2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Automatic Distance-Based Interpolating Unit Detection and Pruning in Self-Organizing Maps

Willem S. van Heerden
Department of Computer Science
University of Pretoria
Pretoria, South Africa
https://orcid.org/0000-0002-9736-7268

***Abstract*—The self-organizing map (SOM) is an unsupervised neural network that uses neuron weight vectors to model training data. Some neurons, called interpolating units, have weight vectors that do not model training data, and instead represent boundaries between emergent data clusters. Interpolating units are useful for distinguishing such clusters using SOM visualizations. However, automatic (non-visual) detection of interpolating units would be advantageous for SOM analysis. This paper proposes a novel algorithm, based on inter-neuron distances in weight vector space, for identifying and possibly pruning interpolating units. Existing methods for interpolating unit detection are surveyed, highlighting drawbacks not associated with the proposed algorithm. Focusing on classification task performance, SOMs which are pruned using the proposed algorithm are compared to unpruned SOMs. This analysis demonstrates that interpolating unit pruning does not adversely affect SOM model quality, and suggests that the proposed algorithm warrants further investigation for application in data science.**

***Index Terms*—artificial neural networks, self-organizing feature maps, data science, data mining, emergent phenomena**

I. Introduction

The self-organizing map (SOM) is an unsupervised neural network [1], which has been the focus of abundant research [2]–[4]. SOMs are useful for data analysis, and have been used in fields as varied as the finance [5], medicine [6], astronomy [7], and pandemic analysis [8], [9].

A SOM consists of neurons, each of which has a weight vector. The weight vectors model the SOM's training data. Often, after SOM training, a subset of weight vectors do not represent any of the modeled data. The neurons of such weight vectors form boundaries between neuron groups representing training data clusters, and are called interpolating units.

In many SOM visualizations [10], interpolating unit marking allows humans to easily analyze clusters. However, SOM-based cluster discovery [11] and rule extraction [12]–[14], use the weight vectors as a model of the SOM's training data. Here, interpolating units are noise that does not characterize training data [15]. It is thus hypothesized that interpolating unit removal will simplify the SOM data model, but not reduce the performance of methods that rely on this data model.

This paper proposes a novel algorithm for detecting and pruning interpolating units. The algorithm relies on the distances between neighboring neurons' weight vectors. The paper surveys existing methods for interpolating unit detection, and identifies associated drawbacks which do not affect the proposed algorithm. Finally, an empirical analysis compares the performance of SOMs pruned by the algorithm and SOMs that remained unpruned. The experiments used a classification task performed on several data sets.

The remainder of this paper is organized as follows: Section II discusses SOMs. Section III describes the properties of interpolating units, while Section IV surveys and critically discusses existing interpolating unit detection methods. Section V discusses the novel algorithm for interpolating unit detection and pruning, and Section VI outlines the empirical investigation. Lastly, Section VII concludes the paper and suggests future avenues of research.

II. Self-Organizing Maps

The SOM is an unsupervised neural network developed by Kohonen [16]. A SOM can train on unclassified data, unlike supervised learning methods, and is thus unaffected by biased class information. SOMs are more versatile than supervised methods due to the prevalence of unclassified data.

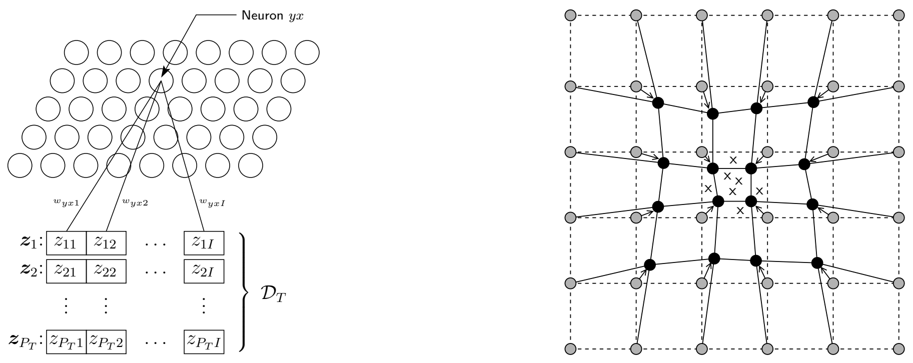
Fig. 1 (a) shows the SOM architecture. A training data set is denoted $\mathcal{D}_T = \{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{P_T}\}$. Each training example is an I-dimensional vector denoted $\vec{z}_s = (z_{s1}, z_{s2}, \ldots, z_{sI})$. Each attribute value, z_{sv}, within $\vec{z}_s$ is a real value.

