Quantifying Nondeterminism and Inconsistency in Self-organizing Map Implementations

Sydur Rahaman, Raina Samuel, Iulian Neamtiu
Department of Computer Science, *New Jersey Institute of Technology*, Newark, NJ, USA
{sr939, res9, ineamtiu}@njit.edu

***Abstract*—Self-organizing maps (SOMs) are a popular approach for neural network-based unsupervised learning. However the reliability of self-organizing map implementations has not been investigated. Using internal and external metrics, we define and check two basic SOM properties. First, *determinism*: a given SOM implementation should produce the same SOM when run repeatedly on the same training dataset. Second, *consistency*: two SOM implementations should produce similar SOMs when presented with the same training dataset. We check these properties in four popular SOM implementations. We ran our approach on 381 popular datasets used in health, medicine, and other critical domains. We found that implementations violate these basic properties. For example, 375 out of 381 datasets have nondeterministic outcomes; for 51–92% of datasets, toolkits yield significantly different SOM clusterings; and clustering accuracy might be so inconsistent as to vary by a factor of four between toolkits. This undermines SOM reliability, and the reliability of results obtained via SOMs. Our study shines a light on what to expect, in practice, when running actual SOM implementations. Our findings suggest that for critical applications, SOM users should not take reliability for granted; rather, multiple runs and different toolkits should be considered and compared.**

***Index Terms*—Self-organizing maps, neural networks, AI testing, AI reliability, nondeterminism, validation**

I. Introduction

Self-organizing maps (SOMs) are a neural network-based approach for mapping relationships between objects in high-dimensional spaces onto a low-dimension space, usually a neuronal grid [1]. Main uses for SOMs include exploratory data mining [2], dimensionality reduction, clustering, or pre-clustering. Figure 1 shows an example SOM on dataset Zoo, which is used to cluster 101 animals into 7 groups based on 17 characteristics (features). Each circle indicates a neuron, while the clusters, e.g., “Fish” or “Mammal” are indicated via neurons of the same color. Note how animals that have related attributes in the 17-dimensional space are clustered together in the 2-dimensional output SOM.

SOM have been used in critical domains, e.g., finance [3], [4], drug discovery [5], [6], or medical sciences [7]. However, SOM reliability has not been questioned. In this paper we do so, by focusing on two key issues. First, *nondeterminism*: when running an SOM implementation repeatedly on the same dataset yields different results. Second, *inconsistency*: when running two different SOM implementations on the same dataset yields different results.

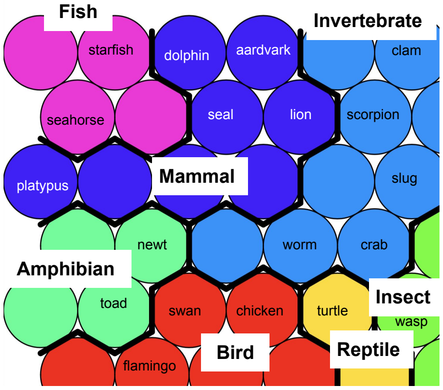


Fig. 1. SOM for dataset Zoo, toolkit RKoh.

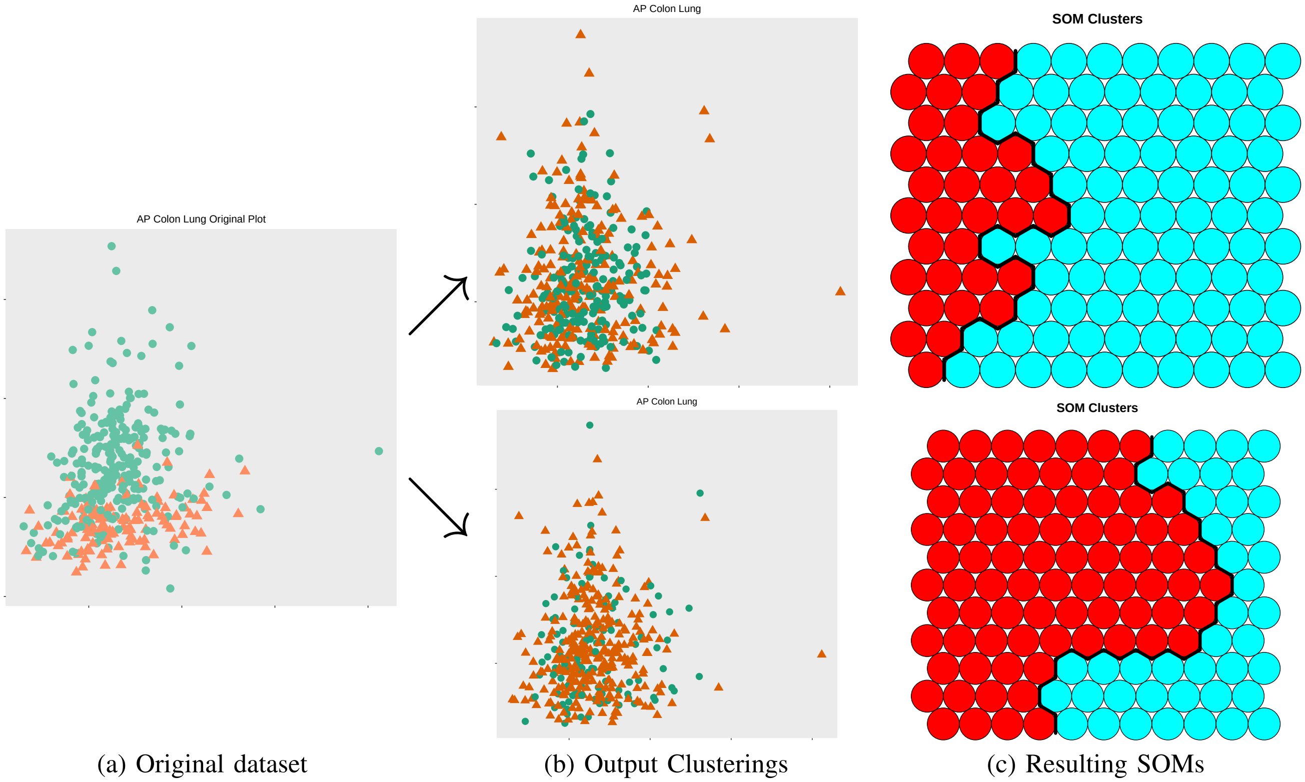
We illustrate nondeterminism in Figure 2, on the AP Colon Lung dataset, from the Gene Expression for Oncology repository.[1] Specifically, we show that two independent runs of the same, simple procedure – training an SOM on AP Colon Lung – can yield two very different results between the two runs. Figure 2(a) shows the original dataset with ground truth (two clusters shown in green circles and orange triangles, respectively). Figure 2(b) shows the SOMs constructed by the R/Kohonen (RKoh) toolkit, on this dataset, in two different runs. Finally, Figure 2(c) shows the resulting SOMs and clusters, for the two runs in the middle; the red and cyan colors indicate the different neurons clusters on the map (separated by the thick black line). Notice the substantial differences in cluster assignments and SOMs between the top and bottom figures; this is due to nondeterminism.

