



3-25-2017

# Detecting Poisoning Attacks on Hierarchical Malware Classification Systems

Dan P. Guralnik

*University of Pennsylvania, guraldan@seas.upenn.edu*

Bill Moran

*RMIT, Melbourne, Australia, bill.moran@rmit.edu.au*


Ali Pezeshki

*Colorado State University - Fort Collins, pezeshki@engr.colostate.edu*

Omur Arslan

*University of Pennsylvania, arslan@seas.upenn.edu*

Follow this and additional works at: [http://repository.upenn.edu/ease\\_papers](http://repository.upenn.edu/ease_papers)

 Part of the [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

## Recommended Citation

Dan P. Guralnik, Bill Moran, Ali Pezeshki, and Omur Arslan, "Detecting Poisoning Attacks on Hierarchical Malware Classification Systems", *SPIE Proceedings* 10185. March 2017. <http://dx.doi.org/doi:10.1117/12.2266556>

This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/ease\\_papers/783](http://repository.upenn.edu/ease_papers/783)

For more information, please contact [repository@pobox.upenn.edu](mailto:repository@pobox.upenn.edu).

---

# Detecting Poisoning Attacks on Hierarchical Malware Classification Systems

## **Abstract**

Anti-virus software based on unsupervised hierarchical clustering (HC) of malware samples has been shown to be vulnerable to poisoning attacks. In this kind of attack, a malicious player degrades anti-virus performance by submitting to the database samples specifically designed to collapse the classification hierarchy utilized by the anti-virus (and constructed through HC) or otherwise deform it in a way that would render it useless. Though each poisoning attack needs to be tailored to the particular HC scheme deployed, existing research seems to indicate that no particular HC method by itself is immune. We present results on applying a new notion of entropy for combinatorial dendrograms to the problem of controlling the influx of samples into the data base and deflecting poisoning attacks. In a nutshell, effective and tractable measures of change in hierarchy complexity are derived from the above, enabling on-the-fly flagging and rejection of potentially damaging samples. The information-theoretic underpinnings of these measures ensure their indifference to which particular poisoning algorithm is being used by the attacker, rendering them particularly attractive in this setting.

## **Keywords**

poisoning attack, hierarchical clustering, hierarchical entropy measure

## **Disciplines**

Electrical and Computer Engineering | Engineering | Systems Engineering

# Detecting Poisoning Attacks on Hierarchical Malware Classification Systems

Dan P. Guralnik<sup>a,\*</sup>, Bill Moran<sup>b</sup>, Ali Pezeshki<sup>c</sup>, and Omur Arslan<sup>a</sup>

<sup>a</sup>University of Pennsylvania, Kodlab, Electrical & Systems Engineering, 200 South 33rd Street, Moore Bldg #203, Philadelphia, PA 19104-6314, USA

<sup>b</sup>MIT University, Electrical & Computer Engineering, 376 Swanston Street, Melbourne VIC 3000, Australia

<sup>c</sup>Colorado State University, Electrical and Computer Engineering, 1373 Campus Delivery, Fort Collins, CO 80523-1373

## ABSTRACT

Anti-virus software based on unsupervised hierarchical clustering (HC) of malware samples has been shown to be vulnerable to poisoning attacks. In this kind of attack, a malicious player degrades anti-virus performance by submitting to the database samples specifically designed to collapse the classification hierarchy utilized by the anti-virus (and constructed through HC) or otherwise deform it in a way that would render it useless. Though each poisoning attack needs to be tailored to the particular HC scheme deployed, existing research seems to indicate that no particular HC method by itself is immune. We present results on applying a new notion of entropy for combinatorial dendrograms to the problem of controlling the influx of samples into the data base and deflecting poisoning attacks. In a nutshell, effective and tractable measures of change in hierarchy complexity are derived from the above, enabling on-the-fly flagging and rejection of potentially damaging samples. The information-theoretic underpinnings of these measures ensure their indifference to which particular poisoning algorithm is being used by the attacker, rendering them particularly attractive in this setting.

**Keywords:** poisoning attack, hierarchical clustering, hierarchical entropy measure

## 1. INTRODUCTION

Rapid detection of cyber-attacks such as viruses, worms, distributed denial-of-service (DDoS), botnets, is a critical aspect of cyber-security, and many implemented and potential algorithms are available for this purpose. A key underpinning technology in many advanced cyber-attack detection algorithms is some form of hierarchical clustering (HC) method. Such methods appear in detecting malware, viruses, and worms,<sup>1-3</sup> identifying compromised domains in DNS traffic,<sup>4</sup> classifying sources and methods of attacks,<sup>5-7</sup> detecting DDoS and privacy attacks,<sup>8-10</sup> identifying hacker and cyber criminal communities,<sup>11</sup> detecting repackaged code in applications for mobile devices,<sup>12</sup> and clustering of files.<sup>13</sup> Many other examples demonstrate the fundamental importance of HC to cyber-security; see for instance Refs. 4, 14, 15.

Given the central role of HC in cyber-security it is critical to understand and design around the vulnerabilities of hierarchical clustering methods. An interesting set of articles by Biggio *et al.*<sup>4,16,17</sup> highlights a major vulnerability in HC: sensitivity to *poisoning* attacks. Biggio *et al.*<sup>4</sup> emphasizes the centrality of clustering of malware families in the identification of common characteristics and the design of suitable countermeasures. As they point out, common approaches to clustering, including the single-linkage hierarchical clustering (SLHC) method,<sup>18</sup> are then susceptible to “poisoning” of the input data to the clustering algorithm, which leads to a flawed classification and creates opportunities for disguising an attack. They provide a method for such a poisoning attack on a behavioral clustering algorithm based on “bridges” to link clusters that would otherwise be distinct.<sup>4</sup> Their analysis provides convincing poisoning schemes against other existing algorithms.<sup>16,17</sup>

In this paper, we test the hypothesis that, by employing natural entropy based diversity measures developed here, one could counter poisoning attacks against the SLHC method using a fairly simple reactive control

---

\* Send correspondence to Dan Guralnik, [guraldan@seas.upenn.edu](mailto:guraldan@seas.upenn.edu)

mechanism based on allowing only small variations in the above measures for each time step. The idea is that large variations of the measure may only occur as a result of very coarse-grained alterations in the underlying hierarchy.

Our work extends the notion of Shannon entropy<sup>19</sup> from the domain of partitions to the domain of hierarchies. By viewing clustering of a data set as a lossy encoding of the data, we develop a quantitative measure of diversity (or information) for a hierarchy. Abrupt falls in the amount of hierarchical information content encoded by a previously effective database of malware samples will serve as a good indicator that recent entries may have been “poison pills”.

While this scheme suffices for demonstrating a capability for withstanding hypothetical poisoning attacks of the kind simulated by Biggio *et al.*, there is no doubt that more intricate approaches are required for more realistic settings (e.g. in the presence of compression), as well as for the more meaningful ones (e.g. in the presence of a learning algorithm which varies the dissimilarity mapping on the feature space of samples — a central design component for determining the clustering hierarchy of the database — as a function of the input). These are set aside as topics for future research.

## 2. PRELIMINARIES

This section reviews some of the notions and results from Ref. 20 required for discussing hierarchies, as well as for motivating and constructing the “discrete entropy” measure we propose for combinatorial hierarchies. We retain the notations of Ref. 20. Where proofs of statements are appropriate, they are provided in the Appendix.

### 2.1 Weights, metrics

Let  $X$  be a finite non-empty set. By  $\mathfrak{Wgt}_X$  we denote the set of all non-negative real-valued and symmetric functions  $w : X \times X \rightarrow \mathbb{R}$  satisfying  $w_{xx} = 0$  for all  $x \in X$ , and henceforth refer to them simply as *weights* on  $X$ . We will usually denote  $w(x, y)$  by  $w_{xy}$ . When thinking of  $e = xy$  as an undirected edge in the complete graph on  $X$  we will also use the symbol  $w_e$  to denote the value  $w(x, y)$ . Two sub-spaces of  $\mathfrak{Wgt}_X$  are of special interest in this paper:  $\mathfrak{Met}_X$ , the space of all weights  $w$  satisfying the triangle inequality (†); and  $\mathfrak{Ult}_X$ , the sub-space of all weights satisfying the ultra-metric inequality (‡):

$$(\dagger) \quad \forall_{x,y,z \in X} w_{xz} \leq w_{xy} + w_{yz}, \quad (\ddagger) \quad \forall_{x,y,z \in X} w_{xz} \leq \max\{w_{xy}, w_{yz}\}$$

Despite prevailing conventions we will refer to elements of  $\mathfrak{Met}_X$  and  $\mathfrak{Ult}_X$  as “metrics” and “ultra-metrics”, respectively\*. Also, all operations and comparisons among weights are to be understood as pointwise operations and comparisons, e.g., we write  $u \leq v$  for  $u, v \in \mathfrak{Wgt}_X$  to mean  $u_{xy} \leq v_{xy}$  for all  $x, y \in X$ , and analogously for operations such as sums, products, and max.

### 2.2 Hierarchies and Single-Linkage Hierarchical Clustering (SLHC)

In the Introduction we reviewed the many motivations for restricting attention to distance-based clustering in the context of unsupervised hierarchical classification schemes, and singled out the distinguished role of single-linkage hierarchical clustering (SLHC) in this context. Now we establish the relevant language, to be used throughout this work.

