Advances in Reliability, Safety and Security, Part 8 – Kołowrocki, Magryta-Mut (eds.)

Advances in Reliability, Safety and Security

ESREL 2024
Monograph Book Series

Application Of Self Organizing Map Algorithm For Analysis Of Products Quality

Doha Meslem[a], Marcin Hinz[b]

[a]*University of Wuppertal, Wuppertal, Germany*
[b]*Munich University of Applied Sciences, Munich, Germany*

Abstract

Addressing the challenge of monitoring product-conditions, particularly in cases without predefined quality criteria, necessitates image-based monitoring in the manufacturing process. The lack of target classes requires an unsupervised learning approach. This paper explores unsupervised learning through the implementation of Self-Organizing Maps (SOM) as a clustering method, performing a parameter study including data preprocessing using standardization and normalization techniques, along with tuning SOM parameters. The objective is to identify the parametric sets resulting in clusters aligned with manufacturer-specified criteria, mainly roughness values (Ra). The unsupervised machine learning algorithm is expected to analyze various parameters and produce results that align with a predefined standard (Ra), simulating a supervised approach without explicit data training but by pattern analysis. While briefly delving into the theoretical aspects drawn from published research, the paper primarily focuses on demonstrating the feasibility of the proposed methodology.

Keywords: condition monitoring, unsupervised learning, machine learning, self-organizing maps algorithm, surface topography, unsupervised neural networks

1. Introduction

With the advances happening among the various industries in terms of improving their products and production processes, there was an increasing reliance on data analysis techniques. Industries seek ways to integrate data to advance their processes and keep up with the world in relying on machine learning.

In the world of manufacturing, decisions often hinge on the fine details on the surfaces of tools. The idea is to study the hidden patterns in these surfaces using advanced machine learning techniques. Surface roughness is one of the indicative properties. As a form of quality control for everyday tools such as knives, the manufacturer uses surface roughness and specifies numerical roughness standards, outside of which are the knives considered rejects. The manufacturer categorizes the produced knives accordingly in three categories. Those categories are set as the guideline for quality-control. This production process incorporates machine learning techniques to improve it, by using the categories as the set of standards. In this study, self-organizing maps, SOM, a neural networks algorithm is used on data taken from the knives, that do not include roughness values, and are compared to the set of standards, in the effort of producing results that mimic the standard as much as possible. The process followed by this paper has relevance beyond knives, however. It ensures reproducibility by integrating machine learning techniques into production processes in general. The paper is divided into two main parts, the theoretical; which mainly discusses SOM and practical aspects; describing the feasibility of SOM.

2. Literature Review

The general theory behind SOM and the application of algorithms is saturated in literature such as Kohonen (1990), Rahaman et al. (2021), Zietsman and Vuuren (2022), and Fu et al. (2023), respectively to name a few.

Existing literature, as shown by Tasdemir and Merenyi (2009) in their article discussing topology in SOM, predominantly emphasizes on the mapping and visualization aspect of Self-organizing maps. Self-organizing maps being primarily a dimension reduction algorithm, explains why this is necessary, however this has left a gap in literature, whereas to explore SOM's full potential, more literature should be done on the other aspect of it, clustering.

Clustering in neural network is a topic explored often by research, however as inferred from Bock (1998) and Shah and Murtaza (2000) for example, it is rarely SOM's clustering, but rather other algorithms. Moreover, whenever SOM is studied in clustering, most research is hybrid approaches, that can be divided into two main approaches. Either a combination of SOM and other clustering algorithms; Li et al. (2013) in a semi supervised approach to visualize SOM, using k-means to cluster, and Chakraborty et al. (2015) hybrid's technique using additionally fuzzy algorithm; which still doesn't explore SOM's clustering abilities and focuses on SOM's mapping strength, or combinations as to enhance the expected results; Li and Pan (2013a) with the introduction of an improved SOM, Isa et al. (2009) in an enhanced hybrid approach, and Jin et al. (2004) in an enhanced algorithm named: Expanding SOM.

There exists literature that deals with neural networks and unsupervised machine learning algorithms on surface roughness. Pontes et al. (2009) for example, in their article, artificial neural networks on surface roughness modeling. However, exploring this literature reveals that it is done using other algorithms such as, but not limited to, k-means. Such as done in Qin et al. (2018) and Castilho et al. (2023) leading to the often-mentioned clustering of the SOM and not using the SOM.

