electronics

Article

A Comparative Analysis of Machine Learning Algorithms for Identifying Cultural and Technological Groups in Archaeological Datasets through Clustering Analysis of Homogeneous Data

Maurizio Troiano [1], Eugenio Nobile [2], Flavia Grignaffini [1], Fabio Mangini [1], Marco Mastrogiuseppe [3], Cecilia Conati Barbaro [4] and Fabrizio Frezza [1,*]

[1] Department of Information Engineering, Electronics and Telecommunications (DIET), University of Rome "La Sapienza", 00185 Rome, Italy; maurizio.troiano@uniroma1.it (M.T.); flavia.griffagnini@uniroma1.it (F.G.); fabio.mangini@uniroma1.it (F.M.)

[2] The Sonia and Marco Nadler Institute of Archaeology, Tel Aviv University, Tel Aviv 6997801, Israel; eugenion@mail.tau.ac.il

[3] Department of Human Sciences, Link Campus University, 00165 Rome, Italy; marco.mastrogiuseppe@uniroma1.it

[4] Department of Sciences of Antiquities, University of Rome "La Sapienza", 00185 Rome, Italy; cecilia.conati@uniroma1.it

* Correspondence: fabrizio.frezza@uniroma1.it

Abstract: Machine learning algorithms have revolutionized data analysis by uncovering hidden patterns and structures. Clustering algorithms play a crucial role in organizing data into coherent groups. We focused on K-Means, hierarchical, and Self-Organizing Map (SOM) clustering algorithms for analyzing homogeneous datasets based on archaeological finds from the middle phase of Pre-Pottery B Neolithic in Southern Levant (10,500–9500 cal B.P.). We aimed to assess the repeatability of these algorithms in identifying patterns using quantitative and qualitative evaluation criteria. Thorough experimentation and statistical analysis revealed the pros and cons of each algorithm, enabling us to determine their appropriateness for various clustering scenarios and data types. Preliminary results showed that traditional K-Means may not capture datasets' intricate relationships and uncertainties. The hierarchical technique provided a more probabilistic approach, and SOM excelled at maintaining high-dimensional data structures. Our research provides valuable insights into balancing repeatability and interpretability for algorithm selection and allows professionals to identify ideal clustering solutions.

Keywords: machine learning; clustering analysis; classification; archaeology; neolithic

check for updates

Citation: Troiano, M.; Nobile, E.; Grignaffini, F.; Mangini, F.; Mastrogiuseppe, M.; Conati Barbaro, C.; Frezza, F. A Comparative Analysis of Machine Learning Algorithms for Identifying Cultural and Technological Groups in Archaeological Datasets through Clustering Analysis of Homogeneous Data. *Electronics* **2024**, *13*, 2752. https://doi.org/10.3390/electronics13142752

Academic Editor: Ping-Feng Pai

Received: 6 June 2024
Revised: 5 July 2024
Accepted: 9 July 2024
Published: 13 July 2024

1. Introduction

In recent years, artificial intelligence (AI), including both machine learning (ML) and deep learning (DL), has found various applications in archaeology. These applications include identifying the locations and extensions of archaeological sites, mapping the dispersion of artifacts within sites [1–6], and taxonomic/typological artifact identification [7–9] or metrics prediction for broken regular or standardized archaeological artifacts [10,11]. The diverse nature of archaeological artifacts, encompassing typological and technological variations, presents significant potential for applying these methodologies. This potential is further amplified by the wide range of algorithms available within AI, offering flexibility in addressing the multitude of parameters and variables inherent in archaeological studies.

Clustering, or cluster analysis, is an unsupervised ML technique to find similar data structures within a dataset. Clustering algorithms exploit the underlying structure of the data and define rules to group data with similar features, partitioning the dataset according to clustering criteria without any prior knowledge of the dataset itself. In an ideal scenario, each cluster consists of data instances more related to each other than objects belonging to different clusters. The following subsections present the most used clustering algorithms for data analysis that are suitable for archaeological datasets.

Clustering analysis through an ML or DL approach has lately been applied to archaeology in study cases that often involved exclusively the fuzzy algorithm [12–16]. Other applications are usually combined with different techniques, such as 2D shape elaboration or 3D scanning technologies, to detect clusters based on silhouette affinities [17,18] or combined with aerial scanning to detect any heavy change in site morphology due to looting behaviors or destruction of archaeological sites [19,20].

In this comprehensive case study, we employed three distinct algorithms—the Self-Organizing Map (SOM), hierarchical clustering, and K-Means clustering—to demonstrate their unique abilities in decoding a uniform archaeological dataset, advantages, and limits. We also aimed to gain a deeper understanding of the potential sub-cultural aspects associated with these sites by applying these algorithms, considering the technological and typological characteristics of the lithic assemblages. The dataset primarily comprises a uniform lithic assemblage derived from Middle Pre-Pottery B Neolithic sites in various Southern Levant regions, including Nahal Yarmuth 38, Motza, Yiftahel, and Nahal Reuel.