#### 2.2.1 Hierarchies and Ultra-Metrics

A well-established equivalence between phylogenetic hierarchies (also known as *dendrograms*) and ultra-metrics has been known for decades.<sup>21</sup> Namely, given an ultra-metric  $u \in \mathfrak{Ult}_X$  and any  $\varepsilon \geq 0$ , a partition  $[u]_\varepsilon$  of  $X$  is defined as the quotient space of the relation  $x \sim y \iff u_{xy} \leq \varepsilon$ . This relation is an equivalence relation because  $u$  is an ultra-metric. The resulting family of partitions  $([u]_\varepsilon)_{\varepsilon > 0}$ , also known as the *dendrogram associated with*

---

\*It is unnecessary in the context of this paper to enforce the convention that metrics (resp. ultra-metrics) also satisfy  $w_{xy} > 0$  whenever  $x \neq y$ . Under this convention our metrics would qualify only as “quasi-metrics” — or “semi-metrics” for some — but we find it easier to use a shorter, more inclusive nomenclature.

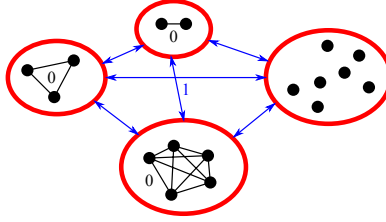


Figure 1. Red circles represent blocks of a partition  $P$ , represented in terms of distances according to (1).

$u$ , has the property that  $[u]_\varepsilon$  refines  $[u]_\delta$  whenever  $\varepsilon \leq \delta$ . By this we mean that every block  $B \in [u]_\varepsilon$  is contained in a block of  $[u]_\delta$ .

EXAMPLE 2.1 (PARTITIONS AS ULTRA-METRICS). Let  $P$  be a partition of  $X$ . For any  $x \in X$  denote the block of  $P$  containing  $x$  by  $P(x)$ . Define:

$$\eta^P = \begin{cases} 0 & \text{if } P(x) = P(y), \\ 1 & \text{if } P(x) \neq P(y). \end{cases} \quad (1)$$

Then  $\eta^P$  is an ultra-metric, called the partition ultra-metric associated with  $P$  — see Figure 1. It is easy to see that  $[\eta^P]_\varepsilon = P$  for all  $\varepsilon \in [0, 1)$ , and becomes the trivial partition  $\{X\}$  for all  $\varepsilon \in [1, \infty)$ .

Since  $X$  is finite, the family  $([u]_\varepsilon)_{\varepsilon \geq 0}$  only has finitely many distinct elements. In fact, no more than  $|X|$ : starting with  $[u]_0$ , blocks of the partition  $[u]_\varepsilon$  keep fusing together as we increase  $\varepsilon$ , the process ending for  $\varepsilon \geq \text{diam}(X, u)$ , where  $[u]_\varepsilon$  becomes a single block. The list of *distinct* partitions obtained in this way, written in order of reverse refinement, will be referred to as the *combinatorial structure of  $u$* , denoted  $\mathfrak{S}(u)$ . The coarsest non-trivial partition in  $\mathfrak{S}(u)$  will be referred to as the *top split associated with  $u$* . The following lemma summarizes some well-known and useful properties — also see Figure 2:

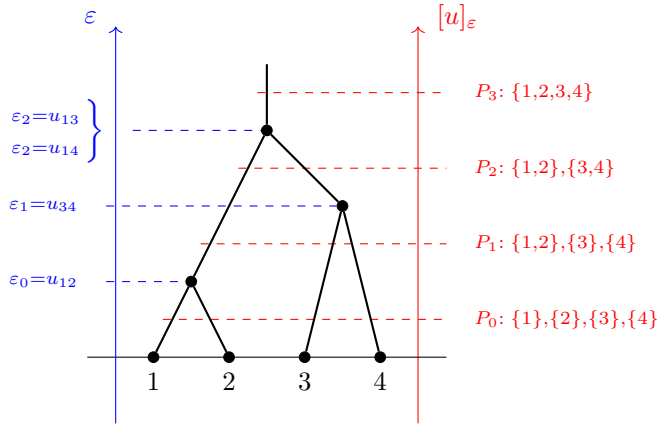


Figure 2. The dendrogram representation of an ultra-metric  $u$  on the space  $X = \{1, 2, 3, 4\}$ . The distance  $u_{xy}$  is recovered as the least height  $\varepsilon$  (left, blue) for which  $x, y$  occur in the same block of the partition  $[u]_\varepsilon$  (right, red), giving rise to the presentation of  $u$  in the form (2).

LEMMA 2.2. Let  $u \in \mathfrak{Utt}_X$  have structure  $\mathfrak{S}(u) : P_0, P_1, \dots, P_k$ . Let  $\varepsilon_{i-1}$  denote the least value of  $\varepsilon$  for which  $P_i$  refines  $[u]_\varepsilon$ . Then: (1)  $k \leq |X| - 1$ ; (2)  $\varepsilon_0 < \varepsilon_1 < \dots < \varepsilon_{k-1}$ ; and (3)  $u$  may be rewritten in the form:

$$u = \max_{i=0, \dots, k-1} \varepsilon_i \eta^{P_i} \quad (2)$$

In particular, any  $u \in \mathfrak{Utt}_X$  takes at most  $|X| - 1$  non-zero values. Those are  $\varepsilon_0 < \dots < \varepsilon_{k-1}$ .

The set of all possible structures (ranging over all possible  $u \in \mathfrak{Ult}_X$ ) will be referred to as the set of *combinatorial dendrograms* or *hierarchies* over  $X$ .

### 2.2.2 Geometry of the Space of Metrics, and Spanning Trees

Let  $K_X$  denote the complete graph on the vertex set  $X$ . By a spanning tree  $T \subset E(K_X)$  of  $X$  we mean a collection of edges of  $K_X$  which, together with the vertex set  $X$ , forms a tree. We denote the set of all spanning trees by  $\mathbf{Trees}(X)$ . Then, viewed as subsets of the vector space of all real-valued maps  $E(K_X) \rightarrow \mathbb{R}$ , the spaces  $\mathfrak{Wgt}_X$  and  $\mathfrak{Met}_X$  are both pointed closed convex cones with vertex at the origin, where  $\mathfrak{Wgt}_X$  coincides with the non-negative orthant and  $\mathfrak{Met}_X$  is a sub-cone of it. Both have a natural decomposition as the union of pairwise interiorly-disjoint simplicial cones of the form  $W(T)$  (denoted  $C(T)$  for  $\mathfrak{Met}_X$ ), where: (1)  $T \in \mathbf{Trees}(X)$ , and (2)  $w \in W(T)$  (repectively  $w \in C(T)$ ) if and only if  $T$  is a minimal spanning tree (MST) for the weight  $w$  (resp. metric) — see Lemmas 2.4 and 2.5 in Ref. 20. The collection of minimal spanning trees associated with a particular weight/metric  $w$  is denoted by  $\mathbf{MST}(w)$ .

**EXAMPLE 2.3 (MSTs FOR A PARTITION).** *Let  $P$  be a partition of  $X$ . It would be sensible to guess that a minimal spanning tree for  $\eta^P$  is the union of a spanning forest  $F$  of total weight zero with respect to  $\eta^P$  with a set  $G$  of  $|P| - 1$  edges satisfying the requirement that the image of  $G$  under the contraction mapping  $K_X \rightarrow K_P$  induced by the natural quotient mapping  $X \rightarrow P$  is any spanning tree of  $K_P$  (see Figure 3). We shall see presently that a similar description holds for all ultra-metrics.*

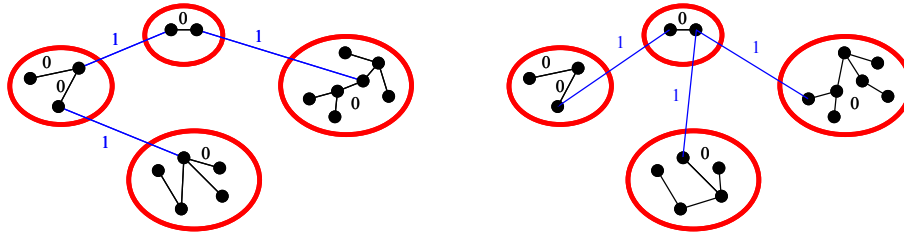


Figure 3. Constructing an MST for the partition ultra-metric from Figure 1. Black edges represent length 0 edges belonging to ‘local’ MSTs of the individual blocks, while blue edges represent edges extending the forest of black edges to a MST of the whole space  $X$ , as described in Example 2.3.

A key property of the cone decomposition is that it respects the single linkage clustering map, in the following sense. Recall that the single linkage clustering map sends a weight/metric  $w$  to an ultra-metric  $\mathfrak{sl}(w)$  having the following two alternative definitions:

$$\mathfrak{sl}(w) = \sup\{u \in \mathfrak{Ult}_X \mid u \leq w\} \quad (3)$$

$$\mathfrak{sl}(w)_{xy} = \max\{w_e \mid e \text{ separates } x \text{ from } y \text{ in } T\} \quad (4)$$

which results in every MST of  $w$  being an MST of  $\mathfrak{sl}(w)$  (Ref. 20, Proposition A.1). Moreover, an explicit formula for the pre-image of an ultra-metric  $u \in \mathfrak{Ult}_X$  under  $\mathfrak{sl}(\cdot)$  is obtained (Ref. 20, Lemma 2.6 and Equations 5,9):

$$\mathfrak{sl}^{-1}(u) = \bigcup_{T \in \mathbf{MST}(u)} C(T, u), \quad C(T, u) = \{w \in \mathfrak{Met}_X \mid u \leq w \leq \omega(T^u)\}, \quad (5)$$

where  $\omega(T^u)_{xy}$  is the cumulative distance from  $x$  to  $y$  along the tree  $T$  as defined by the weights incurred from  $u$ . In other words,  $w \in C(T, u)$  if and only if  $w_{xy}$  lies between the maximum of the weights on the edges between  $x$  and  $y$  on  $T$  and the sum of those weights, for all  $x, y \in X$ .