While surface roughness using SOM has not been fully explored, SOM research was still well founded in many applications. The versatility of SOM has allowed for that. Applications include Roden et al. (2015) in geology and seismic attributes, Schneider et al. (2019) in drug discovery, Hsu et al. (2009) in stock price forecasting, Xiao et al. (2003) in gene clustering and Cassano et al. (2015) in weather temperature analysis. Diving into hybrid clustering approaches and the various applications, the practical significance becomes evident, especially in terms of production processes and quality control. Quality control in various aspects is well studied, however the introduction of neural networks in this field is mostly limited to Convolutional neural networks as shown by Azizah et al. (2017), Mamaeva et al. (2022), and Adibhatla et al. (2020) to name a few, with a noticeable number of literature on Artificial neural networks Wang et al. (2019). Where Quality control using SOM was studied, in cases presented by Li and Pan (2013a) and Li and Pan (2013b), it was dealt with in hybrid scenarios as previously discussed, with addition of k-means by the former, and the application of an enhanced SOM model by the latter. And while SOM was sometimes used for production processes-monitoring, as mentioned by AlHoniemi et al. (1999), it was done in a way different to what this paper aims to cover.

As shown, in the exploration of neural network clustering or the implementation of quality control checks, a variety of algorithms have been prevalent. This study aims to exclusively employ SOM, highlighting its distinctive capabilities. The focus is specifically on a standalone application of SOM, as opposed to hybrid methodologies. The scope of application extends to surface roughness, an aspect that has received limited attention in the context of SOM applications. The main goal however is the reproducibility of the results beyond surface roughness, as a quality control measure across the spectrum of applications, using an extensive parametric study.

3. Machine learning in context

Machine learning, a subset of artificial intelligence, deals with applying algorithms on data, as to train and test data in one of the three main methods, unsupervised, supervised and reinforcement learning. Unsupervised learning deals with unlabelled data, where algorithms are applied as to discover the data's hidden patterns as opposed to training the results into an expected output. (Holmes and Jain (2006))

Deep learning is a class of machine learning and is the scope of this paper. Deep learning is based on neural networks. Which resemble the human brain neurons in terms of complexity and layers, thought of as to mimic the signals transfer between neurons. Artificial neural networks consist of an input layer, (usually) one hidden layer or more, and an output layer. As is the case with all methods, neural network types are numerous. However, the main algorithms are Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), containing convolutional layers and neurons, used mainly for computer vision and pattern recognition, and Recurrent Neural Networks (RNN) used for time series data, and future predictions. The main types are feed-forward neural networks: processing data in one direction, backpropagation: with continuous learning using corrective feedback loops, and convolutional neural networks: the hidden layers carry out mathematical operations known as convolutions. These are trained using the supervised, unsupervised and reinforcement approaches. (Alom et al. (2019), Guo et al. (2016))

SOM is a dimensionality reduction-clustering unsupervised neural network approach. This algorithm uses a type of competitive learning process. SOM consists of an input layer and an output layer, where the input layer is the data with its dimensions, and the output layer is the representative layer, usually in 2D. This ensure a dimension reduction, with topological preservation of the relationships of the input data, where similar input patterns are mapped onto neighbouring locations on the output grid, called the map. Where the output layer representatives are called nodes. The nodes are arranged in proximity, with the degree of closeness reflecting their similarity to each other. The SOM learning process is deterministic, leading to reproducible results.

Each input dimension is connected to all output nodes by specific weights to each. This means that each output node has a vector of weights from all input dimensions, acting as its coordinates. Each of the weights in the vector will have coordinates for the dimension and for the datapoint. The output nodes have a place in the space, and this place is adjusted throughout as will be explained. After that, for each datapoint of the input dimensions, the algorithm tries to find the closest node to it. This happens as follows: each datapoint has a specific value in each of the dimensions, for each of the nodes there is also a specific weight for each of the dimensions, the distance between these points is found for each of the nodes using the Euclidean distance. The node that has the smallest distance is hence called the best matching unit for this datapoint.

$$Distance = \sqrt{\sum_{i=0}^{i=n}(V_i - W_i)^2} \quad (1)$$

V: input vector values, W: weights, N: number of dimensions

Now, the adjusted SOM must modify its weights to achieve an even greater proximity to this datapoint in the dataset. This adjustment is because only output nodes can be updated. The map gets closer to the datapoint by shifting best matching unit towards it, leading to eventually a map that's aligned with the dataset. As it shifts towards the datapoint, the neighbouring nodes move along with it. This allows for clustering the data.