We now illustrate inconsistency: how SOM clustering outcomes (hence accuracy) for the same dataset differ not only across runs, but also across toolkits. We conducted 30 independent runs for each toolkit on the aforementioned Zoo dataset. Figure 3 shows violin plots for clustering accuracy (i.e., distribution of accuracy across the 30 runs). Note that RKoh yields consistently high accuracy (0.78–0.79), whereas for MiniSom, accuracy varies from 0.47 to 0.82 depending on the run. In contrast, TFSom’s accuracy (0.23–0.29) is less than the minimum accuracy for the other toolkits. Hence the choice of toolkit crucially impacts the resulting accuracy.

In the rest of the paper we quantify, via statistical tests on internal/external metrics, how SOMs obtained via training *on the same dataset* differ *across runs* and *across toolkits*.

In Section II we define SOM and discuss the experimental setup: metrics, datasets, toolkits. We investigate four popular

[1] GEMLeR by Stiglic and Kokol [8]: using genetic markers to differentiate between different clinical conditions such as various types of cancer.



(a) Original dataset (b) Output Clusterings (c) Resulting SOMs

Fig. 2. Different SOMs obtained via two different runs in RKoh, dataset AP Colon Lung.

toolkits (SOM packages) – MiniSom, R/Kohonen, TensorFlow SOM, MATLAB – described in Section II-C. We ran our analysis on 381 datasets: about 290 of these were medical datasets, and the rest were benchmarking datasets; a qualitative and quantitative description is provided in Section II-D.

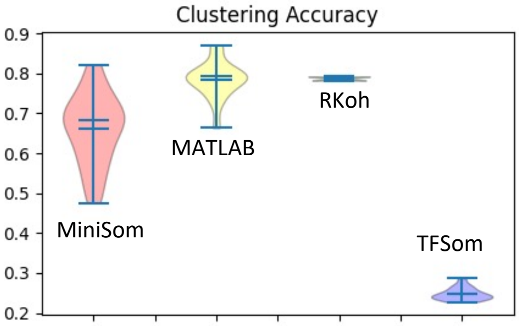


Fig. 3. Clustering accuracy ranges for dataset Zoo.

In Section III we define nondeterminism via a rigorous statistical test. In Sections IV and V we quantify nondeterminism using internal and external metrics. For a given toolkit and dataset, we measure how output SOMs vary across 30 runs. We found that, for our examined 381 datasets, at most 6 lead to deterministic results; hence the vast majority of datasets induce nondeterministic SOM outcomes. In Section VI we define inconsistency via a statistical test, and present our findings: for 51–92% of datasets, toolkits yield SOM clusterings with significantly different accuracy distributions.

II. Definitions and Experimental Setup

We now define the main concepts and describe the setup for our approach.

A. SOM Definition

SOMs are based on unsupervised competitive learning using neural networks. An SOM clusters (maps) high-dimensional data onto a two-dimensional neuron grid. Typically, the grid topology (how neurons are connected) is hexagonal or rectangular. The network "learns" as the grid neurons adapt to the latent structure of the dataset; in other words, SOMs apply competitive learning to adjust weights to neurons. SOMs are useful for managing and visualizing large datasets or high-dimensional datasets, because the datasets are simplified into clusters in the two-dimensional space. As neurons might shift from run to run, SOMs might yield solutions and results that are potentially inconsistent from run to run.

B. SOM performance metrics

Prior research [9], [10] has introduced metrics for SOM performance and the quality of the training algorithm. Forest et al.'s SOMperf package [11] measures SOM quality via internal and external metrics. Internal metrics reflect the "native" quality of the SOM construction and its fit to the input data. In contrast, external metrics measure the implementation based on output labels compared to ground truth, e.g., SOM clusters vs. known clusters. We leverage Forest et al.'s metrics and package to collect input data for our analyses.

TABLE I

NUMBER OF DATASETS WITH STATISTICALLY INVARIANT RUNS; '-' INDICATES THAT ALL DATASETS' RUNS VARIED SIGNIFICANTLY. "MED" AND "TRM" ARE SHORT FORMS OF MEDIAN AND TRIMMED MEAN, RESPECTIVELY.

Toolkit	Quantization Error			Topographic Product			Trust-worthiness			Neighborhood Preservation			Distortion			Kruskal-Shepard Error			Topographic Error		
	Mean	Med	TrM	Mean	Med	TrM	Mean	Med	TrM	Mean	Med	TrM	Mean	Med	TrM	Mean	Med	TrM	Mean	Med	TrM
MiniSom	-	1	-	-	-	-	2	2	2	3	5	4	-	1	-	-	-	-	-	-	-
MATLAB	1	4	1	-	4	-	1	4	1	5	6	5	-	-	-	-	-	-	-	-	-
RKoh	-	-	-	-	3	-	3	4	3	5	6	6	-	4	-	-	6	-	-	-	-
TFSom	-	3	-	-	-	-	-	4	-	-	2	-	-	3	-	-	5	-	-	2	1

C. Toolkits

We investigate four popular[2] SOM packages, as follows. **MiniSom** [15], based on Python/Numpy; **RKoh** – the Kohonen package [16] for R [17]; **MATLAB**'s selforgmap toolbox [18]; and **TFSom** – the TensorFlow Self-Organizing Map package [14] built on top of TensorFlow [19].