A SOM has a map structure of neurons, usually forming a two-dimensional $Y \times X$ grid, where Y and X are the number of rows and columns, respectively. The neuron at row x and column y is n_{yx}. A lattice defines the connections between adjacent neurons. Each n_{yx} has an I-dimensional weight vector, $\vec{w}_{yx} = (w_{yx1}, w_{yx2}, \ldots, w_{yxI})$. The real-valued weight w_{yxv} represents training set attribute z_{sv}.

SOM training adapts the map structure to model the I-dimensional training data, where the map has fewer dimensions than I. The SOM model has two characteristics:

1) The map models the training data's probability density function. A neuron represents similar training examples, and weight vectors fall within dense training data areas.
2) The model preserves the training data's local topological structure. Mutually similar data examples are represented by neurons close to one another in the map space.

Fig. 1 (b) shows the outcome of SOM training using a two-dimensional map and training set. Crosses represent training examples. Gray circles are the original weight vector positions, where dashed lines connect adjacent weight vectors. Black


Neuron yx
w_{yx1}
w_{yx2}
w_{yxI}
$\boldsymbol{z}_1$: z_{11} z_{12} ... z_{1I}
$\boldsymbol{z}_2$: z_{21} z_{22} ... z_{2I}
$\boldsymbol{z}_{P_T}$: z_{P_T1} z_{P_T2} ... z_{P_TI}
$\mathcal{D}_T$

(a) (b)

Fig. 1. General structure and operation of a SOM. (a) Architecture of a SOM. (b) The local effect of SOM training on two-dimensional data.

circles show weight vectors after training, with solid lines connecting adjacent weight vectors. Weight vector movement towards training examples illustrates the first property of SOM models. The location of training examples close to adjacent trained weight vectors highlights the second property.

Several SOM training algorithms exist, all of which update weight vectors to achieve the two model properties. The popular stochastic method [16] is assumed here, which repeatedly selects a training example and updates all weight vectors relative to the example's map location. The specifics of stochastic training are comprehensively described by Kohonen [1].

After SOM training, it is possible to find the neuron (called the best matching unit or BMU) that is responsible for modeling a data example. For data example $\vec{z}_s$, the BMU has the weight vector that is closest to $\vec{z}_s$ using a distance measure. Euclidean distance is often used, and is assumed here.

III. Properties of Interpolating Units

Interpolating units are neurons with weight vectors that do not represent part of the training data a SOM models [17], and are thus redundant. Interpolating units are a natural byproduct of stochastic SOM training, and have the following properties:

1) Interpolating units are isolated in the map, with weight vectors far from all neighboring weight vectors. Usually the distance measure that defines BMUs is used.
2) Because interpolating units do not model training data, such neurons are usually not BMUs of any data examples from the same distribution as the training data.

An emergent cluster is a set of neurons with similar weight vectors, which model a cluster in the training data. Interpolating units often form borders between emergent clusters. Interpolating unit marking thus often aids data analysis.

Interpolating unit pruning has potential benefits when methods focus only on emergent clusters. For example, pruning improves the performance when clustering algorithms discover emergent clusters, which Section IV-B describes [11], [17].

IV. Existing Detection Techniques

Existing interpolating unit detection techniques can be extended into pruning methods by removing the identified neurons from the map. Pruning disrupts the neuron grid's lattice, possibly disconnecting parts of the map. This prevents the use of methods relying on a fully connected lattice.

Detection methods are based on the two properties of interpolating units. Most existing interpolating unit detection techniques use map structure visualizations to allow human-led exploratory discovery of interpolating units. An interactive tool is used to view the visualization and manually mark interpolating units. Some examples of such visualizations are shown in Fig. 2. All visualizations are based on the same SOM, which was trained on the Iris data set [18].

A. Detection Using Local Distance

The first category of interpolating unit detection methods focuses on distances between the weight vectors of neighboring neurons. These techniques thus identify interpolating units using the first property that Section III mentions [17].

Fig. 2 (a) shows a U-matrix [19] of the Iris SOM. Hexagons containing dots represent neurons, and empty hexagons denote connections between neurons. Grayscale encoding represents inter-weight vector distances. Lighter shades represent smaller distances and darker shades indicate larger distances. Dark areas contain interpolating units that are far from all neighboring neurons. The U-matrix shows that interpolating units often benefit SOM visualizations, because these units help humans distinguish emergent clusters (here interpolating units form a boundary between two lighter-colored clusters).