Then, the algorithm draws a radius around the best matching unit using equation (2), causing all nodes within this vicinity to update their weights, to allow for the shift, and closer neighbouring nodes have heavier weights updated, hence preserving the topology. This is where the cooperation process is applied and is done using the weight updating equation (3).

$$\sigma(t) = \sigma_0 \, exp\left(-\frac{t}{\lambda}\right) \quad (2)$$

σ_0 : Width of lattice at time t0, $\sigma(t)$: width of lattice at time t, t: time, λ: time constant

$$W(t+1) = W(t) + \varphi(t)L(t)\big(V(t) - W(t)\big) \quad (3)$$

$W(t+1)$: The new weight

$$L(t) = L_0 exp\left(-\frac{t}{\lambda}\right) \quad (4)$$

L(t): Learning rate at t, L_0 : learning rate at t0

$$\varphi(t) = exp\left(-\frac{Distance^2}{2\sigma^2(t)}\right) \quad (5)$$

$\varphi(t)$: Influence rate

The map here changes slightly as to align. Now that there are new weight vectors (no need for weight initialization), this is then repeated for every data point, until the map aligns completely with the input data, meaning convergent, or till a stopping criterion is reached. (Van Hulle (2012), Kohonen (2013), Cottrell et al. (2018), Kohonen (1990), Ponmalai and Kamath (2019), Natita et al. (2016), Ng and Chan (2019)).

4. Operational framework

In the practical aspect discussed in the following section of the paper, the study of SOM pursues two main goals. Firstly, it involves an extensive, methodical exploration of various parameter setups to determine the most efficient set. Secondly, it includes a quantitative assessment aimed at comparing the results achieved.

4.1. Data source

The dataset is multivariate, featuring 42 variables, and is structured and numerical in nature. This dataset originates from three distinct knife types, provided by the manufacturer. The dataset is divided into two main components: optical and mechanical. The optical dataset consists of various line and optical measurements, analyzed through computer vision. The optical measurements are confined to specific regions of the surfaces, as

illustrated in Fig. 1, where all images are uniformly cropped, and the reference section of the knives is clearly identified (Hinz et al., 2019). Whereas the mechanical dataset encompasses parameters such as color, roughness, and gloss values.

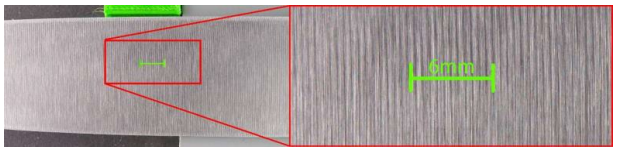

Fig. 1. Knife-image taken by the camera.

The objective is to compare the mechanical data with the optical data to explore potential correlations. Among the three categories of target variables, emphasis is placed on roughness, denoted by the Ra value. This value is calculated using Equation (6) (Sancaktar and Gomatam, 2001):

$$R_a = \left(\frac{1}{L_m}\right) \int_{x=0}^{x=L_m} \lceil y \rceil dx \quad (6)$$

Where Lm is the total scanned length in the x direction.

4.2. Data parameterization and method implementation

The scope of this research is an extensive parametric study to explore both clustering and mapping capabilities of SOM, as to form reproducible results for future quality control and a variety of applications. For this purpose, the Minisom implementation of this algorithm is used. This was chosen based on its extensive presence in research (Fu et al. (2023), Rahaman et al. (2021), and Muhammad et al. (2022)).

The initial step involves preprocessing the mechanical data. This entails employing two standardization techniques, one normalization technique, and a combination of both. The sequence of combining these techniques is varied, resulting in seven distinct combinations of preprocessed mechanical data, while one dataset remains untreated.

Then, the following parameters where tuned: number of columns and rows, sigma, learning rate, neighborhood function, topology, and the number of iterations. The topographies of the output layer varied between rectangular and hexagonal in 2D. The learning rate explained previously was varied between 0.1, 1, 2.5, 4 and 5 while the number of iterations was changed between 5, 50, 400, and 1000. These selections for those two parameters are made to encompass a broad spectrum, examining the patterns they follow and exploring the tendencies. After going through multiple runs of the algorithm on various segments of the data with diverse parameter values, it was concluded that these are satisfactory. For the neighborhood functions, the four possible types are tested, namely: gaussian, Mexican hat, bubble, and triangle. Each of those functions is described by its shape, acting as the de facto 'radius' around the winning neuron, dictating the shape within which the neurons receive weights as to update them. (31, 32)