D. Datasets

We used 381 datasets from OpenML [20] for MiniSom, RKoh, and MATLAB. For TFSom we only used 361 of these 381 datasets (on 20 datasets, runtime exceeded our imposed 3-hour limit per run). About 290 of these datasets are drawn from the medical domain or bioinformatics, while the rest are specifically designed to evaluate ML implementations. As these datasets are used to benchmark classification approaches we have cluster labels (ground truth). The following table summarizes the characteristics of our datasets: on average, datasets have 219 instances, 16 dimensions, and 2.38 clusters.

	Min	Max	Geometric Mean
Instances	36	2201	219
Features (attributes)	1	61,359	16
K (# of clusters)	2	50	2.38

III. NONDETERMINISM DEFINITION AND TEST

We define *nondeterminism* for a toolkit as follows: constructing SOMs repeatedly using that toolkit, on the *same dataset*, with the *same parameters* leads to *statistically significant variation* in the resulting SOM.

Nondeterminism is fundamentally problematic for several reasons. First, it violates users' expectation that repeated runs have the same outcome, or at least outcomes that are statistically indistinguishable. Second, it leaves SOM users at the mercy of the random number generator, i.e., a "lucky" random seed can lead to a better SOM. Finally, nondeterminism undermines users' confidence in SOM reliability in general. We now define nondeterminism in a statistically rigorous way.

Statistical test for nondeterminism: We use a sensitive statistical measure of nondeterminism that improves over the tests introduced by Yin et al. in the context of clustering nondeterminism [21]: the metric values are nondeterministic if the 30 runs' outcomes have statistically significant variance. Yin et al. used Levene's [22] test set up as follows: the 30 values constitute one group, while the other group has the same mean, size, and no variance (all 30 elements are equal to the mean of the first group). If Levene's test yields a $p < 0.05$, they concluded that the runs vary significantly. The problem with using Levene's test is that it is a mean-based test hence it is most appropriate for symmetric, moderate-tailed distributions. We improve the statistical tests as follows: we run a Levene's mean-based test, as well as Brown-Forsythe's median-based test (good for skewed distributions) and Brown-Forsythe's trimmed mean-based test (good for heavy-tailed distributions) [23]. Of these three, we pick the most sensitive, i.e., the one that finds variance across the largest number of datasets. If the underlying test results in a $p < 0.05$ we conclude that the toolkit is nondeterministic.

IV. NONDETERMINISM RESULTS: INTERNAL METRICS

Internal metrics use the native properties of the SOM model and input dataset in order to evaluate the quality of the SOM implementation on dimensionality reduction. We consider six internal metrics; we discuss their definition, significance, and analysis results shortly.

Parameters: SOM's recommended size is $5 \times \sqrt{N}$ neurons where N is the number of samples in the dataset to analyze [24]. For example, if a dataset has 150 samples, we have $5 \times \sqrt{150} = 5 \times 12.24 = 61.23$. Hence the recommended map has 64 neurons, arranged in an 8-by-8 grid. To keep the SOM map settings consistent across all the tools, we have used a *hexagonal* topology and *Manhattan distance* as the activation distance. We now discuss each metric in turn.

A. Quantization Error

1) Definition: Quantization error applies to clustering algorithms in general. The error is computed from the average Euclidean distance of sample vectors to the centroid, or best matching unit, by which they are represented. A lower quantization error value is desirable.

2) Results: Our nondeterminism hypothesis was confirmed using the three statistical tests. The "Quantization errors" columns of Table I show the number of datasets for which the tests indicate statistical invariance across runs. These numbers are small and consistent across tests (1–4 datasets depending on the toolkit and test). Put otherwise, for the vast majority, 375–381 datasets quantization error *differs significantly across runs*, confirming our nondeterminism hypothesis.

We illustrate one such difference between runs of the dataset ecoli[3] in Figure 4. The figure shows the quality, i.e., mean distance of objects mapped to a neuron to the original data point. For each neuron, quality ranges from dark blue (low error) to green (moderate error) to red (high error). Note the

[2] As indicated by the number of users [12], [13] or GitHub stars [14], [15].

[3] Protein localization sites in bacteria.

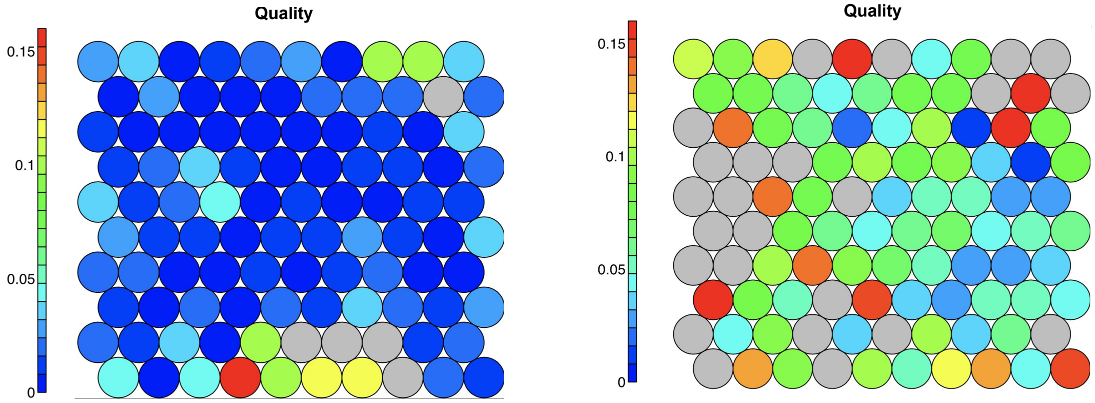


Fig. 4. Quantization error nondeterminism for dataset ecoli, toolkit RKoh. Low error in dark blue, higher error in light blue/green/red. The run with minimum quantization error (59.75) is shown on the left while the run with maximum error (79.07) is shown on the right.