An aggregate distance matrix [20] is less detailed than a U-matrix. Only the mean or median distance between each neuron and the neuron's neighbors is shown.

Weight vector projections visually preserve all inter-weight vector distances [21]. Fig. 2 (b) shows a Sammon's mapping projection [22] of the Iris SOM. Points represent weight vectors and lines connect weight vectors of adjacent neurons. Points surrounded by large spaces are interpolating units.

Gradient fields [23] represent, for each n_{yx}, a visualization vector pointing towards the probable cluster center of n_{yx}. The visualization vector of n_{yx} is influenced by all other neurons, where neurons with weight vectors closer to $\vec{w}_{yx}$ contribute more. Thus the visualization vector of n_{yx} points towards

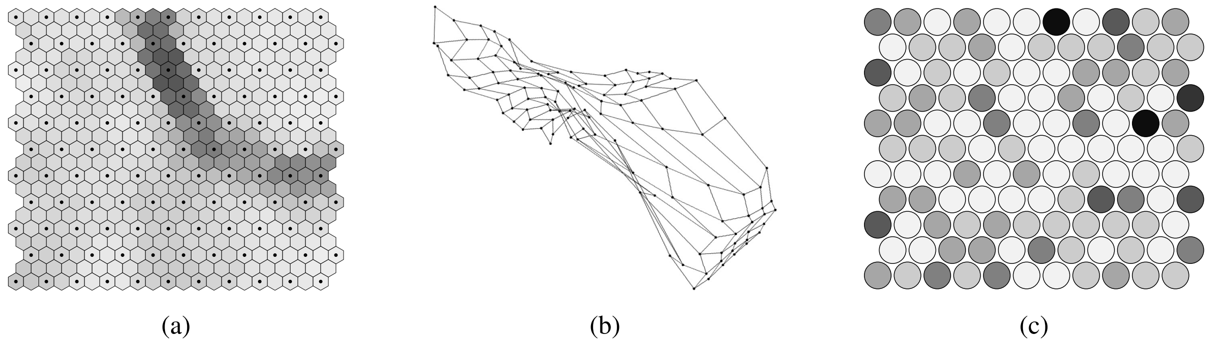


Fig. 2. Visualizations of the same SOM trained on the Iris data set. (a) U-matrix. (b) Sammon's mapping. (c) Data histogram.

neurons closer in weight space. Neurons farther from n_{yx} in map space are also scaled to contribute less to the visualization vector of n_{yx}. Information from local neuron neighborhoods is thus more strongly represented. The degree of the scaling increases or decreases the effect of more distant neurons. A gradient field is therefore similar to a U-matrix or aggregate distance matrix, but takes a variable amount of local weight information into account. In a gradient field visualization, a neuron is an interpolating unit if all neighboring visualization vectors point away from the neuron.

A borderline visualization [17] is similar to a gradient field visualization. The visualization draws a line segment orthogonal to the gradient field visualization vector for each n_{yx}. Line segments thus represent likely cluster boundaries. Longer line segments surround interpolating units on the map.

B. Detection Using Cluster Discovery

It is possible to directly detect emergent clusters as a whole. Emergent clusters are discovered by applying a clustering algorithm to the map weight vectors [11]. Such methods group together similar weight vectors that represent emergent clusters. These methods thus also consider the first property of interpolating units, but focus on larger uniform map areas. Interpolating units are neurons that fall on the borders between clusters, and are identifiable if clusters are visualized.

Many general-purpose clustering algorithms exist (e.g., Ward [24] and k-means [25] clustering), and are often used for emergent cluster discovery. SOM-specific clustering methods have also been proposed (e.g., the U*C algorithm [26]).

C. Detection Using Data Density

The focus now shifts to interpolating unit detection using the second property of interpolating units. These techniques thus concentrate on the density distribution of data examples as related to the weight vectors of a trained SOM.

Fig. 2 (c) shows a data histogram visualization [27]. All Iris training examples were first mapped to BMUs. The visualization grayscale encodes each neuron, showing the number of examples sharing the neuron as a BMU. The darker a neuron's shade is, the higher the number of mapped examples. Neurons with the lightest color have no mapped examples.