2. Materials and Methods

The objective of this study is to present a comparative analysis between three different algorithms in a neural network (NN) environment to demonstrate the most efficient algorithm and set of parameters that can decode a complex archaeological dataset made of various types of variables for each artifact within a diverse archaeological assemblage (based on both qualitative and quantitative characteristics), and that can make clear distinctions among different technological and/or sub-cultural groups within the same chrono-cultural period of the selected sites. The artifact selection was made according to a random-stratified method within a lithic assemblage (so-called chipped stone industry). This allowed us to obtain any possible information concerning the typology and technology of each site the artifacts belong to [21–23]. The analysis comprised at most 147 variables for each artifact, both qualitative and quantitative, and pertained to the entire tool creation process, from raw materials selection to the shaping of blanks for tool creation and reuse [24–32] (less relevant variables were not used for this study). One thousand nine hundred fifty artifacts were analyzed for this purpose. The expectations are to find different clusters within the selected sites that denote different steps of the chain of operations that archaeological products represent in a lithic assemblage from a technological point of view and at least a cluster that diverges from a typological point of view among the sites. The variables related to the site of the artifacts, including geographical, geomorphological, and phytogeographical characteristics, as well as climatic and chronological information, were not included in the input for the NN. This deliberate omission was intended to avoid any predetermined clustering. The artifacts belong to the middle phase of Pre-Pottery B Neolithic (10,500–9500 cal B.P) sites such as Nahal Yarmuth 38, Motza, Yiftahel, and Nahal Reuel (Israel). All the artifacts were chosen from undisturbed layers/loci unaffected by earlier or later materials (Nahal Yarmuth's materials belong to MPPNB structures; Motza's materials come from area B-10 (rectangular domestic structure); Yiftahel's materials come from areas E20–E23 and F20–F22, which belong to layers C2 and C3, the deepest MPPNB layers from stratum IV of area E; and the sample from Nahal Reuel includes domestic structures (I, II, and IV) and open areas (III, IX, and XII), as well as knapping areas (III and X).

Three distinct algorithms/techniques were chosen for the comparative analysis: SOM, K-Means algorithm, and hierarchical clustering. Each technique demonstrated varying efficacy in decoding the archaeological dataset, resulting in different cluster outcomes.

All techniques were assessed under identical conditions. The comparison entailed two tests, each with 36 and 20 clusters for every method. These cluster numbers were deemed most pertinent for the dataset utilized in this investigation. The K-Means and hierarchical techniques were compared using silhouette analysis, while the SOM network used neighbor weight distance and hits analysis. The silhouette, weight distance, and hits analyses must be designated as valuable tools for the final archaeological interpretation.

The archaeological perspective is necessary to evaluate these algorithms' efficiency and methodology. All the tests are indeed interpreted from an archaeological point of view, specifically concerning the technological and typological meaning of relevant clusters. Technology and typology indeed represent two significant aspects in lithic studies, enabling the possibility to deepen the comprehension of the chain of operations that led up to the production of final artifacts (so-called tool, due to the presence of "retouch"), such as arrowheads, sickle blades, scrapers, etc. Each tool represents the final stage of a long operational chain that starts with selecting and reducing a specific raw material source, which is chert/flint in this case. The artifact production passes through different steps. First, the core reduction produces mainly laminar blanks and, secondarily, flakes. The Levantine neolithic, specifically the Middle Pre-Pottery B phase, has a laminar-oriented production (primarily blades and secondarily bladelets) all over the Levant. This shared cultural aspect highlights a so-called cultural koiné [24–32]. The archaeological check and interpretation are therefore fundamental not only to validate the analysis and the capability of the neural networks to decode/read the archaeological dataset but also to evaluate the method or technique that produced the most accurate results regarding technological and typological clustering.

2.1. Self-Organizing Map Network

The Self-Organizing Map (SOM) is an unsupervised clustering technique developed by Teuvo Kohonen in the 1980s [33,34] and successfully used in a wide range of applications [35] such as data, industrial and biomedical analysis, pattern recognition, time series prediction, and brain modeling. This technique allows multidimensional data to be projected into a two-dimensional space so that similar data points are placed close together on the map [36]. The resulting map can be used to visualize and analyze data more efficiently because each point on the SOM representation is assigned a color that depends on the weight vectors associated with the neurons and represents the relative position of the neurons on the map based on their distance. Neighboring neurons on the map have similar colors, while distant neurons have different colors.

The SOM consists of a feedforward NN in which the output layer is a two-dimensional grid of m neurons, each of which is fully connected to the n neurons in the input layer. It is important to note that the number of input nodes is much larger than the number of output nodes $(n \gg m)$. Figure 1 shows a graphical representation of the SOM output layer.

