined with a low number of clusters per host suggests that each host is primarily split across a few clusters in a dominant-subordinate pattern (e.g., 84%-16%). This pattern still suggests effective clustering, as the majority of instances from a single host were fully linked to a single cluster. The reverse mapping reinforced these findings. Several real hostnames were found to align perfectly with individual clusters, showing strong one-to-one correspondence. Furthermore, clusters with high standard deviation but few hostnames typically indicated that those clusters were dominated by one host, again pointing to well-separated groupings.

While a small number of misclassifications did occur, these were minimal relative to the overall dataset size. Furthermore, the high standard deviation in the affected clusters suggests that most errors were concentrated within a small subset of hosts. Overall, the results confirm that by relying solely on timing data and without any access to real hostnames, we can infer functional relationships and communication structures within the system.

Clustering performs well on medium-sized networks, but for large networks it poses computational and memory challenges, meaning that attacks would necessitate more targeted approaches.

4 Conclusion

Network structures remain vulnerable to privacy breaches even when hostnames are obfuscated, as attackers can exploit timestamp data to infer connections. Our study shows that applying K-means clustering to timestamp data can effectively reconstruct these hidden structures, thus exposing critical limitations in existing metadata protection strategies and underscoring the need to address timestamp privacy. Developing advanced timestamp obfuscation techniques that maintain system functionality while mitigating inference risks will be key to strengthening the security and resilience of distributed systems and we will explore techniques that alter timestamps appropriately.

5 Acknowledgments

This work was conducted as part of the CLEVER project, EU Grant Number 101097560 and EI No: IR-2022-0065, and supported in part by a research grant from Science Foundation Ireland (SFI), Grant Number 13/RC/2077 P2.

References

1. Dell Technologies: Data confidence drives better decisions. `https://www.delltechnologies.com/asset/en-us/solutions/business-solutions/industry-market/data-confidence-drives-better-decisions.pdf`, Accessed 1 May 2025
2. GitHub - Project-Alvarium: Repository containing work related to the forthcoming java sdk implementation. `https://github.com/project-alvarium/alvarium-sdk-java`, Accessed 1 May 2025
3. Herrmann, D., Maaß, M., Federrath, H.: Evaluating the security of a DNS query obfuscation scheme for private web surfing. In: ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference. pp. 205–219. Springer (2014)
4. Hou, T., Wang, T., Lu, Z., Liu, Y.: Combating adversarial network topology inference by proactive topology obfuscation. IEEE/ACM Transactions on Networking **29**(6), 2779–2792 (2021)
5. Humaira, H., Rasyidah, R.: Determining the appropiate cluster number using elbow method for k-means algorithm. In: Proc. of 2nd Workshop on Multidisciplinary and Applications. pp. 1–8 (2020)
6. Khalid, A., Óg Murphy, S., Sreenan, C.J., Roedig, U.: Obfuscating network structure from blockchain analysis. In: Barolli, L. (ed.) Advanced Information Networking and Applications. pp. 95–104. Springer Nature Switzerland, Cham (2025)
7. Khandkar, V.S., Hanawal, M.K.: Masking host identity on internet: Encrypted TLS/SSL handshake. arXiv preprint arXiv:2101.04556 (2021)
8. Lovmar, L., Ahlford, A., Jonsson, M., Syvänen, A.C.: Silhouette scores for assessment of SNP genotype clusters. BMC genomics **6**, 1–6 (2005)
9. Luna-Romera, J.M., García-Gutiérrez, J., Martínez-Ballesteros, M., Riquelme Santos, J.C.: An approach to validity indices for clustering techniques in big data. Progress in Artificial Intelligence **7**, 81–94 (2018)
10. Masood, R., Vatsalan, D., Ikram, M., Kaafar, M.A.: Incognito: A method for obfuscating web data. In: Proceedings of the 2018 world wide web conference. pp. 267–276 (2018)
11. Wu, J.: Cluster analysis and k-means clustering: an introduction. In: Advances in K-Means clustering: A data mining thinking, pp. 1–16. Springer (2012)
12. Xiao, J., Lu, J., Li, X.: Davies bouldin index based hierarchical initialization k-means. Intelligent Data Analysis **21**(6), 1327–1338 (2017)