The sigma values for gaussian and Mexican hat however differ from those used for bubble and triangle. This is dictated by the minisom as described in its documentation. Moreover, while for the former two, values: 0.1, 0.5, 0.8, 1, and 1.5 are used, values 1, 2, 3, and 4 are used for the latter two. The rationale behind this is that for the latter two, values less than one and non-integers are not allowed. Additionally, for all values, whenever the sigma value equals that of the column or row, the sigma is deemed too high for the dimension of the map. Using decimals less than one for gaussian and Mexican hat was done with the purpose of exploring the behaviour with low shifts and because based on separate several runs of the algorithm, these values seemed to have an impact on the outcome. The number of rows and columns is varied between 1, 2, 3, and 4. SOM cluster data based on its output map, hence not allowing for a pre-tuning of the number of clusters. For the purposes of this study however, it is necessary to limit the number of clusters to exactly three, as to allow for a compatibility study between these and the clusters stated by the manufacturer or standardized clusters which are also three. After the data is preprocessed, it goes through the algorithm where all the parametric combinations are run. Each set of parametric combinations is run and fitted to form a map that best represents it. The algorithm further clusters based on the map, resulting in different number of clusters each run. Whenever this number is three, it is counted into the study, all other cluster numbers and their parametric settings are discarded in this study.

Clusters are fitted to correspond to an array of 0, 1, or 2, indicating the cluster each datapoint belongs to. The paper's primary focus is the roughness value, classified into three classes. This classification is based on the manufacturer's criteria, which provide a set of roughness values with an upper and lower limit. Roughness values outside these limits are deemed useless by the manufacturer and thus avoided in knife production. Consequently, the clustered data are set to be compared against these limits. The number of clusters is selected to match the

number of roughness value classes set by the manufacturer, and the order of the clusters must align with the roughness order.

The SOM algorithm labels clusters arbitrarily but consistently, so the cluster order might need reassignment. This is achieved by renaming the clusters based on the roughness values in each cluster. Additionally, the mean roughness value for each cluster is calculated. Clusters are then renamed in ascending order of their roughness mean: the lower the mean, the lower the cluster number. With the clusters relabeled, they are ready for comparison to assess how well the clustering method aligns with the manufacturer's criteria. The algorithm's efficiency is evaluated using various metrics, with overall efficiency being paramount. The most efficient parameter set is chosen for analyzing the surface topographies of knives within this criterion and production process.

As demonstrated earlier, the data has no labels and is trained through unsupervised learning techniques. But by comparing it to some prescribed set of data and expecting it to comply with it, by calculating the efficiency, and 'training' it, by setting a SOM-parameter analysis, this study mimics a supervised machine learning method. Consequently, this investigation emulates aspects of supervised machine learning, thus transforming an initially unsupervised method into one that, to some extent, mirrors the characteristics of supervised machine learning.

5. Discussion of results

It is *crucial* for the understanding of the results to note that, as explained in data parameterization, there are many variables. Moreover, since that in this algorithm the number of clusters cannot be predefined while tuning, all the parametric setting-combinations are run, and the ones resulting in clusters more or less than three are discarded. This means that for each of those parameters, the chosen options have a different number of runs (model numbers). Meaning they cannot be directly compared to each other. Additionally, the chosen sigma values used for the chosen neighbourhood functions are not the same for all of them. This leads to lesser control variables such as model number and sigma values, hence preventing direct comparability. This is further illustrated by figure 2. Additionally, the conclusions drawn from results are based on the efficiency values collected by the runs and not from such graphs, the trends were also studied separately.

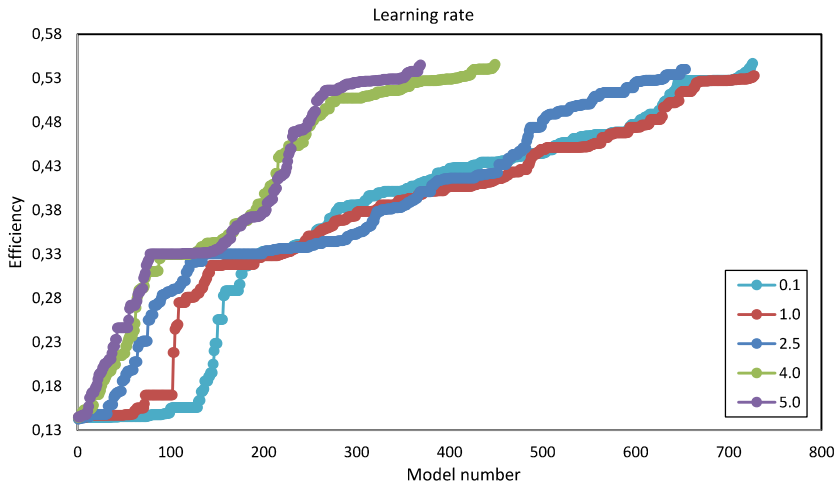

Fig. 2. Learning rates efficiencies and model numbers.