TABLE II
WIDEST-3 DIFFERENCES IN QUANTIZATION ERROR ACROSS RUNS.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	lsvt	5.8E+8	9.7E+8	3.9E+8	9.3E+7
	micro-mass	1.4E+7	1.5E+7	9.1E+5	2.2E+5
	tokyo1	3.9E+5	5.1E+5	1.3E+5	2.7E+4
MATLAB	lsvt	1.7E+8	4.0E+8	2.3E+8	6.2E+7
	micro-mass	9.1E+6	9.5E+6	3.7E+5	7.9E+4
	schlvote	1.6E+5	5.2E+5	3.6E+5	9.0E+4
TFSom	tokyo1	1.2E+6	1.3E+6	9.1E+4	3.9E+4
	analcatdataoly	1.8E+5	2.3E+5	4.6E+4	1.7E+4
	sleuth_ex1221	2.0E+4	5.8E+4	3.8E+4	8.1E+3
RKoh	lsvt	5.5E+8	6.9 E+8	1.4E+8	3.9E+7
	micro-mass	1.2E+7	1.3E+7	4.0E+5	9.3E+4
	schlvote	6.7E+5	9.8E+5	3.1E+5	6.4E+4

TABLE III
WIDEST-3 DIFFERENCES IN TOPOGRAPHIC PRODUCT.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	analcatdata_reviewer	0.70	1.94	1.24	0.29
	arsenic-male-bladder	0.16	0.50	0.34	0.07
	arsenicfemalebladder	0.18	0.52	0.34	0.07
MATLAB	analcatdata_reviewer	1.86	3.52	1.66	0.42
	analcatdata_neavote	2.24	3.84	1.60	0.38
	aids	0.26	1.50	1.24	0.24
TFSom	haberman	0.25	3.77	3.52	1.10
	energy-efficiency	0.37	2.80	2.43	0.75
	rmftsa_ctoarrivals	0.28	2.26	1.98	0.60
RKoh	analcatdata_reviewer	1.66	2.41	0.75	0.14
	Titanic	0.36	0.72	0.36	0.11
	analcatdata_neavote	0.15	0.37	0.22	0.06
RKoh's negative TP values	fri_c0_500_50	-0.17	-0.14	0.03	0.01
	fri_c0_250_50	-0.15	-0.12	0.03	0.01
	fri_c0_500_25	-0.15	-0.13	0.02	0.01
	fri_c1_1000_10	-0.14	-0.12	0.02	0.01
	analcatdatabankruptcy	-0.003	0.01	0.01	0.002
	autoUniv-au6-750	-0.003	0.003	0.006	0.002
	volcanoes-e4	-0.003	0.002	0.005	0.001

good fit on the left (mostly dark blue) and the worse fit on the right (more green and light blue neurons).

Table II shows the widest-3 ranges across runs. For example, in MiniSom, for dataset lsvt, quantization error varied between 5.8×10^8 and 9.7×10^8; for the same dataset, but using MATLAB, the range varied between 1.7×10^8 and 4×10^8. Therefore, MiniSom and MATLAB have non-overlapping ranges across our 30-run experiments, which is a source for concern. Finally, note that the minimum vs. the maximum quantization error can vary by 2x–3x, e.g., MATLAB schlvote (min: 1.6×10^5, max: 5.2×10^5) or TFSom sleuth_ex1221 (min: 2.0×10^4, max: 5.8×10^4). A quantization error that differs by a factor of 3 across different runs raises a reason for concern.

B. *Topographic Product*

1) Definition: The topographic product (TP) indicates whether the size of the map is an appropriate fit onto the dataset. TP is computed by comparing the ranking orders in the input and output spaces, respectively; essentially, TP measures the quality of the topology preservation. If TP < 0, the map size is too small. Conversely, if TP > 0, the map size is too large. Surprisingly, we found datasets where the TP can be *positive* in one run and *negative* in the next run. However, it is important to note that the topographic product presents reliable results only for linear datasets [11].

2) Results: In Table III, we see how topographic product varies across runs and tools. Certain datasets such as analcatdata_reviewer consistently have a larger topographic product, indicating a larger mapsize. However despite this consistency, certain toolkits have better results. For example, in MiniSom, the min TP for analcatdata_reviewer is 0.70 and the max TP is 1.94, which, while still greater than 0, it is the best performing toolkit as its max value is around the minimum value of the other toolkits. For instance, MATLAB has a max TP of 3.52 and min of 1.86 for the same dataset.

In the last seven rows of Table III we focus on the RKoh toolkit. While TP is positive for the other three toolkits in all cases, RKoh managed to produce negative TP values for some datasets where other toolkits had positive TP values (see the four fri_c* rows). Additionally, RKoh also managed to simultaneously indicate that a dataset's SOM size is too small and too large as shown with the datasets analcatdata_bankruptcy, autoUniv-au6-750, and volcanoes-e4 (last three rows). Figure 5 provides a visualization of TP nondeterminism for dataset colic,[4] with a good fit on the left and a poor fit on the right.

C. *Trustworthiness/Neighborhood Preservation*

1) Definition: Trustworthiness and Neighborhood Preservation are both topological preservation measures. Trustwor-

[4] Horse surgery: surgical lesions and surgery outcome dataset.

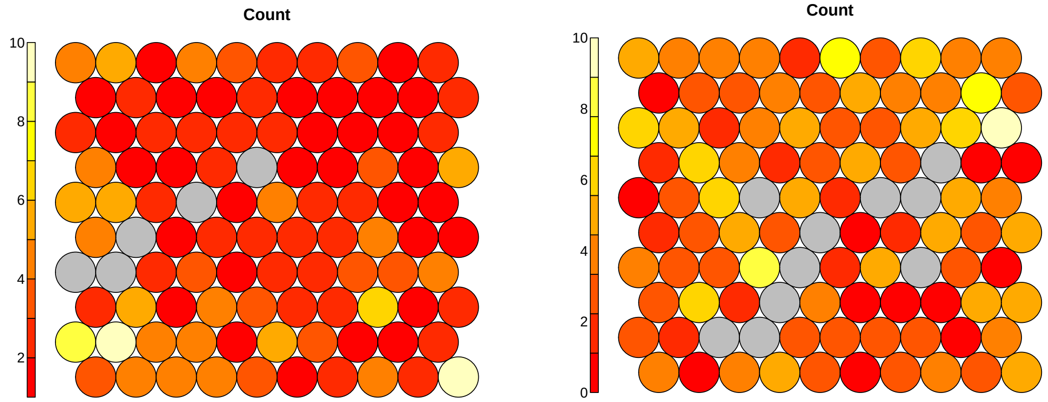


Fig. 5. Topographic product nondeterminism for dataset colic, toolkit RKoh, exposed by plotting the number of inputs mapped to each neuron. Grey spaces (representing empty nodes) indicate that the map size is too large. The left map (predominantly red or darker orange) shows a more uniform distribution, $TP = 0.0006$. The right map shows more empty nodes and thus a higher topographic product, $TP = 0.0023$; yellow or lighter orange spaces indicate a skewed distribution, where many samples map to a single node.