When comparing Fig. 2 (c) to Fig. 2 (a), it is clear that every interpolating unit is never a BMU. However, many non-BMU neurons fall within emergent clusters and are not interpolating units. This is because the data set is sparse and does not adequately cover the map. Data density is thus not a reliable indicator of interpolating units when training data are scarce, particularly if maps are very large.

Smoothed data histograms [28] map each $\vec{z}_s$ to the b neurons with $\vec{w}_{yx}$ closest to $\vec{z}_s$. Example $\vec{z}_s$ belongs to n_{yx} with a degree inversely proportional to the distance between $\vec{w}_{yx}$ and $\vec{z}_s$. Clusters and interpolating units are more easily identifiable than in a standard data histogram.

P-matrices [29] display a pareto density estimate [30], or PDE, for each neuron. The PDE for n_{yx} is the number of data points within a hypersphere radius of $\vec{w}_{yx}$. The radius defines the minimum volume that preserves maximum information. Neurons with low PDE values are interpolating units.

Graph based cluster visualizations [31] use graph edges to connect similar data examples (either the d closest examples, or examples within a hypersphere radius of one another). Examples are mapped to BMUs, and the graph connections are superimposed on the map. Networks of edges cover emergent clusters, and no edges connect to interpolating units.

Zhang and Li [32] propose non-BMU interpolating unit identification, which automatically marks neurons that are never BMUs as interpolating units. Erroneous interpolating unit identification is likely, especially for sparse training data and large maps. This approach is therefore discouraged.

D. Detection Using Combined Measures

Finally, detection can combine the two interpolating unit characteristics. Thus map weight space information and the training data's relationship to the map are both used.

A U*-matrix [33] combines the U-matrix and P-matrix. U-matrix values are modified with a scaling factor. High P-matrix values decrease corresponding U-matrix values, reducing local distance importance and smoothing out erroneous interpolating units within large clusters. For low P-matrix values, the scaling increases the corresponding U-matrix value, ensuring clear interpolating unit representation.

V. Proposed Algorithm

This section presents a novel algorithm for automatic interpolating unit detection using local distances. The algorithm avoids many problems associated with existing methods.

```
Create and train a SOM, $map$, with $Y \times X$ neurons
Define $frac$, a parameter defining a pruning threshold
for all neurons $n_{yx}$ in $map$ do
    Define $neigh_{yx}$, a set of neurons neighboring $n_{yx}$
    Define $total_{yx} = 0$, an inter-neighbor distance for $n_{yx}$
    for all neurons $n_{y'x'} \in neigh_{yx}$ do
        Update $total_{yx} = total_{yx} + \|\vec{w}_{yx} - \vec{w}_{y'x'}\|_2$
    end for
    Compute $mean_{yx} = total_{yx} \div |neigh_{yx}|$
end for
Find $mean(map)$, the mean of all $mean_{yx}$
Find $stdev(map)$, the standard deviation of all $mean_{yx}$
for all neurons $n_{yx}$ in $map$ do
    if $mean_{yx} - mean(map) > frac \times stdev$ then
        Mark $n_{yx}$ as an interpolating unit
        Optionally remove $n_{yx}$ from $map$
    end if
end for
```

Fig. 3. Algorithm for distance-based interpolating unit pruning.

Visualization requires human analysis, which is time consuming, error-prone, and biased. This affects all the methods in Section IV, except non-BMU interpolating unit identification. The proposed algorithm avoids human interpretation. Clustering algorithms, which often have shortcomings or biases [34], are also not used. Non-BMU interpolating unit identification tends to erroneously mark neurons. The proposed algorithm avoids this problem due to not relying on data density.

Fig. 3 shows the algorithm, which marks neurons unusually far from neighbors in weight space. For each n_{yx}, the mean distance is found between $\vec{w}_{yx}$ and all neighboring neuron weight vectors. The overall mean and standard deviation of these distances are computed over all n_{yx}. Neurons are marked as interpolating units (and possibly pruned) if they have mean distances greater than a fraction, *frac*, of a standard deviation from the overall mean. Algorithm efficiency is impacted mainly by the number of neurons in the map.

Fig. 4 marks interpolating units in the Iris SOM using varying *frac* values. Comparing Fig. 4 (b) to the visualizations in Fig. 2 shows that interpolating units are successfully identified if an appropriate *frac* value is used. Fig. 4 (a) shows that a lower *frac* value selects interpolating units more liberally, while in Fig. 4 (c) a larger *frac* value causes under-selection.