This set of learning rates was used for all parameters in all their combinations, meaning it could be considered as a control variable for the combinations. Even with this being the case, five options result in variant model numbers. Moreover, the model numbers for each of the five rates was inversely proportional to the learning rate. This requires a comparison of the model numbers that achieved success and convergent. This highlights the notion that the model number is related to the success of this parameter choice and combination, hence a form of efficiency-testing. Moreover, parameters with a higher model number have lesser parametric combinations discarded, meaning they have more models that fulfilled the requirements within the context already described. Trends can therefore not be inferred from the graph. Consequently, conclusions for all the parameters and their

model numbers were however analysed from the efficiencies and discussed in the following sections. Trends, number of models and efficiencies are analysed and compared below.

For learning rates, the higher rates of 4 and 5 behaved similarly and rates 2.5, 0.1 and 1 showed similar trends. All rates almost stabilize at 33% efficiency, meaning a considerable number of parametric combinations attain this percentage. The highest learning rate 5 has the lowest number of combinations that are under 33%, while 2.5 was the most spread. The lowest efficiency achieved was attained by rate of 1, which was 1.38% less the maximum, achieved by learning rate: 0.1.

For the topology parameter, both types replicate each other. The rectangular map shape had slightly more runs for all parametric combinations and three clusters. The Hexagonal shape has attained a maximum efficiency of 0.162% more than that of the rectangular one. But occasionally achieved slightly (around 1%) lower efficiencies than rectangular shape, at around the efficiencies of 35%.

While all combinations of neighborhood functions cover the full range of efficiency, bubble has achieved the lowest efficiency with 0.66% less than the overall maximum achieved by Mexican hat. As expected, bubble and triangle behave like each other as do gaussian and Mexican hat. However, Mexican hat, having the second highest number runs, amounts to around 50% of those achieved by gaussian. Triangle has the least number of combinations under 35% percent. The three other functions stabilize at around 33%. The triangle function as shown has more spread over the efficiencies. Overall, gaussian and Mexican hat excel in efficiencies with learning rates higher than 2.5, lower sigma values (less than 1) and lower iterations numbers.

As sigma values are coupled with the neighborhood functions, they behave in accordance with the values used for the neighborhood functions. Meaning, 2 and 3 behave similarly. however as can be noticed, 1.5 also behaves like those. While values 0.5 and 0.8 behave like each other. sigma of value 1 shows a tendency to the middle. Sigma value 3 has the fewest runs and it has the highest initial value in comparison to all other combinations, of about 5% more than the lowest. It however also gives the lowest overall efficiency. Sigma values of 0.1 show steeper changes in efficiencies with many percentage skips throughout, because it covers the full range of efficiencies. Sigma value 4 didn't exist and 58% of the sigma value 1 used were combined with bubble and triangle.

Five iterations have the highest number of runs similar in behaviour to 400 iterations, with 1000 and 50 behaving similarly. The 1000 iterations had the largest number of runs under 20% efficiency in comparison to the others, with 400 having the lowest, percentagewise. Under 33% all values behave similarly, due to combinations including mainly Mexican hat and gaussian, and lower sigma values. 1000 iterations show the lowest highest efficiency of all the options.

While all pre-processing techniques cover the whole range of efficiency, there are noticeable discrepancies. Mixed techniques excel, but at the trade-off of 1% and the correct parametric combinations: 50 iterations, 0.1 learning rate, sigma equals to 1, gaussian neighborhood function and 2*4 map-shape, unpreprocessed data could be used. The lowest highest efficiency was achieved by un-preprocessed data. Normalization-standarization (NS) has the least number of lowest efficiencies achieved with its parametric combinations, with less than 5% of its run attain under 30% efficiency, it has a good spread an achieves a maximum of just 0.08% less than the overall achieved maximum, making NS an attractive option.

Shape affected number of clusters mainly. An observation of that shows that 50% of the total iterations were represented in maps on 1*3 ad 3*1. While only 20% of the combinations were 3*3, 4*4, and 2*2.

The highest 4% of the efficiency attained is achieved by predominantly the lowest numbers of iterations, a combination of learning rates and sigmas, and mainly preprocessed data by more than one method each (standardization, normalization, normalization-standardization, and minmax-standardization). Efficiencies vary with those parameters largely, with the lowest being around 40% less than the highest overall attained efficiency. Un-preprocessed data predominates in the middle to low efficiency ranges, around the 30%. More than 60% of the data that was compatible was of the gaussian function, with its combinations present all over the efficiency spectrum (lowest and highest). 24% of all the parametric combinations achieve the highest 4% of efficiency. Bubble and triangle functions are the least represented of all the functions, because of the sigma and map limitations. They however outperform the other algorithms over all in efficiencies. Combinations with mid-high learning rate (2.5) achieve lower efficiencies in their combinations as compared to combinations of lower learning rates. The lowest 5% of the efficiencies attained were achieved by combinations with the highest numbers of iterations.