TABLE IV
WIDEST-3 DIFFERENCES IN TRUSTWORTHINESS.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	kc2	0.02	0.85	0.83	0.27
	blood-transfusion	0.24	0.79	0.55	0.1
	cm1_req	0.60	0.99	0.39	0.13
MATLAB	kc2	0.19	0.69	0.50	0.21
	dbworld-subjects	0.41	0.73	0.32	0.07
	dbworld-subjects-stem	0.50	0.78	0.28	0.07
TFSom	kc2	0.35	0.94	0.59	0.14
	Titanic	0.35	0.88	0.53	0.23
	pc1_req	0.45	0.88	0.43	0.14
RKoh	cm1_req	0.66	0.99	0.33	0.05
	blood-transfusion	0.54	0.86	0.32	0.13
	dbworld-subjects-stem	0.67	0.83	0.16	0.03

TABLE V
WIDEST-3 DIFFERENCES IN DISTORTION.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	micro-mass	1.5E+15	1.6E+15	1.9E+14	4.6E+13
	oil_spill	1.3E+13	3.3E+13	2E+13	4.9E+12
	tokyo1	1.2E+13	2.2E+13	1E+13	2.2E+12
MATLAB	micro-mass	1.9E+15	2.3E+15	3.8E+14	1E+14
	tokyo1	4.2E+12	6E+12	1.8E+12	4.6E+11
	PieChart3	1E+11	6.2E+11	5.2E+11	1.1E+11
TFSom	oil_spill	3.0E+13	3.5E+13	5.4E+12	2.7E+12
	tokyo1	1.9E+13	2.3E+13	3.5E+12	8.0E+11
	analcatdataoly	1.5E+12	2.2E+12	7.0E+11	2.2E+11
RKoh	lsvt	5.1E+19	7.3E+19	2.2E+19	5.6E+18
	micro-mass	9.2E+14	9.6E+14	4.4E+13	1E+13
	analcatdatabo	1.7E+12	2.9E+12	1.2E+12	3E+11

TABLE VI
WIDEST-3 DIFFERENCES IN KRUSKAL-SHEPARD ERROR.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	chscase_adopt	0.09	0.24	0.15	0.03
	kc1-binary	0.07	0.21	0.14	0.02
	arsenic-female-lung	0.12	0.26	0.14	0.03
MATLAB	analcatdata_reviewer	0.00	0.22	0.22	0.05
	analcatdata_neavote	0.03	0.18	0.15	0.05
	fri_c4_250_100	0.24	0.39	0.15	0.04
TFSom	eucalyptus	0.05	0.28	0.23	0.07
	fri_c4_250_100	0.21	0.41	0.20	0.04
	fri_c4_500_100	0.16	0.34	0.18	0.04
RKoh	analcatdata_reviewer	0.02	0.14	0.12	0.04
	ar5	0.06	0.15	0.09	0.02
	aids	0.06	0.14	0.08	0.02

thiness displays whether the projected data points that are visualized are actually close to each other in the input space. Whenever one of the neighbors on the map lattice is not one of the closest neighbors in the actual input space, the error is increased. Trustworthiness is calculated from the average of these errors. By swapping the input and output space rankings in the calculations, we obtain Neighborhood Preservation. This penalizes the data points which are close in the input space but far apart in the output space. Both Trustworthiness and Neighborhood Preservation values are weighted to be kept within 0 to 1 (where 1 means perfect).

2) Results: Table IV shows trustworthiness results. For RKoh, while there is variation across runs, ranging from 0.33 to 0.16, the overall trustworthiness is ideal, with values being higher than 0.50 and closer to 1. MiniSom, however, has the widest range (min 0.02, max 0.85) for dataset for kc2.

D. Distortion

1) Definition: Distortion is essentially the cost function that the SOM tries to optimize: the sum of squared Euclidean distances between samples and SOM prototypes, weighted by a neighborhood function that depends on the distances to the map's best-matching unit. As distortion measures loss (that the SOM function minimizes), a lower distortion is more desirable.

2) Results: Table V shows the differences in distortion. We observed high distortion in RKoh (lsvt where range was 2.2×10^{19}); for MATLAB, the largest range in variation is found in the dataset micro-mass with 3.8×10^{14}. Similarly for MiniSom, we see the same dataset having a range of 1.9×10^{14}. For a more apparent understanding of distortion, Figure 6 visualizes how distortion varies between runs on the same dataset, analcatdata_boxing1.[5]

E. Kruskal-Shepard Error

1) Definition: This value measures distance preservation between the input space and the output space. The input space is measured using Euclidean distance; in the output space, Manhattan distance between the best matching units is used.

[5] Boxing match results.

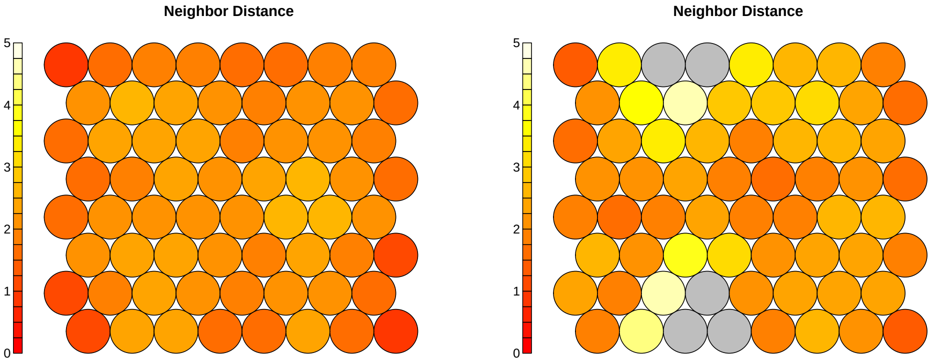


Fig. 6. Distortion nondeterminism: in the analcatdata_boxing1 dataset, toolkit RKoh, there are variations in distortion between each node and its neighbors. The figure on the left ($distortion = 4.36$) shows significantly less distortion than the right ($distortion = 6.53$): orange indicates more similar nodes. The higher the distance, the more dissimilar the nodes are (depicted in yellow or white). An ideal mapping would have predominantly red nodes.