In this way, we see that pre-image under SLHC of a fixed ultra-metric  $u$  is, essentially<sup>†</sup>, the disjoint union of compact polytopes indexed by the collection  $\mathbf{MST}(u)$  of minimal spanning trees associated with  $u$ . Moreover, the following lemma demonstrates  $\mathbf{MST}(u)$  is an invariant of the combinatorial hierarchy  $\mathfrak{S}(u)$  determined by  $u$ . This

<sup>†</sup>That is, up to overlaps of Lebesgue measure zero

motivates our attempt, realized in the next section, of considering the cardinality of  $\text{MST}(u)$  as a surrogate for an entropy measure associated with  $\mathfrak{S}(u)$ , the number of minimal spanning trees supported by  $\mathfrak{S}(u)$  representing the “degree of uncertainty” regarding what metric may have produced  $\mathfrak{S}(u)$  through SLHC clustering.

LEMMA 2.4. *Let  $u$  be an ultra-metric. The collection of spanning trees  $T$  such that  $C(T)$  intersects  $\mathfrak{sl}^{-1}(u)$  in a set of positive Lebesgue measure depends only on the combinatorial structure of  $u$ .*

This last observation merits a more careful study of the relationship between  $\text{MST}(u)$  and  $\mathfrak{S}(u)$  for an ultra-metric  $u$ .

DEFINITION 2.5. *Let  $T \in \text{Trees}(X)$  and  $P$  be a partition of  $X$ . We say that  $P$  is compatible with  $T$  if  $P$  is the partition induced by the connected components of a spanning sub-forest of  $T$ .*

LEMMA 2.6 (COMPATIBILITY LEMMA). *Let  $w \in \mathfrak{Met}_X$ . Then the partition  $[w]_\varepsilon$  is compatible with every  $T \in \text{MST}(\mathfrak{sl}(w))$ , for every  $\varepsilon \geq 0$ .*

The proof of this lemma amounts to a restatement of a well-known algorithm for computing  $\mathfrak{sl}(w)$  out of a minimum spanning tree for  $w$ . Our goal in presenting it, however, is to reverse the procedure and turn it into a tool for computing  $\text{MST}(u)$  out of an ultra-metric  $u$ . We close the preliminaries section by stating the following well-known fact as a corollary of the last lemma:

COROLLARY 2.7. *Let  $u \in \mathfrak{Ultr}_X$  with structure  $\mathfrak{S}(U) : P_0, \dots, P_k$ . Then, for any  $T \in \text{MST}(u)$  and  $t = 0, \dots, k - 1$  the partition  $P_t$  coincides with the partition of  $X$  into connected components of the forest obtained from  $T$  by removing all the edges of the  $k - t$  highest lengths. In particular, the top split of  $u$  is obtained by removing from  $T$  all the edges of length  $\text{diam}(X, u)$ .*

### 3. AN ENTROPY FOR DISCRETE HIERARCHIES

The decomposition of the space of metrics into the cones  $C(T)$  ( $T \in \text{Trees}(X)$ ) hints at the possibility of viewing  $\text{Trees}(X)$  as a very coarse discretization of  $\mathfrak{Met}_X$ . The matching decomposition of  $\mathfrak{sl}^{-1}(u)$  given by (5) then supports the point of view that  $\mathfrak{S}(u)$  should be seen as defining a ‘macro-state’ in a space of ‘micro-states’ represented by all possible spanning trees of  $K_X$ . By analogy with Boltzmann’s definition of entropy as the number of micro-states consistent with a given macro-state, then we define entropy as follows.

DEFINITION 3.1. *Let  $u \in \mathfrak{Ultr}_X$ . The discrete entropy of  $u$  is the quantity*

$$DH(u) := -\frac{1}{|X|} \log \frac{|\text{MST}(u)|}{|\text{Trees}(X)|}. \quad (6)$$

The number of spanning trees of a complete graph on  $n$  vertices is, by Cayley’s Formula,  $n^{n-2}$ . Assuming the correctness of the assertion in Example 2.3, it is then clear that  $|\text{MST}(\eta^P)| = |P|^{|P|-2} \prod_{B \in P} |B|^{|B|-2}$ , whereas  $|\text{Trees}(X)|$  is  $|X|^{|X|-2}$ . From these, we see that

$$DH(\eta^P) = -\frac{1}{|X|} ((|P| - 2) \log |P| + \sum_{B \in P} (|B| - 2) \log |B| - (|X| - 2) \log |X|).$$

Extending this simple calculation inductively allows us to obtain a formula for general ultra-metrics.

#### 3.1 Counting Minimal Spanning Trees

Lemma 2.4 demonstrates that  $\mathfrak{S}(u)$ , for  $u \in \mathfrak{Ultr}_X$ , completely determines  $\text{MST}(u)$ , yet a more detailed description of  $\text{MST}(u)$  in terms of  $\mathfrak{S}(u)$  is possible.

LEMMA 3.2. *Let  $u \in \mathfrak{Ultr}_X$ ,  $\varepsilon \geq 0$  and  $P = [u]_\varepsilon$ . Let  $\pi : K_X \rightarrow K_P$  denote the natural quotient map sending each  $x \in X$  to its block under  $P$  (denoted  $P(x)$ ), and let  $\bar{u} \in \mathfrak{Ultr}_P$  be defined by:*

$$\bar{u}_{P(x)P(y)} = u_{xy} \eta_{xy}^P = \begin{cases} u_{xy} & \text{if } P(x) \neq P(y) \\ 0 & \text{if } P(x) = P(y) \end{cases} \quad (7)$$

*Then every  $T \in \text{MST}(u)$  is the union of a spanning forest  $F$  and a set of edges  $G$  satisfying –*



1. The connected components of  $F$  induce the partition  $P$  on  $X$ ;
2. For every  $x$ , the connected component  $F(x)$  of  $F$  containing  $x$  satisfies  $F(x) \in \text{MST}(u|_{P(x)})$ .
3. The contraction map  $\pi$  restricts to an injective mapping of  $G$  onto an element of  $\text{MST}(\bar{u})$ .

Conversely, any tree  $T$  of this form lies in  $\text{MST}(u)$ . One quickly notes the following corollaries.

**COROLLARY 3.3.** *If  $u, v \in \mathfrak{Utt}_X$  satisfy  $\mathfrak{S}(u) \subseteq \mathfrak{S}(v)$  then  $|\text{MST}(u)| \geq |\text{MST}(v)|$ .*

**COROLLARY 3.4.** *Let  $u \in \mathfrak{Utt}_X$  and let  $P$  be its top split. Then:*

$$|\text{MST}(u)| = |X|^{|P|-2} \prod_{B \in P} |B| \cdot |\text{MST}(u|_B)| \quad (8)$$

### 3.2 Shannon Entropy and Hierarchical Entropy

We seek a more practical way of computing  $DH(u)$  for ultra-metrics  $u \in \mathfrak{Utt}_X$ . Corollary 3.4 produces the following interesting identity:

**PROPOSITION 3.5.** *For any  $u \in \mathfrak{Utt}_X$ , if  $P$  is the top split of  $u$  then:*

$$DH(u) = - \sum_{B \in P} \frac{|B|-1}{|X|} \log \frac{|B|}{|X|} + \sum_{B \in P} \frac{|B|}{|X|} DH(u|_B) \quad (9)$$

An immediate corollary of this identity is the recovery of Shannon entropy from the discrete hierarchical entropy in the thermodynamical limit:

**COROLLARY 3.6.** *For each  $k \in \mathbb{N}$ , let  $P_k = \{B_1^{(k)}, \dots, B_t^{(k)}\}$  be a partition of a finite space  $X_k$  such that  $\lim_{k \rightarrow \infty} |X_k| = +\infty$  and*

$$(\star) \quad \lim_{k \rightarrow \infty} \frac{|B_i^{(k)}|}{|X_k|} = p_i \in (0, 1)$$

for all  $i = 1, \dots, t$ . Then for  $u_k = \eta^{P_k} \in \mathfrak{Utt}_{X_k}$ , one has

$$\lim_{k \rightarrow \infty} DH(u_k) = H(p_1, \dots, p_t)$$

where  $H(p_1, \dots, p_t) = - \sum_{i=1}^t p_i \log p_i$  is Shannon entropy.

Returning to the general case, we slightly extend the notion of a cluster of a hierarchy used in Ref. 22:

**DEFINITION 3.7.** *Let  $u \in \mathfrak{Utt}_X$ . A subset  $B \subseteq X$  is a cluster of  $u$  if it is a block of some partition belonging to the structure of  $u$ . We denote the set of all clusters of  $u$  by  $\mathcal{C}(u)$ .*

The recursive formula for  $DH(u)$  may be rewritten in closed form as follows:

**COROLLARY 3.8.** *Let  $u \in \mathfrak{Utt}_X$ . For each  $B \in \mathcal{C}(u)$ ,  $B \neq X$ , define  $B^+$  to be the smallest cluster of  $u$  properly containing  $B$ , and set  $X^+ = X$ . Then:*

$$DH(u) = \log |X| + \frac{1}{|X|} \sum_{B \in \mathcal{C}(u)} \log \frac{|B|}{|B^+|} \quad (10)$$

This formula exposes the key feature of discrete entropy: that it measures “rate of expansion” across the hierarchy. The summands on the right are all negative, so that  $DH(u) \leq \log |X|$ . A slightly less elegant but more useful formula for  $DH(u)$  in some situations is the following reformulation of (10).