The highest efficiency was achieved by preprocessing standardization followed by normalization, 400 iterations, sigma = 1, hexagonal, learning rate of 0.1, Mexican hat as the neighborhood function, and a shape of 4*4. With 54,65%, the highest 0.6% of all efficiencies were achieved by Mexican hat combinations. It is important to note that within this parametric study, it isn't possible to infer one overall parameter that has the overall control over the results. All parameters here except topography have an impact on the efficiency. This is partly because as previously discussed, most parameters are intertwined.

Figures 3 a-d show a visualization of the most efficient parametric setting achieved by this study on a self-organizing map of 24*24 since the cluster number was already ensured to be three. This is done as to allow for a clearer visualization of the mapping. Figures 3b, c, and d show mappings of the 42 dimensions reduced to two for datapoints that represent roughness (Ra3) classes 0, 1 and 2 respectively. Moreover, each graph shows where the datapoints relating to one class would be mapped if they were filtered and mapped alone without the influence of the other Ra3 classes. Figure 3a shows how the SOM algorithm maps all these datapoints together. Datapoints do not map in figure 3a the way the map separately because of the influences of other datapoints as mentioned earlier in the discussion of SOM causing shifts in specific positions. The overall mapping of all classes at certain positions on the graph shows the datapoints most represented there. At points that are red for example, there could exist other datapoints (blue or green), but the colour code shown is the one represented the most at this position (more red datapoints are placed here than other colours). Shape discrepancies such as coloured corners reveal that another colour also exists at this position but is less represented in quantity. Another discrepancy revealed upon closer inspection is datapoints represented with slight corners as opposed to dots, this insinuates to more of the same colour datapoints at that position, forming a squarer shape.

As can be inferred, the data is heterogenous, and shows a lack of distinguishability between classes. This consequently poses as an obstacle to yield higher efficiencies according to the predefined classes by the manufacturer. The SOM reduces the dimensions by mapping differently from the manufacturer set-quality criteria, that is. As intended by the study, once the mapping is achieved, the algorithm further clusters the data in a manner shown by unsupervised learning algorithms such as K-Means and Gaussian Mixture Models (GMM) (Hinz et al. (2022a, 2022b)).

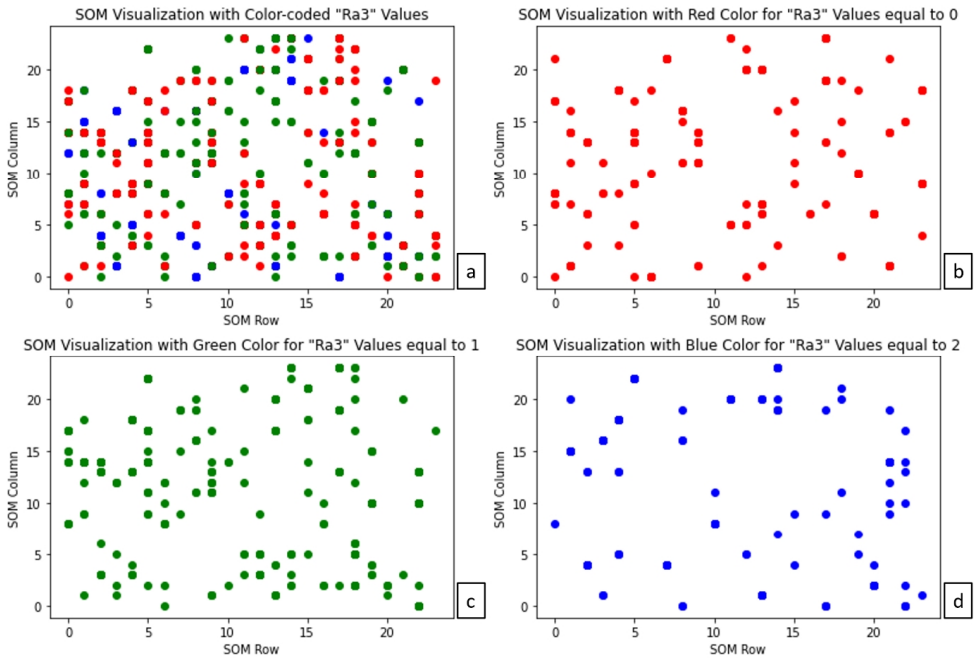

Fig. 3 a-d: Roughness values based-mapping, a: SOM and all roughness classes, b: SOM and roughness class 0, c: SOM and roughness class 1, d: SOM and roughness class 2.