TABLE I
OPTIMAL SOM AND DISTANCE-BASED PRUNING PARAMETERS

Data Set	Y, X	$\eta(0)$	τ_1	$\sigma(0)$	τ_2	$frac$
Iris	5	5.488	1 432.617	2.119	77.539	0.369
Ionosphere	7	3.848	1 209.961	1.818	59.570	0.800
Monks 1	14	6.055	849.609	10.445	9.766	1.620
Monks 2	15	8.867	1 365.234	1.348	31.641	1.401
Monks 3	11	9.941	577.148	3.029	83.008	1.162
Pima	12	2.070	1 376.953	9.797	52.734	0.103

VI. Experimental Work

The proposed algorithm is only justified if it does not mark neurons critical to a SOM data model. The performance of unmodified SOMs was thus compared to SOMs pruned by the algorithm, using SOM-based data classification [35].

Fully unsupervised SOMs (both unpruned and pruned) were trained on several data sets. The experiments used the Iris, ionosphere, monk's problems, and Pima Indians diabetes data sets, all from the UCI Machine Learning Repository [36]. Data sets were preprocessed by one-hot encoding nominal attributes and scaling attribute values to a $[0.0, 1.0]$ range.

Neurons were labeled using supervised example-centric neuron labeling [37], which has been shown to be superior to other supervised labeling approaches [15]. Unsupervised labeling was not used because its results are difficult to interpret [38]. Example-centric neuron labeling matches training examples to BMUs. Each BMU is then labeled with the most common class within the BMU's matched examples. Neurons that are never BMUs are left unlabeled.

Each training set example was then matched to a BMU, the label of which became the classification of the example. Finally, the same classifications were applied to a test data set, which was unseen by the SOM during training.

Square maps were used, implemented in SOM_PAK [39] with extensions suggested by the author [15]. Training ceased when a moving average (over 30 training iterations) of the training quantization error's standard deviation reached 0.0001, or after 100 000 training iterations elapsed.

Table I shows the results of parameter tuning for unmodified SOMs on each data set. The parameters are Y and X (number of map rows and columns), $\eta(0)$ (initial learning rate), τ_1

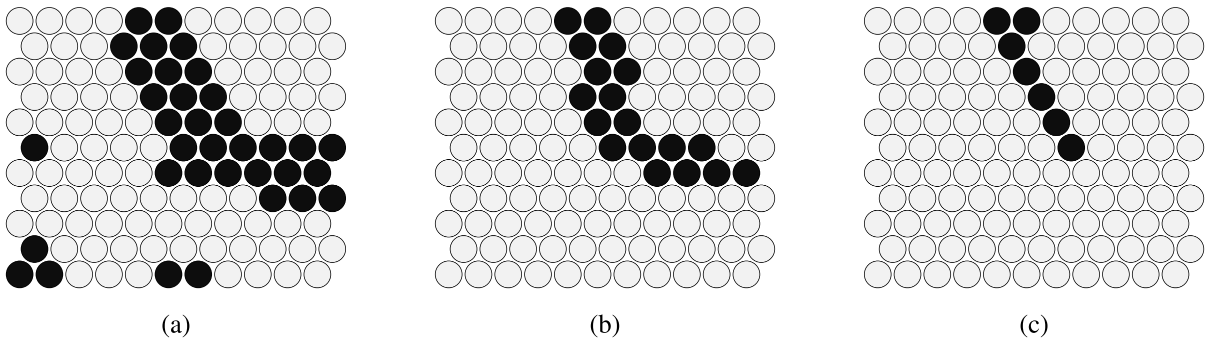


Fig. 4. Iris SOM interpolating units marked by the proposed algorithm. (a) Using $frac = 0.05$. (b) Using $frac = 1.0$. (c) Using $frac = 2.5$.

(learning rate decay constant), $\sigma(0)$ (initial kernel width), and τ_2 (kernel width decay constant). The pruned SOMs used the same parameters, but $frac$ was additionally tuned.

Tables II to V show the experimental results, each focusing on a different performance measure. Means and standard deviations of measures are reported, calculated over 30-fold cross-validations that dictated training and test set sizes. For each comparison between the unpruned and pruned approaches, a non-parametric Wilcoxon signed-rank hypothesis test [40] was performed at a 0.05 confidence level. A Bonferroni correction [41] accounted for the multiple comparisons problem. A p-value is reported for each hypothesis test. When the p-value indicates a statistically significant difference, the more optimal performance measure is highlighted in bold.