COROLLARY 3.9. Let  $u \in \mathfrak{Ult}_X$  and let  $0 = \delta_0 < \dots < \delta_m$  be a sequence of real numbers containing all the structure heights of  $u$ . Write  $P_i = [u]_{\delta_i}$  for all  $i = 0, \dots, m$  and, for each  $B \in P_i$ , denote by  $ch(B)$  the set of  $B^\bullet \in P_{i-1}$  contained in  $B$ ; for  $i = 0$  set  $ch(B) = \emptyset$ . Then:

$$DH(u) = \log |X| + \frac{1}{|X|} \sum_{i=1}^m \left( \sum_{B \in P_{i-1}} \log |B| - \sum_{B \in P_i} |ch(B)| \log |B| \right) \quad (11)$$

### 3.3 Functoriality and Extensions

The ‘functoriality’ of SLHC refers to the way SLHC interacts with mappings between datasets, first described by Carlsson and Mémoli.<sup>23</sup> This particular kind of interaction enables our analysis of how  $DH$  should be expected to change as samples are added to the database.

In this paper we are specifically interested in how  $\mathfrak{sl}(\cdot)$  and  $DH$  relate to *extensions*.

LEMMA 3.10. Fix non-empty sets  $X, Y$  with  $X \subset Y$  and a weight  $w$  on  $Y$ . Let  $v \in \mathfrak{Wgt}_X$  be obtained from  $w$  by restricting it to  $X$ , let  $u = \mathfrak{sl}(v)$  and let  $\bar{u}$  be the restriction of  $\mathfrak{sl}(w)$  to  $X$ . Then, for every  $\varepsilon \geq 0$ , every block of the partition  $[u]_\varepsilon$  refines the partition  $[\bar{u}]_\varepsilon$ .

Carlsson and Mémoli<sup>23,24</sup> observe that, in the language of Category Theory, Lemma 3.10 is a special case of what is most compactly expressed by saying that SLHC is a *functor* from the category of weights with non-expansive maps onto the category of dendrograms. However, our use of this property is too restricted to merit a detailed review of categorical notions and the Carlsson-Mémoli theory.

## 4. COUNTERING POISONING ATTACKS

### 4.1 HC-databases

A *dissimilarity-based hierarchical clustering database* is any algorithm maintaining a collection  $X_t$  of samples from a *universe*  $\mathcal{X}$  (we think of  $\mathcal{X}$  as a feature space for the possible samples),  $t$  always a non-negative integer, as follows. The *initial collection*  $X_0$  evolves through a 2-step process enacted at every time  $t$ :

- **Deliberation.** A sample packet  $P_t \subset \mathcal{X}$  is submitted to the database, and is either accepted or rejected;
- **Assimilation.** If  $P_t$  is rejected then  $X_{t+1} = X_t$ , but if it is accepted, a compression procedure may be applied to  $X_t \cup P_t$  to obtain  $X_{t+1}$ .

Both deliberation and assimilation depend in some way on a fixed dissimilarity measure  $\mu : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  that is provided from the outset, as well as on an HC method, which is SLHC in this article. The *clustering hierarchy at  $t$*  provided by the database is then the *dendrogram* encoded by the ultra-metric  $\mathfrak{sl}(X_t, \mu)$ , with  $\mu$  restricted to  $X_t$ , by abuse of notation.

The work by Biggio and his collaborators<sup>4,16,17</sup> assumes a database with no deliberation powers, focusing on the analysis of the effectiveness of specific poisoning attacks. In other words, all packets are assimilated, and poisoning attacks may proceed unhindered — the attacker only needs to tailor her “poison pills” to the particular HC method employed by the database. Below we present an admittedly primitive — but computationally efficient — deliberation procedure to enable the rejection of single element packets  $P_t = \{p_t\}$  causing “too much” damage to the existing database hierarchy. In a nutshell, it is assumed that the database designer has constructed the dissimilarity measure  $\mu$  on the feature space  $\mathcal{X}$  in a way that guarantees a satisfactory coarse-grain classification of the seed data set  $X_0$ ; then, calculating threshold values beyond which differences in diversity — for example,  $\Delta DH_t := DH(HC(X_t, \mu)) - DH(HC(X_t \cup P_t, \mu)|_{X_t})$  — are no longer tolerable will ensure the survival of the highest levels of the original hierarchy throughout the life of the database. We present the details in the next section.

## 4.2 A Simple Deliberation Protocol for SLHC-databases

We consider a setting without compression (that is,  $X_{t+1} = X_t \cup P_t$  for an accepted sample packet), assuming single element packets ( $P_t = \{p_t\}$  for all times  $t$ ), yet we do introduce a non-trivial deliberation component based on the extension property of SLHC discussed above (Section 3.3, Lemma 3.10).

### 4.2.1 Comparing Consecutive Hierarchies

Suppose at time  $t$  we have  $X_t = X$  and an incoming sample  $p_t = y$ . In the notation of Lemma 3.10, we set  $Y = X \cup \{y\}$  and  $w = \mu|_Y$ , the restriction of the database dissimilarity measure  $\mu$  to  $Y$ . The quantity  $\Delta DH_t$  becomes:

$$\Delta DH_t = DH(u) - DH(\bar{u}), \quad (12)$$

with the key property that  $[u]_\varepsilon$  refines  $[\bar{u}]_\varepsilon$  for all  $\varepsilon \geq 0$ .

We apply Corollary 3.9, forming the sequence  $\delta_i$  by splicing together the height sequences of the ultra-metrics  $u$  and  $\bar{u}$ . We have that  $P_i$  refines  $\bar{P}_i$  for all  $i = 0, \dots, m$ , and:

$$\Delta DH_t = \frac{1}{|X|} \sum_{i=1}^m \left( \sum_{B \in P_{i-1}} \log |B| - \sum_{\bar{B} \in \bar{P}_{i-1}} \log |\bar{B}| - \sum_{B \in P_i} |ch(B)| \log |B| + \sum_{\bar{B} \in \bar{P}_i} |ch(\bar{B})| \log |\bar{B}| \right) \quad (13)$$

For any  $i = 0, \dots, m-1$  let  $C_i$  denote the block of  $[w]_{\delta_i}$  which contains  $y$ . Since  $[w]_{\delta_i}$  refines  $[w]_{\delta_{i+1}}$  we have  $C_i \subseteq C_{i+1}$ . The blocks  $C_i$  give rise to blocks  $\bar{B}_i := X \cap C_i = C_i \setminus \{y\}$  of  $\bar{P}_i$ , so that  $\bar{B}_i \subseteq \bar{B}_{i+1}$  for all  $i$ .

We say that  $\bar{B}_i$  *splits*, if it contains more than one block of  $P_i$ . Let  $a$  be the least value of  $i$  for which  $\bar{B}_i$  splits and let  $b$  denote the least value of  $i$  such that no  $\bar{B}_j$ ,  $j \geq b$  splits. Then, for every  $i < a$  and  $i \geq b$  we have  $P_i = \bar{P}_i$ , while for  $a \leq i < b$  and  $B \in P_i$  we have that  $B$  is a block of  $\bar{P}_i$  if and only if  $B \cap \bar{B}_i = \emptyset$ . Continuing the computation of  $\Delta DH$  we obtain:

$$\Delta DH_t = \frac{1}{|X|} \sum_{i=a}^{b-1} \left( -\log |B_i| + \sum_{B \in P_i, B \subset B_i} \log |B| \right) - \frac{1}{|X|} \sum_{i=a}^{b-1} \left( -|ch(B_i)| \log |B_i| + \sum_{B \in P_i, B \subset B_i} |ch(B)| \log |B| \right) \quad (14)$$

$$= \frac{1}{|X|} \sum_{i=a}^{b-1} \left( (|ch(B_i)| - 1) \log |B_i| - \sum_{B \in P_i, B \subset B_i} (|ch(B)| - 1) \log |B| \right) \quad (15)$$

In other words,  $\Delta DH_t$  depends only on the sizes of clusters of  $\bar{u}$  which split, and on the sizes of the clusters of  $u$  into which they split at the same height. Hence, if  $\Delta DH_t$  is non-zero then the hierarchy at time  $t$  and the hierarchy of the projected extension disagree over  $X$ .

### 4.2.2 Comparing Truncated hierarchies

It seems sensible to allow some flexibility for change in the lower (deeper) levels of the hierarchies  $\mathfrak{sl}(X_t, \mu)$ , which means one would like to de-sensitize  $\Delta DH_t$  to the presence of split blocks  $B_i$  arising at height less than a parameter  $\delta \geq 0$ . For this purpose one considers the following operation on ultra-metrics.

**DEFINITION 4.1.** Let  $u \in \mathfrak{Utt}_X$  and  $\delta \geq 0$ . The ultra-metric  $u^\delta$  is defined to be  $u_{xy}^\delta = u_{xy}$  if  $u_{xy} > \delta$  and zero otherwise. We refer to it as the truncation of  $u$  at height  $\delta$ .

One easily verifies that  $u^\delta$  is, indeed, an ultra-metric. Moreover, truncation at height  $\delta$  (for a fixed  $\delta$ ) also enjoys functorial properties similar to those of  $\mathfrak{sl}(\cdot)$ , namely:

**LEMMA 4.2.** If  $u, \bar{u} \in \mathfrak{Utt}_X$  and  $\bar{u} \leq u$  then, for all  $\varepsilon \geq 0$ ,  $[u^\delta]_\varepsilon$  refines  $[\bar{u}^\delta]_\varepsilon$ .