6. Conclusion

In conclusion, this paper studies the SOM unsupervised machine learning algorithm by application on data derived from the surfaces of three knife types provided by the manufacturer. Data extracted using computer vision undergoes clustering with the help of a neural network algorithm and is subsequently compared with specified roughness values provided by the manufacturer. These predefined criteria serve as determinants

regarding the knives' usability. The model examines surface topographies and assesses their quality, using a parameter-study that explores eight varying parameters. The outcomes of each parameter set are compared against the manufacturer's set upper and lower limits, and the efficiency of each set is calculated and discussed.

Looking ahead to future research, a comprehensive analysis of diverse unsupervised algorithms will be undertaken and thoroughly examined. Through the evaluation of multiple algorithms, the overall efficiencies, as well as the identification of the optimal algorithm for conducting similar analyses with similar sets of data will be determined. These advancements aim not only to expand our understanding of unsupervised machine learning applications in surface quality analysis but also to provide reproducible methodologies accessible to other manufacturers, contributing to the broader progress of this field.

References

Adibhatla, V. A., Chih, H. C., Hsu, C. C., Cheng, J., Abbod, M. F., Shieh, J.-S. 2020. Defect detection in printed circuit boards using you-only-look-once convolutional neural networks. Electronics 9(9), 1547.

Alhoniemi, E., Hollmén, J., Simula, O., Vesanto, J. 1999. Process monitoring and modeling using the self-organizing map. Integrated Computer-Aided Engineering 6(1), 3–14.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A., & Asari, V. K. 2019. A state-of-the-art survey on Deep Learning Theory and Architectures. Electronics 8(3), 292.

Azizah, L. M., Umayah, S. F., Riyadi, S., Damarjati, C., Utama, N. A., 2017. Deep learning implementation using convolutional neural network in mangosteen surface defect detection. 7th IEEE International Conference on Control System, Computing and Engineering, 242-246.

Bock, HH. 1998. Clustering and Neural Networks. Rizzi, A., Vichi, M., Bock, HH. (Eds) Advances in Data Science and Classification Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin, Heidelberg.

Cassano, E., Glisan, J., Cassano, J., Gutowski, W., Seefeldt, M. 2015. Self-organizing map analysis of widespread temperature extremes in Alaska and Canada. Climate Research 62(3), 199–218.

Castilho, V. M., Balthazar, W. F., da Silva, L., Penna, T. J. P., Huguenin, J. A. O. 2023. Machine learning classification of speckle patterns for roughness measurements. Physics Letters A.

Chakraborty, B., Menezes, A., Dandapath, S., Fernandes, W. A., Karisiddaiah, S. M., Haris, K., Gokul, G. S. 2015. Application of hybrid techniques (self-organizing map and fuzzy algorithm) using backscatter data for segmentation and fine-scale roughness characterization of seepage-related seafloor along the western continental margin of India. IEEE Journal of Oceanic Engineering 40(1), 3–14.

Holmes, D., Jain, L. 2006. Innovations in Machine Learning : Theory and Applications. Springer.

Sancaktar, E., Gomatam, R. 2001. A Study on the Effects of Surface Roughness on the Strength of Single Lap Joints. Journal of Adhesion Science and Technology 15, 97-117.

Fu, Q., Li, Y., Albathan, M. 2023. A supervised method to enhance distance-based neural network clustering performance by discovering perfect representative neurons. Granular Computing 8(5), 1051–1065.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M. S. 2016. Deep Learning for Visual Understanding: A Review. Neurocomputing 187, 27–48.

Hinz, M., Meslem, D., Bracke, S. 2022a. The application of the Gaussian Mixture Model algorithm for the unsupervised analysis of surface topographies. Probabilistic Safety Assessment and Management PSAM 16.

Hinz, M., Meslem, D., Bracke, S. 2022b. The application of the K-means algorithm for the unsupervised analysis of surface topographies. Proceedings of the 32nd European Safety and Reliability Conference ESREL2022.

Hinz, M., Radetzky, M., Guenther, L. H., Fiur, P., Bracke, S. 2019. Machine Learning Driven Image Analysis of Fine Grinded Knife Blade Surface Topographies. Procedia Manufacturing 39, 1817-1826.

Hsu, S. H., Hsieh, J. P. A., Chih, T. C., Hsu, K. C. 2009. A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. Expert Systems with Applications 36(4), 7947–7951.