Table II shows the mean, $\overline{c}$, and standard deviation, σ_c, of training set classification error. A statistically significant performance difference is only highlighted for the ionosphere and Pima Indians diabetes data sets. Here, the unpruned SOMs outperformed the pruned SOMs, but not by very large margins. Mean performance differed by 2.02% on the ionosphere set, and 3.526% on the Pima Indians diabetes set. All other cases showed no significant differences. Pruning thus did not introduce a great training performance penalty.

Test set classification error is a better indicator of real-world algorithmic performance, and is shown in Table III. The mean, $\overline{t}$, and standard deviation, σ_t, of the measure are shown. There was no statistically significant performance difference for any of the data sets. This indicates no negative real-world classification performance penalty introduced by interpolating unit pruning, for the investigated data sets.

Table IV focuses on the mean, $\overline{u}$, and standard deviation, σ_u, of the percentage of unlabeled neurons left by the supervised example-centric neuron labeling. Unlabeled neurons represent less interesting areas of the SOM model. If pruning reduces the unlabeled neuron percentage, then neuron removal is focusing on these uninteresting (and thus redundant) areas. Pruning reduced the percentage of unlabeled neurons significantly in all cases. In the Iris, ionosphere, and Pima Indians diabetes sets, unlabeled neurons were almost entirely removed.

Finally, Table V illustrates the mean, $\overline{r}$, and standard deviation, σ_r, for the percentage of pruned neurons. Of course, no neurons were pruned from the unmodified SOMs. However, the proposed algorithm pruned a significant number of neurons for all data sets. Particularly high mean percentages were pruned for the Iris, ionosphere, and Pima Indians diabetes sets, although the standard deviations were also quite high. In all these cases more than 24% of neurons were removed, despite maps being relatively small, with less room for redundancies. It is likely that pruning will be even more effective on very large maps, where interpolating units are more prevalent.

Overall, the results indicate that the proposed algorithm's pruning was focused on the least important parts of the SOM data model. This was achieved while not significantly affecting the real-world error produced by the classifying task. In fact, the highest degrees of pruning (in the Iris, ionosphere, and Pima Indians diabetes sets) were associated with the removal of almost all unlabeled neurons. Therefore, the most aggressive pruning removed most redundant neurons from maps. The proposed algorithm thus warrants consideration for either identifying or pruning interpolating units from SOMs.

The optimal $frac$ value was problem dependent, with no clear trend in optimal values. Interestingly, low $frac$ values often resulted in higher pruning percentages (e.g., in the Iris, ionosphere, and Pima Indians diabetes sets). The level of pruning produced by a particular $frac$ value is thus strongly affected by a map's data model characteristics.

TABLE II
TRAINING SET CLASSIFICATION ERROR PERFORMANCE

Data Set	No Pruning		Pruning		p-value
	$\overline{c}$	σ_c	$\overline{c}$	σ_c	
Iris	3.655	0.705	4.046	1.067	0.165
Ionosphere	**10.137**	**1.221**	12.157	1.496	1.080×10^{-5}
Monks 1	18.126	1.084	18.373	1.452	0.390
Monks 2	15.766	0.657	15.774	0.716	0.972
Monks 3	23.828	1.792	24.386	2.002	0.117
Pima	**20.588**	**0.971**	24.114	1.002	1.863×10^{-9}

TABLE III
TEST SET CLASSIFICATION ERROR PERCENTAGE PERFORMANCE

Data Set	No Pruning		Pruning		p-value
	$\overline{t}$	σ_t	$\overline{t}$	σ_t	
Iris	4.000	8.137	2.667	6.915	0.500
Ionosphere	11.515	10.923	13.030	10.592	0.453
Monks 1	20.000	12.358	22.381	12.118	0.434
Monks 2	20.238	10.116	19.286	10.469	0.832
Monks 3	26.429	12.178	28.095	10.927	0.316
Pima	26.133	9.951	26.800	9.974	0.684

TABLE IV
UNLABELED NEURON PERCENTAGE PERFORMANCE

Data Set	No Pruning		Pruning		p-value
	$\overline{u}$	σ_u	$\overline{u}$	σ_u	
Iris	19.467	3.104	**0.400**	**1.221**	1.863×10^{-9}
Ionosphere	3.810	1.912	**0.680**	**1.116**	5.960×10^{-8}
Monks 1	43.469	3.065	**37.466**	**4.204**	2.012×10^{-7}
Monks 2	49.807	1.885	**41.067**	**2.812**	1.863×10^{-9}
Monks 3	32.700	4.967	**23.526**	**3.899**	7.451×10^{-9}
Pima	3.264	1.317	**0.440**	**0.562**	1.863×10^{-9}