2) Results: A low Kruskal-Shepard Error value is desirable as it indicates better preservation between input and output spaces. Table VI shows the differences that occur across toolkits and runs. We see the best error rate (0), but largest range (0.22) for MATLAB with the dataset analcatdata_reviewer. For RKoh, again analcatdata_reviewer shows the greatest variance, having a range of 0.12. Finally, for MiniSom we see the largest error (0.24) and range (0.15) in the dataset chscase_adopt.

F. Topographic Error

1) Definition: Topographic Error (TE), akin to Trustworthiness, is the ratio of total number of errors and number of data points on a SOM. TE is normalized to a range from 0 to 1, where 0 indicates perfect topological preservation.

2) Results: For lack of space we omit a Top-3 table, but results are in line with the nondeterministic outcomes we observed for other metrics. For example, analcatdata_neavote's best run with MATLAB has a min TE of 0.04 but its worst run is a TE of 0.96, showing a range of 0.92, whereas the other toolkits' ranges were 0.73 and 0.64, respectively.

V. Nondeterminism Results: External metrics

So far we studied internal map qualities; we now change our focus to external qualities, measuring SOM performance on clustering tasks. Specifically, external metrics are computed by comparing SOM-induced output with ground truth's class labels. The number of neurons is set to match the number of distinct output classes to be classified, i.e., map size is C, the number of distinct output classes (recommended size when the number of clusters is known [25], [26]). We used the same hexagonal topology and Manhattan distance as in Section IV. We now define external metrics and present the results.

A. Clustering Accuracy

1) Definition: Clustering Accuracy divides the number of samples correctly classified by the total number of samples.

2) Results: To emphasize the potential consequences of nondeterminism for medical analysis, Figure 7 shows the clustering accuracy of MiniSom, MATLAB, and RKoh on AP Colon Lung. Accuracy of MiniSom varies significantly per run (0.24–0.7): this bimodal distribution can be interpreted as a coin toss for classification, which is undesirable. Table VII further details how the accuracy varies greatly across runs and tools. *Hence when using SOMs for medical data analysis, a particular run can influence the outcome decisively.*

TABLE VII
Widest-3 differences in Clustering Accuracy.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	confidence	0.40	0.86	0.46	0.12
	AP_Prostate_Lung	0.24	0.69	0.45	0.16
	AP_Omentum_Prostate	0.60	0.97	0.37	0.16
MATLAB	solar-flare	0.39	0.69	0.30	0.09
	dbworld-bodies	0.55	0.83	0.28	0.08
	dbworld-bodies-stem	0.58	0.86	0.28	0.07
TFSom	ar3	0.50	0.77	0.27	0.11
	mw1	0.51	0.76	0.25	0.11
	CostaMadre1	0.50	0.75	0.25	0.11
RKoh	AP_Omentum_Prostate	0.57	0.98	0.41	0.11
	AP_Endometrium_Lung	0.50	0.90	0.40	0.12
	water-treatment	0.50	0.83	0.33	0.05

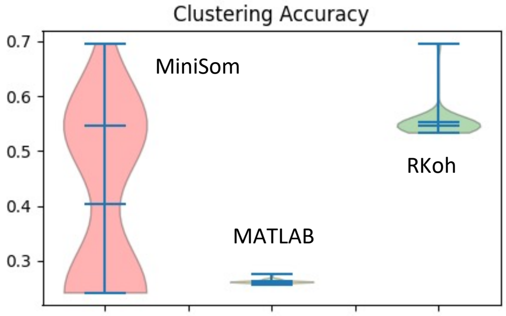


Fig. 7. Clustering accuracy ranges for dataset AP Colon Lung.

B. Purity

1) Definition: Purity is calculated by assigning each cluster to the class which is most frequent in the cluster, and computing the ratio between how many points are accurately assigned to the total number of points. A higher purity value indicates better SOM clustering performance.

2) Results: Table VIII shows the widest-3 results. For MiniSom and RKoh we observed higher purity values, occasionally at the expense of wide range (e.g., on AP_Omentum_Prostate, purity ranges were as high as 0.41).

C. Class Scatter Index (CSI)

1) Definition: The class scatter index measures how the ground truth labels are scattered in the SOM map. Classes

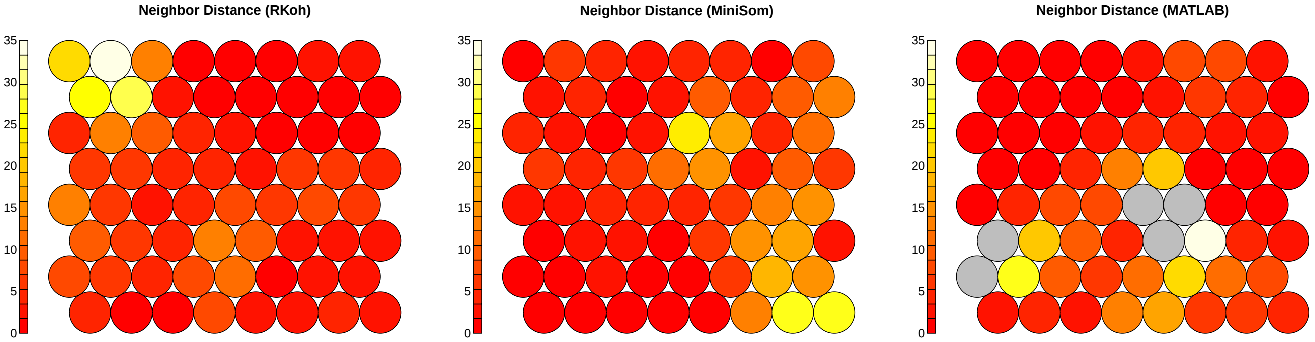


Fig. 8. Neighborhood Preservation inconsistency in the dataset analcatdata_challenger, toolkit RKoh. Though invariant across runs, NP varies across toolkits. The red indicates an ideal mapping, with fewer samples being mapped to the same node. In contrast, yellow or white indicate many samples mapped to the same node, showing a poor map fit and neighborhood preservation. Grey represents empty nodes, i.e., map might be too large.