Isa, D., Kallimani, V. P., Lee, L. H. 2009. Using the self organizing map for clustering of text documents. Expert Systems with Applications 36(5), 9584–9591.

Jin, H., Shum, W. H., Leung, K. S., Wong, M. L. 2004. Expanding self-organizing map for data visualization and cluster analysis. Information Sciences 163(1–3), 157–173.

Kohonen, T., 1990. The self-organizing map. Proceedings of the IEEE 78(9), 1464-1480.

Kohonen, T. 2013. Essentials of the self-organizing map. Neural Networks 37, 52–65.

Li, W., Zhang, S., He, G. 2013. Semisupervised distance-preserving self-organizing map for machine-defect detection and classification. IEEE Transactions on Instrumentation and Measurement 62(5), 869–879.

Li, Y., Pan, F. 2013a. Application of improved SOM neural network in manufacturing process quality control. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

Li, Y. B., & Pan, F. 2013b. Study on the combination of SOM and K-means algorithms in manufacturing process quality control. Applied Mechanics and Materials, 427–429.

Cottrell, M., Olteanu, M., Rossi, F., Villa-Vialaneix, N. N. 2018. Self-Organizing Maps, theory and applications. Revista de Investigacion Operacional 39 (1), 1-22.

Mamaeva, A., Krasnova, O., Khvorova, I., Kozlov, K., Gursky, V., Samsonova, M., Tikhonova, O., Neganova, I. 2022. Quality control of human pluripotent stem cell colonies by computational image analysis using Convolutional Neural Networks. International Journal of Molecular Sciences 24(1), 140.

Muhammad, G., Saeed, U., Islam, N., Kumar, K., Hussain, F., Khurro, M. A., Shaikh, A. A., Ali, I. 2022. GVDeepNet: Unsupervised deep learning techniques for effective genetic variant classification. Pakistan Journal of Engineering and Technology 5(1), 16–22.

Natita, W., Wiboonsak, W., Dusadee, S. 2016. Appropriate learning rate and neighborhood function of self-organizing map (SOM) for specific humidity pattern classification over southern Thailand. International Journal of Modeling and Optimization 6(1), 61–65.

Ng, S., Chan, M. 2019. Effect of neighbourhood size selection in som-based image feature extraction. International Journal of Machine Learning and Computing 9(2), 195–200.
Ponmalai, R., Kamath, C. 2019. Self-Organizing Maps and Their Applications to Data Analysis.
Pontes, F. J., Ferreira, J. R., Silva, M. B., Paiva, A. P., Balestrassi, P. P. 2009. Artificial neural networks for machining processes surface roughness modeling. The International Journal of Advanced Manufacturing Technology 49(9–12), 879–902.
Qin, L., Yi, Z., Zhang, Y. 2018. Unsupervised surface roughness discrimination based on bio-inspired artificial fingertip. Sensors and Actuators A: Physical 269, 483–490.
Rahaman, S., Samuel, R., Neamtiu, I. 2021. Quantifying Nondeterminism and Inconsistency in Self-organizing Map Implementations. IEEE International Conference on Artificial Intelligence Testing (AITest), 85-92.
Roden, R., Smith, T., Sacrey, D. 2015. Geologic pattern recognition from seismic attributes: Principal component analysis and self-organizing maps. Interpretation 3(4).
Schneider, P., Tanrikulu, Y., Schneider, G. 2009. Self-organizing maps in drug discovery: Compound Library design, scaffold-hopping, repurposing. Current Medicinal Chemistry 16(3), 258–266.
Shah, J., Murtaza, M. 2000. A Neural Network Based Clustering Procedure for Bankruptcy Prediction. American Business Review 18(2).
Tasdemir, K., Merenyi, E. 2009. Exploiting data topology in visualization and clustering of self-organizing maps. IEEE Transactions on Neural Networks 20(4), 549–562.
Van Hulle, M. M. 2012. Self-organizing Maps. Rozenberg, G., Bäck, T., Kok, J.N. (Eds), Handbook of Natural Computing. Springer, Berlin, Heidelberg.
Wang, Z., Feng, J., Fu, Q., Gao, S., Chen, X., & Cheng, J. 2019. Quality control of online monitoring data of air pollutants using artificial neural networks. Air Quality, Atmosphere & Health 12(10), 1189–1196.
Xiao, X., Dow, E. R., Eberhart R., Miled, Z. B., Oppelt, R. J. 2003. Gene clustering using self-organizing maps and particle swarm optimization. *Proceedings International Parallel and Distributed Processing Symposium,* 10.
Zietsman, H., Vuuren, J. 2022. A generic framework for decision support in retail inventory management.