*Proof.* Let  $P = [u]_\delta$  and  $Q = [\bar{u}]_\delta$ . Since  $u_{xy} \leq \delta$  implies  $\bar{u}_{xy} \leq \delta$  for all  $x, y \in X$ , every block of  $P$  is contained in a block of  $Q$ . In other words:  $P$  refines  $Q$ , and hence  $\eta^P \geq \eta^Q$ . Now we make the observation that, for every  $u \in \mathcal{Utt}_X$  and  $\delta \geq 0$ ,  $u^\delta$  may be rewritten as  $u^\delta = u \cdot \eta^P$  where  $P = [u]_\delta$ . Therefore:

$$\bar{u}^\delta = \bar{u} \cdot \eta^Q \leq u \cdot \eta^Q \leq u \cdot \eta^P = u^\delta.$$

Hence, for all  $\varepsilon \geq 0$  and all  $x, y \in X$  we have that  $u_{xy}^\delta \leq \varepsilon$  implies  $\bar{u}_{xy}^\delta \leq \varepsilon$ , or, equivalently, that every block of  $[u^\delta]_\varepsilon$  is contained in a block of  $[\bar{u}^\delta]_\varepsilon$ , as required.  $\square$

Returning to the notation of the preceding section and fixing some  $\delta \geq 0$ , we repeat the argument of the preceding section, *but now applied to  $u^\delta$  and  $\bar{u}^\delta$* , to obtain from (15) the equality:

$$\Delta^\delta DH_t := DH(u^\delta) - DH(\bar{u}^\delta) = \frac{1}{|X|} \sum_{i=a(\delta)}^{b-1} \left( (|ch(B_i)| - 1) \log |B_i| - \sum_{B \in P_i, B \subset B_i} (|ch(B)| - 1) \log |B| \right), \quad (16)$$

where  $a(\delta) = \min\{i \mid \delta_i \geq \delta\}$ .

Thus, considering  $\Delta^\delta DH_t$  instead of  $\Delta DH_t$  allows us to quantify the projected amount of change in the hierarchy of the database  $X_t$  should the sample  $p_t$  be accepted *while disregarding any possible alteration at height less than or equal to  $\delta$* .

It has to be remarked that  $\Delta DH_t$  is not the kind of tool one hopes to employ as part of a proper information-theoretic approach to the problem of quantifying differences between hierarchies: its sign is not constant, and there is much interaction among the summands in (15) and (16). It would be much more appropriate to construct and employ a proper notion of conditional entropy.

## 5. SIMULATIONS

We provide some preliminary numerical evidence supporting our hypothesis that discrete entropy of combinatorial cluster hierarchies can be leveraged for detecting poisoning attacks on (single-linkage) hierarchical clustering.

Our simulations use two different data sets. Both consist of two dimensional data. The measure  $\mu$ , in this case, coinciding with the Euclidean distance in the plane. The first, following Biggio *et al.*,<sup>4</sup> is a ‘‘Banana Dataset’’ produced using the PRtools package<sup>‡</sup>. In this case there is not necessarily a clear prior cluster hierarchy if single-linkage clustering is applied. The data consist of two classes, each sampled from a corresponding banana-shaped distribution, resulting in an instance such as the one depicted in Figure 4(left).

The second is a highly hierarchical data set we have constructed. To do this, we choose,  $N_0$  points uniformly at random in the unit square, and then randomly (uniformly) choose at most  $N_1$  points (the precise number is selected randomly from the  $\{1, 2, \dots, N_1\}$  uniformly) in a disk of radius not exceeding 0.25 around each of those points. These second level points serve as centers for disks of radius not exceeding  $(0.25)^2$  from which we sample at most  $N_2$  points uniformly, and finally these new points serve as centers for disks of radius not exceeding  $(0.25)^4$  from we uniformly sample at most  $N_3$  points (of course, one could continue in the same fashion to arbitrary depths). The disk radius was reduced in size where necessary to prevent data points being placed outside the unit square. An instance of such a data set is shown in Figure 4(right).

For each experiment on any of these data sets, we begin with a hierarchy formed by taking a random selection  $X_0$  of  $F\%$  of the data. This is a key feature of our simulations. We assume that an established structure for the data is in place and that it has a clear hierarchical architecture. Then we serially introduce the remaining data with probability  $1 - p$  or poisoning with probability  $p$ , one sample at a time, until the (possibly poisoned) data set  $X_t$  reaches its original size,  $|X_0| \cdot \frac{100}{F}$ . We will refer to one instance of this process as a *run*.

In this work, because of the complexity of the poisoning algorithm and the limited computational resources committed to this project at this stage, we restricted attention to relatively small data sets. We specify the parameter values for  $F$ ;  $p$ ; the sizes of our data sets; as well as other parameters, in the results section 5.4. Our aim is to detect the poisoning in this situation.

<sup>‡</sup>Source code at <http://37steps.com>.

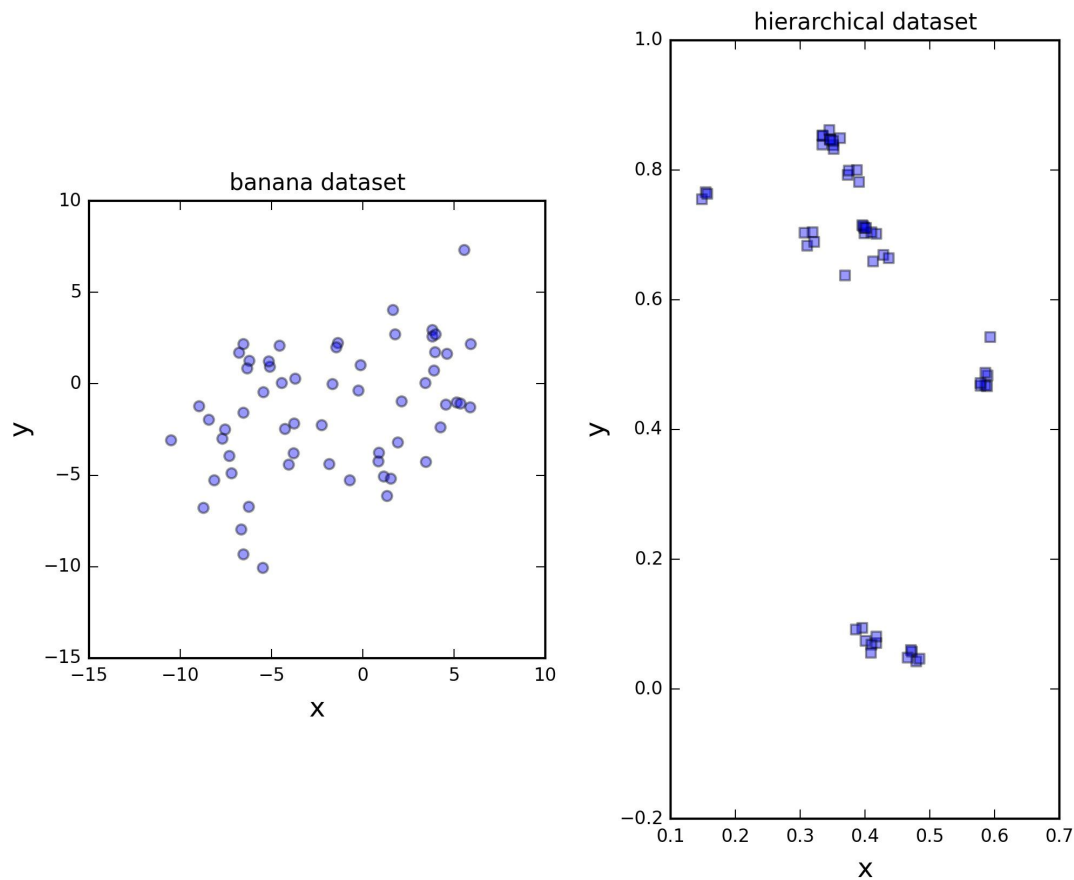


Figure 4. Data sets used for testing detection of poisoning attacks on SLHC. A ‘banana’ dataset from the `PRtools` package (left), with 60 points; and an instance of the ‘hierarchical’ data set designed to have a very strongly expressed SLHC hierarchy (right); here  $N_0 = 3$ ,  $N_1 = 4$ ,  $N_2 = 3$  and  $N_3 = 7$  and the set has 51 points. This plot was generated using the Python `matplotlib`<sup>26</sup> library.

### 5.1 Poisoning Technique

There are many possible poisoning methodologies and we will explore them in future work. Here, to obtain a proof of concept, we focus on just one. This is essentially that of Biggio *et al.*<sup>4</sup> (the so-called “Bridge (Best)” attack) though we have been unable to discern the precise nature of the algorithm they employed. Our attempt to emulate it is described here.

For a partition  $P$  of the data, we let  $Y_P$  be the incidence matrix  $y_{ij} = 1$  if datum  $i$  is in partition element  $P_j$  and 0 otherwise, and  $Y_P Y_P^T$  the *partition adjacency matrix* associated with  $P$ . To compare two partitions  $P$  and  $Q$ , we use the  $L^1$  norm  $\rho(P, Q) = \|Y_P Y_P^T - Y_Q Y_Q^T\|_1$ ; we call this the *adjacency distance*. At this point Biggio *et al.* use the Frobenius norm (that is, the  $L^2$  norm), but the two are equivalent, in any case, and we do not see that this will significantly change the outcomes.

In the notation of Section 4.2.2, at any time  $t$  of a particular run, in order to optimally choose where to insert a poisoning datum (the poison pill  $p_t$ ), we note that insertion of this datum is intended to merge two branches (clusters of  $X_t$ ) at a certain height in the dendrogram. To do this, the datum will be chosen to be at the midpoint between two clusters (in reality). In other words, the collection of potential poison pills (or PPPs) at time  $t$  may be chosen to be the collection of midpoints of edges of a minimum spanning tree of  $X_t$ . We observe that the

effect of adding a point  $x_{PPP}$  from this collection to  $X_t$  can potentially be felt at lower leaves in the hierarchy; in fact, down to a height corresponding to half the distance between those clusters, which we denote by  $h_{PPP}$ .