TABLE V
PRUNED NEURON PERCENTAGE PERFORMANCE

Data Set	No Pruning		Pruning		p-value
	$\overline{r}$	σ_r	$\overline{r}$	σ_r	
Iris	0.000	0.000	**33.733**	**5.626**	1.863×10^{-9}
Ionosphere	0.000	0.000	**24.014**	**3.156**	1.863×10^{-9}
Monks 1	0.000	0.000	**5.731**	**1.134**	1.863×10^{-9}
Monks 2	0.000	0.000	**8.119**	**1.213**	1.863×10^{-9}
Monks 3	0.000	0.000	**11.322**	**1.711**	1.863×10^{-9}
Pima	0.000	0.000	**42.824**	**3.433**	1.863×10^{-9}

VII. Conclusion

This paper discusses interpolating units in SOMs, and surveys existing methods for identifying interpolating units. A novel algorithm for identifying and potentially pruning interpolating units is proposed. The proposed approach has several benefits over existing methods, primarily stemming from the reliance of existing approaches on fallible human analysts. An empirical investigation on several data sets is reported, which showed that the proposed algorithm successfully identified and pruned redundant interpolating units. This was achieved while not disrupting the quality of the SOM's predictive ability.

Future work will analyze the proposed algorithm's classification performance when using other labeling approaches, such as weight-centric neuron labeling [37]. Interpolating unit pruning will be investigated for supervised and semi-supervised SOMs [35]. Automatic interpolating unit pruning will also be compared to SOM-based approaches designed to avoid producing interpolating units, for example SOM-kMER [42]. Edge detection for the identification of interpolating units on the borders of discovered emergent clusters will also be investigated. The improvement of automatic interpolating unit detection using data density is another potential future avenue of investigation. Finally, automatic interpolating unit detection could also be based on the other visualization techniques mentioned in Section IV.

References

[1] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Springer-Verlag, 2001.

[2] S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) papers: 1981–1997," *Neural Comput. Surv.*, vol. 1, pp. 102–350, 1998.

[3] M. Oja, S. Kaski, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) papers: 1998–2001 addendum," *Neural Comput. Surv.*, vol. 3, pp. 1–156, 2003.

[4] M. Pöllä, T. Honkela, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) papers: 2002-2005 addendum," Helsinki Univ. Technol., Tech. Rep. TKK-ICS-R23, 2009.

[5] M. Nordlinder, "Clustering of financial account time series using Self Organizing Maps," Master's thesis, KTH Roy. Inst. Technol., 2021.

[6] P. G. Reddy, T. Ramashri, and L. Krishna, "Brain tumour region extraction using novel self-organising map-based KFCM algorithm," *Pertanika J. Sci. & Technol.*, vol. 31, no. 1, pp. 577–594, 2023.

[7] B. W. Holwerda, D. Smith, L. Porter, C. Henry, R. Porter-Temple, K. Cook, K. A. Pimbblet, A. M. Hopkins, M. Bilicki, S. Turner, V. Acquaviva, L. Wang, A. H. Wright, L. S. Kelvin, and M. W. Grootes, "Galaxy and mass assembly (GAMA): Self-Organizing Map application on nearby galaxies," *Mon. Not. R. Astron. Soc.*, vol. 513, no. 2, pp. 1972–1984, 2022.

[8] D. Galvan, L. Effting, H. Cremasco, and C. A. Conte-Junior, "The spread of the COVID-19 outbreak in Brazil: An overview by Kohonen Self-Organizing Map networks," *Medicina*, vol. 57, no. 3, pp. 1–19, 2021.

[9] P. Melin, J. C. Monica, D. Sanchez, and O. Castillo, "Analysis of spatial spread relationships of coronavirus (COVID-19) pandemic in the world using self organizing maps," *Chaos, Solitons & Fractals*, vol. 138, pp. 1–7, 2020.

[10] J. Vesanto, "Using SOM in data mining," Licentiate's thesis, Helsinki Univ. Technol., 2000.

[11] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, 2000.

[12] A. Ultsch and D. Korus, "Automatic acquisition of symbolic knowledge from subsymbolic neural networks," in *Proc. EUFIT*, vol. 1, 1995, pp. 326–331.