TABLE VIII
WIDEST-3 DIFFERENCES IN PURITY.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	AP_Omentum_Prostate	0.60	0.97	0.37	0.16
	confidence	0.50	0.86	0.36	0.09
	Smartphone	0.48	0.75	0.27	0.07
MATLAB	dbworldbodies	0.54	0.82	0.28	0.08
	dbworldbodies-stem	0.57	0.85	0.28	0.07
	confidence	0.55	0.79	0.24	0.05
TFSom	aids	0.50	0.74	0.24	0.11
	analcatdatahappiness	0.38	0.58	0.20	0.07
	pollution	0.52	0.70	0.18	0.06
RKoh	AP_Omentum_Prostate	0.56	0.97	0.41	0.11
	dbworldbodies-stem	0.54	0.85	0.31	0.05
	dbworldbodies	0.54	0.81	0.27	0.04

that are not scattered (i.e., distributed into fewer groups of neighboring units) indicate a better map.

2) Results: Table IX shows the widest-3 CSI results. We see that with RKoh the CSI is varied but better performing with the max found in breast-tissue and user-knowledge of 2.00. Other than leaf which has a large value across toolkits, the remaining datasets have a max CSI of 2.00. Meanwhile, for MiniSom and MATLAB with the dataset amazon-commerce-rev we have a max of 7.90 and 7.16 respectively, indicating a map with larger groups of neighboring units than ideal.

TABLE IX
WIDEST-3 DIFFERENCES IN CSI.

Toolkit	Dataset	Min	Max	Range	Stddev
MiniSom	amazon-commerce-rev	4.80	7.90	3.10	0.83
	leaf	3.47	5.53	2.06	0.50
	eucalyptus	1.00	2.60	1.60	0.41
MATLAB	amazon-commerce-rev	5.42	7.16	1.74	0.44
	leaf	3.00	4.67	1.67	0.42
	LED-display	2.00	2.90	0.90	0.17
TFSom	soybean	1.95	3.68	1.73	0.37
	spectrometer	2.54	4.17	1.63	0.53
	leaf	3.27	4.77	1.50	0.36
RKoh	leaf	3.80	5.50	1.70	0.39
	breast-tissue	1.00	2.00	1.00	0.19
	user-knowledge	1.00	2.00	1.00	0.27

VI. INCONSISTENCY

We believe that SOM toolkit users should expect toolkits to be interchangeable: when training an SOM on the same dataset via different toolkits one would expect if not the same, at least remotely similar results. However our experiments show that this expectation is typically not met, e.g., the results of two different toolkits on the same dataset are *inconsistent*. We first illustrate inconsistency, then introduce the statistical test and its results, and finally discuss the toolkits and datasets that display the strongest contrast between toolkits.

A. Inconsistency Examples

To emphasize the consequences of inconsistency on a medical dataset note that in Figure 7, on dataset AP Colon Lung, one toolkit's observed accuracy could be 3 times as high compared to another toolkit. *Hence when using SOMs for medical data analysis, a particular toolkit can influence the outcome decisively.*

B. Statistical Test and Results

To expose statistically significant inconsistency between toolkits, for each dataset and each pair of toolkits, we ran a Mann-Whitney U test where the two populations were the clustering accuracies (30 runs). If $p < 0.05$ we conclude that the toolkits are inconsistent. The number of datasets displaying inconsistency are shown in Table X; these numbers translate to *51–92% of datasets yielding inconsistent results*.

TABLE X
#DATASETS WITH STATISTICALLY SIGNIFICANT INCONSISTENCY.

MiniSom vs. MATLAB	MiniSom vs. RKoh	MATLAB vs. RKoh	MATLAB vs. TFSom	MiniSom vs. TFSom	RKoh vs. TFSom
298	254	196	332	333	332

Even in those rare cases where toolkits are deterministic on a certain dataset (the few non-zero values in Table I) inconsistencies still arise between toolkits. For example, Figure 8 shows how Neighborhood Preservation is inconsistent for the deterministic dataset analcatdata_challenger.[6]

C. Mutual ARI Comparison

We now quantify and discuss those cases where the resulting SOMs disagree strongly between toolkits. We use the Adjusted Rand Index (ARI), a metric introduced by Hubert and Arabie [27] that indicates how dissimilar two clusterings of the same dataset are. An $ARI = -1$ indicates strong dissimilarity between clusterings, $ARI = 0$ suggests that the clusterings are independent, whereas $ARI = 1$ indicates a perfect agreement.

For each dataset, we compute "mutual ARIs" between all toolkits pairs, that is, ARI scores between all six toolkits pairs.

[6] Space Shuttle Challenger parameters.

TABLE XI
WORST-3 INCONSISTENCIES (MUTUAL ARI) ACROSS TOOLS.

MiniSom vs. MATLAB		MiniSom vs. RKoh		MATLAB vs. RKoh		MATLAB vs. TFSom		MiniSom vs. TFSom		RKoh vs. TFSom	
shuttle-landing-c	-0.14	shuttle-landing-cl	-0.14	trains	-0.13	pasture	-0.08	MyIris	-0.11	MyIris	-0.12
trains	-0.07	fabert	-0.08	dbworld-bodies	-0.06	ar4	-0.07	pasture	-0.08	pasture	-0.08
fri_c4_100_50	-0.06	triazines	-0.06	dbworld-sbjs-s	-0.05	wine	-0.07	wine	-0.07	wine	-0.07

In other words, for each toolkits pair, say RKoh vs. TFSom, we compute the 30 runs $\times$ 30 runs ARI scores. We focus on the minimum of these 900 pairs, as it indicates the worst-possible disparity users can experience.