Let  $u$  be the ultra-metric associated with the healthy data  $X_t$  and let  $u'$  be the restriction to  $X_t$  of the ultra-metric representing the result of clustering the data base after we insert some  $x_{PPP}$ . The adjacency difference between  $[u]_\varepsilon$  and  $[u']_\varepsilon$  is computed for all heights  $\varepsilon$  from height  $h_{PPP}$  to the smaller of the maximum heights of each dendrogram, which has to equal  $\text{diam}(X_t, u')$ . We write

$$g(x_{PPP}) = \min_{\varepsilon \in [h_{PPP}, \text{diam}(X_t, u')]} \rho([u]_\varepsilon, [u']_\varepsilon) \quad (17)$$

for the objective function associated with  $x_{PPP}$ . Now this is maximized over all locations for the PPP to provide the actual poison pill  $p_t$ .

## 5.2 Poisoning Detection

To detect changes induced by poisoning during a run, we propose to compare the values of  $DH$  for the healthy and poisoned data, as discussed in Section 4.2.2, using the measure  $\Delta^\delta DH_t$  defined in Equation (16), which equals the difference in discrete entropies of the dendrograms associated with the data sets  $X_t$  and  $X_{t+1}$ , after their truncation at height  $\delta$ . A non-zero value for this at some height is taken to be a detection of poisoning. The *height* of a detector is the truncation height  $\delta$  at which the entropy is calculated. We are, in effect, running multiple detectors in parallel. In this preliminary work we will not seek to combine these statistics into a single detector nor will be undertake a serious statistical analysis of performance.

## 5.3 Measure of Performance

In measuring performance of the detector (at a given height) we have to take cognizance of the fact that it is impossible to detect a poison pill inserted to connect two clusters at less than half of the height of the detector, though it will be detected with a detector at a lower height. At the same time, a genuine sample could cause the joining of clusters, which, in general motivates raising the height of the detector.

Therefore, from our viewpoint, a success is that a poison attack capable of being detected at a given height is detected or if a genuine datum or poison attack below the height at which detection is possible is not detected (of course the latter is always the case).

For each separate run on a data set, initialized with  $X_0$ , we average detection successes up to time  $t$  as a proportion of total samples added to the data base (that is, divide the total successes by the number of epochs,  $t$ , up that point), for every  $t$ , yielding a vector describing the evolution of the success rate over time. This is repeated over  $N$  runs per each of  $K$  truncation heights  $\delta_k = \varepsilon^k \text{diam}(X_0)$ . Then, for each truncation height, a mean success rate vector (averaged over  $N$  runs) is formed and plotted against time. In this way, at each truncation height, we obtain a performance measure that is a function of time (epochs) since the initial hierarchy is established and poisoning begins.

## 5.4 Results

Our current preliminary experiments were organized as follows.

The ‘banana’ data set was generated to have two (banana-shaped) clusters consisting of 30 points each. The ‘hierarchical’ data set was generated with parameters  $N_0 = 3$ ,  $N_1 = 4$ ,  $N_2 = 3$ ,  $N_3 = 7$  and has 51 points.

The results for Four truncation levels with  $\varepsilon = 0.5 \times 10$  poisoning runs, each initialized with a random selection of 75% of the data set. Each of these regimens was repeated twice for each data set, with poisoning frequencies  $p = 50\%$  and  $p = 25\%$ , and the resulting mean success rates are plotted in Figure 5. As can be observed our approach works very well for the strongly hierarchical data, less well for the Banana Set, though we emphasize that these results are only preliminary, and that a more principled and optimized detector will be designed in future work.

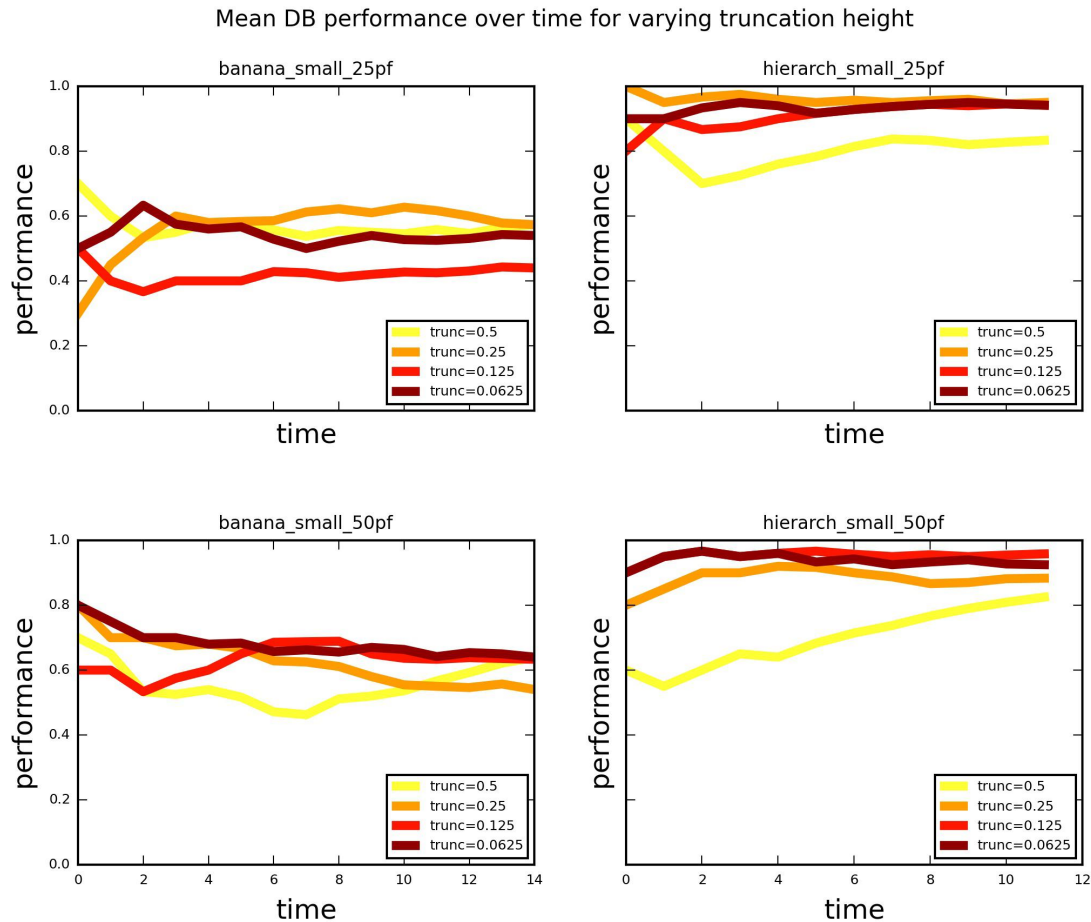


Figure 5. Mean success rate plots for the simulated experiments described in Section 5.4, with the simulation parameters given in Section 5.4. The plots are: ‘Banana’ (left) vs. ‘Hierarchical’ (right); and 25% poisoning frequency (above) vs. 50% poisoning frequency (below). This plot was generated using the Python matplotlib<sup>26</sup> library.

## 6. CONCLUSION

We have developed a concept of entropy (“discrete entropy”) for a hierarchical structure on a data set in a metric space. This measures, in a sense, the amount of uncertainty about a finite metric space given the combinatorial structure of the way it is clustered by SLHC, across scales. We have discussed some of its properties and formulated it in both a local (10) and global (6) way. While the fact that these two distinct interpretations of the entropy of a combinatorial hierarchy coincide is of some independent interest, in this work we stress the low computational complexity of this measure (linear in the number of data points) implied by its local formulation, in conjunction with its very well-organized interaction with the operations of extension (adding a data element) and truncation (contracting clusters to points below a fixed scale).

The low computational complexity of this measure, in tandem with its information-theoretical pedigree, make it an attractive possible alternative to other measures of discrepancy between hierarchies such as those proposed and surveyed in Ref. 22. Those measures were designed for quantifying such discrepancies on a global scale (distances between arbitrary hierarchies are considered) rather than for the (at least) seemingly small-scale transitions such as those exhibited by a meticulously crafted poisoning attack.

It is this constellation of properties, as well as other aspects of the functoriality of SLHC, that motivate this work and its future extensions. In particular, our immediate concern has been to use this idea to measure the

changes in the hierarchical structure of data as a result of poisoning attacks. Preliminary simulations on small data sets suggest that this approach has merit in protecting a single-linkage clustered hierarchy from poisoning of the kind envisaged by Biggio *et al.*<sup>4</sup>

This work is very preliminary and we anticipate further research into combining the multi-height detection process into a single detector, with appropriate thresholding, and with a statistical analysis of the performance of the detector. A comparison with other methods is also required, as are more comprehensive simulations.