[13] J. Malone, K. McGarry, S. Wermter, and C. Bowerman, "Data mining using rule extraction from Kohonen self-organising maps," *Neural Comput. Appl.*, vol. 15, no. 1, pp. 9–17, 2006.

[14] W. S. Van Heerden and A. P. Engelbrecht, "HybridSOM: A generic rule extraction framework for self-organizing feature maps," in *Proc. CIDM*, 2009, pp. 17–24.

[15] W. S. van Heerden, "Self-organizing feature maps for exploratory data analysis and data mining: A practical perspective," Master's thesis, Univ. Pretoria, 2017.

[16] T. Kohonen, "Self-organizing formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[17] G. Pölzlbauer, "Advanced data exploration methods based on Self-Organizing Maps," Ph.D. dissertation, Technischen Universität Wien, 2008.

[18] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.

[19] A. Ultsch and H. P. Siemon, "Kohonen's self organizing feature maps for exploratory data analysis," in *Proc. INNC*, vol. 1, 1990, pp. 305–308.

[20] A. Varfis and C. Versino, "Clustering of european regions on the basis of socio-economic data — A Kohonen feature map approach," in *Proc. PASE*, 1991, pp. 57–68.

[21] G. Pölzlbauer, "Application of Self-Organizing Maps to a political dataset," Master's thesis, Technischen Universität Wien, 2004.

[22] J. W. Sammon, Jr, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. 18, no. 5, pp. 401–409, 1969.

[23] G. Pölzlbauer, A. Rauber, and M. Dittenbach, "A vector field visualization technique for Self-Organizing Maps," in *Proc. PAKDD*, 2005, pp. 399–409.

[24] J. H. Ward, Jr, "Hierarchical grouping to optimize an objective function," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 236–244, 1963.

[25] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. BSMSP*, vol. 1, 1967, pp. 281–297.

[26] A. Ultsch, "U*C: Self-organized clustering with emergent feature maps," in *Proc. LWA*, 2005, pp. 240–244.

[27] J. Vesanto, "SOM-based data visualization methods," *Intelligent Data Anal.*, vol. 3, no. 2, pp. 111–126, 1999.

[28] E. Pampalk, A. Rauber, and D. Merkl, "Using smoothed data histograms for cluster visualization in Self-Organizing Maps," in *Proc. ICANN*, 2002, pp. 871–876.

[29] A. Ultsch, "Maps for the visualization of high-dimensional data spaces," in *Proc. WSOM*, 2003, pp. 225–230.

[30] ——, "Pareto Density Estimation: A density estimation for knowledge discovery," in *Proc. GfKl*, 2003, pp. 91–100.

[31] G. Pölzlbauer, A. Rauber, and M. Dittenbach, "Graph projection techniques for Self-Organizing Maps," in *Proc. ESANN*, 2005, pp. 533–538.

[32] X. Zhang and Y. Li, "Self-organizing map as a new method for clustering and data analysis," in *Proc. IJCNN*, vol. 3, 1993, pp. 2448–2451.

[33] A. Ultsch, "U*-Matrix: a tool to visualize clusters in high dimensional data," Philipps-Universität Marburg, Tech. Rep. 36, 2003.

[34] D. Fasulo, "An analysis of recent work on clustering algorithms," Univ. Washington, Dept. Comput. Sci. Eng., Tech. Rep. 01-03-02, 1999.

[35] W. S. van Heerden and A. P. Engelbrecht, "A comparison of map neuron labeling approaches for unsupervised self-organizing feature maps," in *Proc. IJCNN*, 2008, pp. 2139–2146.

[36] D. W. Aha, C. L. Blake, S. J. Hettich, E. J. Keogh, C. J. Merz, and P. M. Murphy, "UCI repository of machine learning databases," 1998, univ. California, Irvine.

[37] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Springer, 1989.

[38] W. S. van Heerden and A. P. Engelbrecht, "Unsupervised weight-based cluster labeling for self-organizing maps," in *Proc. WSOM*, 2012, pp. 45–54.

[39] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM_PAK: The Self-Organizing Map program package," Helsinki Univ. Technol., Tech. Rep. A31, 1996.

[40] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[41] R. G. Miller, Jr, *Simultaneous Statistical Inference*, 2nd ed. Springer-Verlag, 1981.

[42] T. C. Siong, "A hybrid artificial neural network model for data visualisation, classification, and clustering," Ph.D. dissertation, Universiti Sains Malaysia, 2006.