We present the worst disparities in Table XI. The strongest observed dissimilarity were between MiniSom and MATLAB, and MiniSom and RKoh, respectively: for dataset shuttle-landing-control we have $ARI = -0.14$. When comparing MATLAB and RKoh, other than trains with $ARI = -0.13$, we see that overall the discrepancy between toolkits is much less than compared with MiniSom. These negative ARI values are concerning, because a negative ARI indicates that the clusterings achieved via the two toolkits are worse than unrelated and tending toward disagreement.

VII. Related Work

We are not aware of any work that addresses SOM reliability. Nondeterminism and inconsistency were studied before, but in the context of discrete clustering algorithms [21], [28], [29] rather than neural networks. The literature discusses how to use SOM effectively, e.g., in image classification [26] and choosing appropriate weights and features correctly [30]. To better understand SOM functionality and performance, a variety of SOM quality metrics have been proposed by Forest et al. [11] Pölzlbauer [9], Lutz [10], yet there were no investigations based on variations of SOM results. Overall, we have found no other investigation into quantifying SOM disparities, either across runs or across toolkits.

VIII. Conclusions

Given the popularity of SOMs and neural networks in general, we conduct the first study to investigate SOM reliability in terms of determinism and consistency. Running four popular SOM packages on 381 datasets shows that users should expect wide variation across runs and toolkits. Our findings indicate a need to scrutinize SOM results, especially in high-stakes scenarios. Our study could spur further research into the causes of, and remedies for, SOM nondeterminism and inconsistency.

Acknowledgments

We thank Pranav Vijayanath for his help with TFSom, and the anonymous reviewers for their valuable feedback. This material is based upon work supported by the National Science Foundation under Grant No. CCF-2007730.

References

[1] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[2] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on neural networks*, vol. 11, no. 3, 2000.

[3] E. Séverin, "Self organizing maps in corporate finance: Quantitative and qualitative analysis of debt and leasing," *Neurocomputing*, vol. 73, no. 10, pp. 2061–2067, 2010.

[4] G. Deboeck and T. Kohonen, *Visual explorations in finance: with self-organizing maps*. Springer Science & Business Media, 2013.

[5] G. Schneider and P. Schneider, "Macromolecular target prediction by self-organizing feature maps," *Expert opinion on drug discovery*, 2016.

[6] P. Schneider, Y. Tanrikulu, and G. Schneider, "Self-organizing maps in drug discovery: Compound library design, scaffold-hopping, repurposing," *Current medicinal chemistry*, vol. 16, pp. 258–66, 02 2009.

[7] A. Skupin, J. R. Biberstine, and K. Börner, "Visualizing the topical structure of the medical sciences: A self-organizing map approach," *PLOS ONE*, vol. 8, no. 3, 03 2013.

[8] G. Stiglic and P. Kokol, "Stability of ranked gene lists in large microarray analysis studies," *Journal of biomedicine & biotechnology*, vol. 2010, p. 616358, 06 2010.

[9] G. Pölzlbauer, "Survey and comparison of quality measures for self-organizing maps," in *Proceedings of the Fifth Workshop on Data Analysis (WDA'04)*, 2004, pp. 67–82.

[10] L. Hamel, "Som quality measures: An efficient statistical approach," in *Advances in Self-Organizing Maps and Learning Vector Quantization*, 2016, pp. 49–59.

[11] F. Forest, M. Lebbah, H. Azzag, and J. Lacaille, "A survey and implementation of performance metrics for self-organized maps," 2020.

[12] "Mathworks fast facts," May 2019, https://www.mathworks.com/company/aboutus.html.

[13] M. Hornick, "Oracle r technologies overview," https://www.oracle.com/assets/media/oraclertechnologies-2188877.pdf.

[14] "TensorFlow Self-Organizing Map," April 2021, https://github.com/cgorman/tensorflow-som.

[15] "MiniSom Self Organizing Maps," April 2021, https://github.com/JustGlowing/minisom.

[16] "CRAN package kohonen: Supervised and Unsupervised Self-Organising Maps," April 2021, https://cran.r-project.org/web/packages/kohonen/index.html.

[17] "The R Project for Statistical Computing," April 2021, https://www.r-project.org/.

[18] "MATLAB selforgmap," April 2021, https://www.mathworks.com/help/deeplearning/ref/selforgmap.html.

[19] "Tensorflow github," https://github.com/tensorflow/tensorflow.

[20] "OpenML," April 2021, https://www.openml.org/.

[21] X. Yin, V. Musco, I. Neamtiu, and U. Roshan, "Statistically rigorous testing of clustering implementations," in *AITEST 2019*, April 2019.

[22] H. Levene, "Robust tests for equality of variances," in *Contributions to probability and statistics: essays in honor of Harold Hotelling*. Stanford University Press: Palo Alto, CA, USA, 1960, pp. 278–292.

[23] M. B. Brown and A. B. Forsythe, "Robust tests for the equality of variances," *J. of the American Statistical Association*, vol. 69, 1974.

[24] M. Lotfi, D. Moazzami, B. Moshiri, and M. Delavar, "Anomaly detection using a self-organizing map and particle swarm optimization," *Scientia Iranica*, vol. 18, pp. 1460–1468, 12 2011.

[25] NeuPy, "NeuPy Neural Networks in Python," March 2021, http://neupy.com/apidocs/neupy.algorithms.competitive.sofm.html.

[26] D. Pratiwi, "The use of self organizing map method and feature selection in image database classification system," *International Journal of Computer Science Issues*, vol. 9, 06 2012.

[27] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 02 1985.

[28] V. Musco, X. Yin, and I. Neamtiu, "Smokeout: An approach for testing clustering implementations," in *ICST 2019*, April 2019.

[29] X. Yin, I. Neamtiu, S. Patil, and S. T. Andrews, "Implementation-induced inconsistency and nondeterminism in deterministic clustering algorithms," in *ICST 2020*, October 2020.

[30] N. Singh, "Self-organizing maps for machine learning algorithms," Jun 2018. [Online]. Available: https://medium.com/@navdeepsingh_2336/self-organizing-maps-for-machine-learning-algorithms-ad256a395fc5