## APPENDIX A. PROOFS

### A.1 Proof of Lemma 2.2

*Proof.* We prove the last item. By definition of the  $P_i$ , for all  $x, y \in X$  and all  $i = 1, \dots, k$  we have that  $\eta_{xy}^{P_i} = 0$  iff  $x, y$  are in the same block of  $P_i$ , iff  $x, y$  are in the same block of  $[u]_\varepsilon$  for all  $\varepsilon \geq \varepsilon_{i-1}$ , iff  $u_{xy} \leq \varepsilon_{i-1}$ . Thus,  $u_{xy}$  equals  $\varepsilon_{i-1}$  for the smallest  $i$  that satisfying  $\eta_{xy}^{P_i} = 0$ . Equivalently,  $u_{xy}$  equals  $(\varepsilon_{i-1}\eta^{P_{i-1}})_{xy}$  where  $i$  is smallest such that  $\eta_{xy}^{P_i} = 0$ . For  $j \leq i-1$  we then have  $\eta_{xy}^{P_j} = 1$  and  $\varepsilon_j \leq \varepsilon_{i-1}$  leading to  $(\varepsilon_j\eta^{P_j})_{xy} \leq (\varepsilon_{i-1}\eta^{P_{i-1}})_{xy} = u_{xy}$ , as required.  $\square$

### A.2 Proof of Lemma 2.4

*Proof.* By (5),  $C(T)$  intersects  $\mathfrak{sl}^{-1}(u)$  in a set of positive measure if and only if  $T \in \text{MST}(u)$ . Given  $u, v \in \mathfrak{Utt}_X$ , Lemma 2.2 in Ref. 20 guarantees  $\text{MST}(u) = \text{MST}(v)$  provided  $u_e \leq u_f \iff v_e \leq v_f$  for all edges  $e, f$ . The latter condition holds whenever  $u$  and  $v$  have the same structure, by Equation (2).  $\square$

### A.3 Proof of Lemma 2.6

*Proof.* Set  $u = \mathfrak{sl}(w)$ . Note that  $[u]_\varepsilon = [w]_\varepsilon = P$  take any  $T \in \text{MST}(u)$ . Let  $F$  be the sub-forest of  $T$  obtained by removing those edges  $xy \in T$  satisfying  $u_{xy} > \varepsilon$ . Let  $Q$  be the partition induced from the connected components of  $F$ . Equivalently,  $Q(x) = Q(y)$  if and only if the path  $p(T)_{xy}$  joining  $x$  with  $y$  along  $T$  only contains edges of  $u$ -length not exceeding  $\varepsilon$ . Since  $u$  is an ultra-metric, this implies that  $Q$  refines  $P$ . It remains to show that no block of  $P$  is split by  $Q$ , or, equivalently, that no two points  $x, y \in X$  with  $u_{xy} \leq \varepsilon$  have an edge  $e$  of length exceeding  $\varepsilon$  somewhere along  $p(T)_{xy}$ . However, if such a configuration existed,  $(T - e) + xy$  would have constituted a spanning tree of smaller weight than that of  $T$  – a contradiction to  $T \in \text{MST}(u)$ .  $\square$

### A.4 Proof of Lemma 3.2

*Proof.* First, if  $T \in \text{MST}(u)$  then lemma 2.6 implies  $P$  is compatible with  $T$ , which means that  $T \cap K_B$  is a spanning tree of  $K_B$  for all  $B \in P$ . Letting  $G = T \setminus \bigcup_{B \in P} K_B$  we observe first that no two edges of  $G$  join the same blocks (or else there would be a cycle in  $T$ ), and that  $\pi(G)$  contains no cycles because a cycle in  $\pi(G)$  would lift to a cycle in  $T$ . We conclude that  $T \cap K_B$  is a minimal spanning tree of  $u|_B$  for every  $B \in P$  (else replace an offending  $T \cap K_B$  with a spanning tree of lower weight, reducing the overall weight of  $T$ ). Thus, every  $T \in \text{MST}(u)$  has the required form.

Conversely, suppose  $T \in \text{Trees}(X)$  satisfies 1.-3., and let  $T' \in \text{MST}(u)$  have  $\kappa(T, T')$  as small as possible. From the above observations, both trees are compatible with  $P$ , and  $T \cap K_B = T' \cap K_B$  for all  $B \in P$ . Let  $F = \bigcup_{B \in P} K_B$  and set  $G = T \setminus F$  and  $G' = T' \setminus F$ . Now it is time to use the fact that  $\pi(G), \pi(G') \in \text{MST}(\bar{u})$ : by the definition of  $P$  and  $F$ , all the edges in  $G$  and  $G'$  have lengths strictly greater than  $\varepsilon$ ; this means that the total weight of  $G$  (under  $u$ ) equals the total weight of  $\pi(G)$  (under  $\bar{u}$ ), and hence also equals the total weight of  $\pi(G')$ , which is the total weight of  $G'$ . Thus,  $T$  had been a minimal spanning tree for  $u$  to begin with.  $\square$

### A.5 Proof of Corollary 3.3

*Proof.* Enriching  $\mathfrak{S}(u)$  imposes additional constraints on the set of minimum spanning trees, resulting in  $\text{MST}(v) \subseteq \text{MST}(u)$ .  $\square$



### A.6 Proof of Corollary 3.4

*Proof.* By the preceding lemma, a tree in  $\text{MST}(u)$  is the disjoint union of a choice of tree  $T_B \in \text{MST}(u|_B)$  for each block of  $P$ , with a set of edges  $G \subset K_X$  having the property that  $\pi$  restricts on  $G$  to a bijection with the edge set of a tree  $T \in \text{Trees}(P)$ . There is no restriction on  $T$  because the metric  $\bar{u}$  in our case ( $P$  is the top split of  $u$ ) is a scalar multiple of the standard unit discrete metric on  $P$ . Thus, for each  $T \in \text{Trees}(P)$ , the set  $G$  may be chosen in  $N_T := \prod_{AB \in T} |A||B|$  way, and the total number of ways to pick  $G$  is therefore

$$\sum_{T \in \text{Trees}(P)} N_T = \left( \sum_{B \in P} |B| \right)^{|P|-2} \cdot \prod_{B \in P} |B| = |X|^{|P|-2} \prod_{B \in P} |B|, \quad (18)$$

by the Cayley-Prüfer spanning tree enumerator formula,<sup>25</sup> and (8) follows.  $\square$

### A.7 Proof of Proposition 3.5

*Proof.* We apply corollary 3.4 directly, while observing

$$|X| - 2 = (|P| - 2) + \sum_{B \in P} (|B| - 1) \quad (19)$$

We calculate:

$$DH(u) = -\frac{1}{|X|} \log \frac{|X|^{|P|-2} \prod_{B \in P} |B| \cdot |\text{MST}(u|_B)|}{|X|^{|X|-2}} \quad (20)$$

$$= -\frac{1}{|X|} \log \left( \frac{|X|^{|P|-2}}{|X|^{|X|-2}} \cdot \prod_{B \in P} |B|^{|B|-1} \frac{|\text{MST}(u|_B)|}{|B|^{|B|-2}} \right) \quad (21)$$

$$= -\frac{1}{|X|} \log \prod_{B \in P} \left( \frac{|B|}{|X|} \right)^{|B|-1} \frac{|\text{MST}(u|_B)|}{|B|^{|B|-2}} \quad (22)$$

$$= -\sum_{B \in P} \frac{|B|-1}{|X|} \log \frac{|B|}{|X|} + \sum_{B \in P} \frac{|B|}{|X|} DH(u|_B), \quad (23)$$

as required.  $\square$

### A.8 Proof of Corollary 3.6

*Proof.* Applying proposition 3.5 to  $u_k$  we obtain

$$DH(u_k) = -\sum_{i=1}^t \frac{|B_i^{(k)}| - 1}{|X_k|} \log \frac{|B_i^{(k)}|}{|X_k|} + \sum_{i=1}^t \frac{|B_i^{(k)}|}{|X_k|} DH(\mathbf{0}). \quad (24)$$

Since the zero metric admits any spanning tree as a minimum one,  $DH(\mathbf{0}) = 0$  and we are done.  $\square$

### A.9 Proof of Corollary 3.8

*Proof.* First we may assume  $u$  is positive-definite. Indeed, if  $u$  is not positive definite, then pick  $\varepsilon > 0$  satisfying  $\varepsilon < u_{xy}$  for any  $x, y$  satisfying  $u_{xy} > 0$ . Letting  $P$  be the partition of  $X$  into singletons, it is clear that the structure of the ultra-metric  $v = u + \varepsilon\eta^P$  equals the structure of  $u$  with the addition of  $P$ . In particular, any spanning tree that is compatible with  $u$  is compatible with  $v$  and vice-versa, resulting in  $DH(v) = DH(u)$ .

Next we rewrite the identity from prop. 3.5:

$$DH(u) = H_X(P) + \sum_{B \in P} \frac{|B|}{|X|} DH(u|_B) + \frac{1}{|X|} \sum_{B \in P} \log \frac{|B|}{|X|}, \quad (25)$$

where this time  $P$  is the first split of  $u$  and  $H_X(P)$  is the Shannon entropy of  $P$  with respect to the uniform distribution on  $X$ . Denote

$$\mathfrak{S}(u) : P_0(u) \succ P_1(u) \succ \dots \succ P_{t-1}(u) = P \quad (26)$$

and set  $\mathcal{C}_k = P_{t-1} \cup \dots \cup P_{t-k}$  for  $1 \leq k \leq t$ . We claim the following formula holds for all  $k$ :

$$DH(u) = H_X(P_{t-k}) + \sum_{B \in P_{t-k}} \frac{|B|}{|X|} DH(u|_B) + \frac{1}{|X|} \sum_{B \in \mathcal{C}_k} \log \frac{|B|}{|B^u|} \quad (27)$$

Note that the particular instance of  $k = t$  finishes the proof:  $P_0$  is the partition of  $X$  into (equiprobable) singletons, giving  $H_X(P_0) = \log |X|$  while  $DH(u|_B) = 0$  for all the  $B \in P_0$ , as they are all singletons.

Using induction on  $k$ , and having verified the case  $k = 1$  already, assume (27) holds for  $k = m$  where  $1 \leq m < t$  is integer and consider the case  $k = m + 1$ . Throughout the following computation the notation  $B_j$  refers to blocks in the partition  $P_{t-j}$  ( $j$  may vary):

$$\begin{aligned} DH(u) &= H_X(P_{t-m}) + \frac{1}{|X|} \sum_{B \in \mathcal{C}_m} \log \frac{|B|}{|B^u|} + \sum_{B_m} \frac{|B_m|}{|X|} DH(u|_{B_m}) \\ &= H_X(P_{t-m}) + \frac{1}{|X|} \sum_{B \in \mathcal{C}_m} \log \frac{|B|}{|B^u|} \\ &\quad + \sum_{B_m} \frac{|B_m|}{|X|} \left( - \sum_{B'_{m+1} \subset B_m} \frac{|B'_{m+1}| - 1}{|B_m|} \log \frac{|B'_{m+1}|}{|B_m|} + \sum_{B'_{m+1} \subset B_m} \frac{|B'_{m+1}|}{|B_m|} DH(u|_{B'_{m+1}}) \right) \\ &= H_X(P_{t-m}) + \sum_{B_m} \frac{|B_m|}{|X|} H_{B_m}(P_{t-m-1}|_{B_m}) \\ &\quad + \frac{1}{|X|} \sum_{B \in \mathcal{C}_m} \log \frac{|B|}{|B^u|} + \sum_{B_m} \sum_{B'_{m+1} \subset B_m} \frac{1}{|X|} \log \frac{|B'_{m+1}|}{|B_m|} \\ &\quad + \sum_{B_m} \sum_{B'_{m+1} \subset B_m} \frac{|B'_{m+1}|}{|X|} DH(u|_{B'_{m+1}}) \\ &= H_X(P_{t-m-1}) + \sum_{B \in \mathcal{C}_{m+1}} \log \frac{|B|}{|B^u|} + \sum_{B_{m+1}} \frac{|B_{m+1}|}{|X|} DH(u|_{B_{m+1}}), \end{aligned}$$

concluding the induction.  $\square$

### A.10 Proof of Corollary 3.9

*Proof.* Starting with the representation of  $DH(u)$  given by Corollary 3.8, observe we may rewrite it in the form:

$$DH(u) = \log |X| + \frac{1}{|X|} \sum_{i=0}^{m-1} \sum_{B \in P_i} \log \frac{|B|}{|B^+|},$$

where, for the purpose of this computation only, we redefine  $B^+$  as the unique block of  $P_{i+1}$  containing  $B$ . This

allows for the following re-indexing and consequent development:

$$\begin{aligned}
DH(u) &= \log |X| + \frac{1}{|X|} \sum_{i=1}^m \sum_{B \in P_i} \sum_{B^\bullet \in ch(B)} \log \frac{|B^\bullet|}{|B|} \\
&= \log |X| + \frac{1}{|X|} \sum_{i=1}^m \sum_{B \in P_i} \sum_{B^\bullet \in ch(B)} (\log |B^\bullet| - \log |B|) \\
&= \log |X| + \frac{1}{|X|} \sum_{i=1}^m \left( \sum_{B \in P_{i-1}} \log |B| - \sum_{B \in P_i} |ch(B)| \log |B| \right),
\end{aligned}$$

as desired.  $\square$

### A.11 Proof of Lemma 3.10

*Proof.* Formally, we have a map  $j : X \rightarrow Y$  defined by  $j(x) = x$ , the *inclusion map*. Restricting any weight on  $Y$  to  $X$  is then tantamount to applying to it the map  $j^*$ , defined by  $j^*(w)_{xy} := w_{j(x)j(y)}$ . Thus, for the particular  $v, w$  in our setting we have  $v = j^*(w)$ .

Now let us consider  $\mathfrak{sl}(\cdot)$  applied to this context, namely: we would like to compare the restriction of  $\mathfrak{sl}(w)$  to  $X$  (in other words “apply HC first, then restrict”) with  $\mathfrak{sl}(v)$  (“restrict, then apply HC”).

Now, it is easy to see that  $w_1 \leq w_2$  implies  $j^*(w_1) \leq j^*(w_2)$ . In particular,  $j^*(\mathfrak{sl}(w)) \leq j^*(w) = v$ . By equation (3) we have  $j^*(\mathfrak{sl}(w)) \leq \mathfrak{sl}(v)$  because  $j^*(\mathfrak{sl}(w))$  is an ultra-metric (the restriction of an ultra-metric on  $Y$  to  $X$  is clearly an ultra-metric on  $X$ ). In particular, for any  $\varepsilon \geq 0$  and  $x, y \in X$  we have that  $\mathfrak{sl}(v)_{xy} \leq \varepsilon$  implies  $j^*(\mathfrak{sl}(w))_{xy} \leq \varepsilon$ . This proves the lemma.  $\square$

## ACKNOWLEDGMENTS

Guralnik gratefully acknowledges support by the U.S. Air Force Office of Scientific Research under grant MURI FA9550-10-1-0567. Moran’s contribution to this work was funded by the U. S. Air Force Office of Scientific Research under grant No. FA9550-12-1-0418. Pezeshki wishes to acknowledge support by NSF under grant CCF-1422658. Arslan was supported by AFRL grant FA865015D1845 (subcontract 669737-1).

## REFERENCES

- [1] Wang, J., Miller, D., and Kesidis, G., “Efficient mining of the multidimensional traffic cluster hierarchy for digesting, visualization, and anomaly identification,” *Selected Areas in Communications, IEEE Journal on* **24**, 1929–1941 (Oct 2006).
- [2] Hijazi, A., Inoue, H., Matrawy, A., Van Oorschot, P., and Somayaji, A., “Discovering packet structure through lightweight hierarchical clustering,” in [*Communications, 2008. ICC '08. IEEE International Conference on*], 33–39 (May 2008).
- [3] Newsome, J., Karp, B., and Song, D., “Polygraph: Automatically generating signatures for polymorphic worms,” in [*Proceedings of the 2005 IEEE Symposium on Security and Privacy, SP '05*], 226–241, IEEE Computer Society, Washington, DC, USA (2005).
- [4] Biggio, B., Rieck, K., Ariu, D., Wressnegger, C., Corona, I., Giacinto, G., and Roli, F., “Poisoning behavioral malware clustering,” in [*Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*], 27–36, ACM (2014).
- [5] Kang, J., Zhang, Y., and bin Ju, J., “Classifying ddos attacks by hierarchical clustering based on similarity,” in [*Machine Learning and Cybernetics, 2006 International Conference on*], 2712–2717 (Aug 2006).
- [6] Wei, S., Mirkovic, J., and Kissel, E., “Profiling and clustering internet hosts.,” *DMIN* **6**, 269–75 (2006).
- [7] Patton, R., Beaver, J., Steed, C., Potok, T., and Treadwell, J., “Hierarchical clustering and visualization of aggregate cyber data,” in [*Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*], 1287–1291 (July 2011).

- [8] Karami, A., “Article: Data clustering for anomaly detection in content-centric networks,” *International Journal of Computer Applications* **81**, 1–8 (November 2013). Full text available.
- [9] Du, H. and Yang, S. J., “Discovering collaborative cyber attack patterns using social network analysis,” in [SBP], Salerno, J. J., Yang, S. J., Nau, D. S., and Chai, S.-K., eds., *Lecture Notes in Computer Science* **6589**, 129–136, Springer (2011).
- [10] Gu, G., Perdisci, R., Zhang, J., and Lee, W., “Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection.”
- [11] Lu, Y., Luo, X., Polgar, M., , and Cao, Y., “Social network analysis of a criminal hacker community,” *Journal of Computer Information Systems* **51**(2), 31–41 (2010).
- [12] Hanna, S., Huang, L., Wu, E., Li, S., Chen, C., and Song, D., “Juxtapp: A scalable system for detecting code reuse among android applications,” in [*Proceedings of the 9th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*], DIMVA’12, 62–81, Springer-Verlag, Berlin, Heidelberg (2013).
- [13] Steinbach, M., Karypis, G., and Kumar, V., “A comparison of document clustering techniques,” in [*KDD Workshop on Text Mining*], (2000).
- [14] Kirda, E., “Malware behavior clustering,” in [*Encyclopedia of Cryptography and Security*], van Tilborg, H. and Jajodia, S., eds., 751–752, Springer US (2011).
- [15] Mahmood, A. N., Leckie, C., and Udaya, P., “An efficient clustering scheme to exploit hierarchical data in network traffic analysis,” *Knowledge and Data Engineering, IEEE Transactions on* **20**(6), 752–767 (2008).
- [16] Biggio, B., Pillai, I., Rota Bulò, S., Ariu, D., Pelillo, M., and Roli, F., “Is data clustering in adversarial settings secure?,” in [*Proceedings of the 2013 ACM workshop on Artificial intelligence and security*], 87–98, ACM (2013).
- [17] Biggio, B., Bulò, S. R., Pillai, I., Mura, M., Mequanint, E. Z., Pelillo, M., and Roli, F., “Poisoning complete-linkage hierarchical clustering,” in [*Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*], 42–52, Springer (2014).
- [18] Carlsson, G. and Mémoli, F., “Characterization, stability and convergence of hierarchical clustering methods,” *J. Mach. Learn. Res.* **11**, 1425–1470 (2010).
- [19] Shannon, C. E. and Weaver, W., “The mathematical theory of information,” (1949).
- [20] Zhu, D., Guralnik, D. P., Wang, X., Li, X., and Moran, B., “Statistical properties of single linkage hierarchical clustering,” *Journal of Statistical Planning and Inference* (2017).
- [21] Jardine, N. and Sibson, R., “Mathematical taxonomy,” *London etc.: John Wiley* (1971).
- [22] Arslan, O., Guralnik, D. P., and Koditschek, D. E., “Discriminative measures for comparison of phylogenetic trees,” *Discrete Applied Mathematics* (2016).
- [23] Carlsson, G. and Mémoli, F., “Persistent clustering and a theorem of J. Kleinberg (preprint),” *arXiv:0808.2241 [stat.ML]* (2008).
- [24] Carlsson, G. and Mémoli, F., “Classifying clustering schemes,” *arXiv preprint arXiv:1011.5270* (2010).
- [25] West, D. B. et al., [*Introduction to graph theory*], Prentice hall Upper Saddle River (2001).
- [26] Hunter, J. D., “Matplotlib: A 2d graphics environment,” *Computing In Science & Engineering* **9**(3), 90–95 (2007).