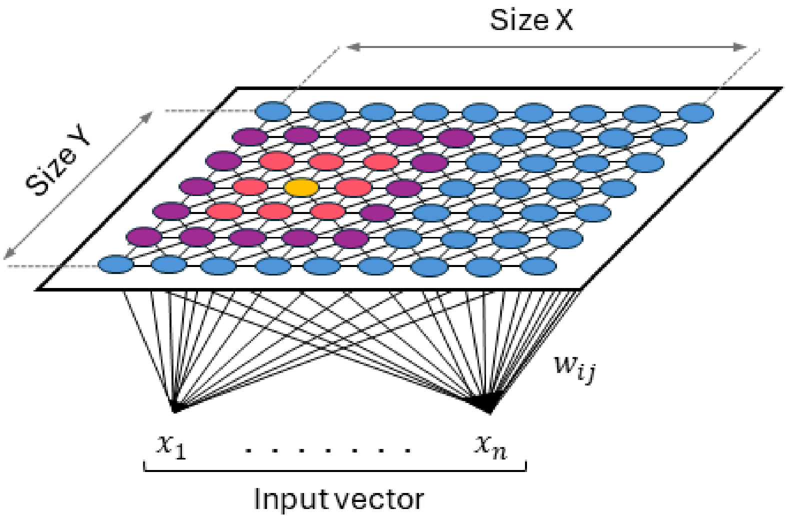


Figure 1. SOM output layer representation.

Connections are weighted through weights w_{ij}, where i is the i-th output node of the output vector $y : \{y_i : i = 1, \ldots, m\}$ and j is the j-th input node of the input vector $x : \{x_j : j = 1, \ldots, n\}$ [25], and the vector of weights resulting from all connections has the same size as the input vector [26]. Figure 2 shows an example of the connections between an n-dimensional input vector and the m output nodes represented linearly.

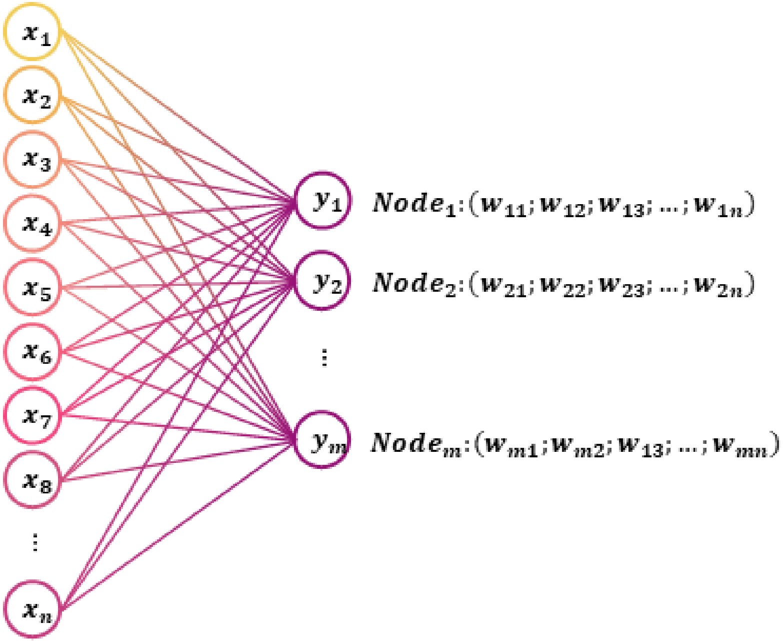

x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
⋮
x_n
y_1
$Node_1$: $(w_{11}; w_{12}; w_{13}; \ldots; w_{1n})$
y_2
$Node_2$: $(w_{21}; w_{22}; w_{23}; \ldots; w_{2n})$
⋮
y_m
$Node_m$: $(w_{m1}; w_{m2}; w_{13}; \ldots; w_{mn})$

Figure 2. Example of connections between an n-dimensional input vector and the m output nodes.

Thus, the NN has weighted connections between the nodes of successive layers just like a classical NN; however, the substantial difference between the two is that the SOM neurons have no activation function. During training, the SOM adjusts its weights to represent the input data optimally. This process occurs through the iterative presentation of input data to the network. In fact, during each iteration, the SOM selects the neuron with the closest weights to the input data, which therefore represents the winning neuron, also called the Best Matching Unit (BMU) [37], for that iteration. Then, through a process known as neighborhood adaptation, the SOM adjusts the weights of the neurons surrounding the winning neuron so that the neurons close to the winner become more similar to the input data [38,39]. In this way, input data with similarities are grouped in the output node grid. There are two variants of the SOM training algorithm [40]: traditional sequential training, in which samples are considered one at a time, and batch training, in which the dataset is presented to the SOM. The steps of the traditional SOM training algorithm are described in detail below.

(1) Weight initialization. The weights of the neurons in the SOM are randomly initialized, preferably from the input vectors domain [41].
(2) Input vector selection. A random input vector x is selected from the training dataset X.
(3) Distance calculation. The distance between the input vector and the vector of weights of each neuron in the map is calculated, generally using Euclidean distance as a metric [38].
(4) Identification of the winning neuron. The neuron with the weight vector most similar to the input vector is identified as the BMU. The equation for determining such a neuron is as follows [42]:

$$i^* = \text{argmin}_i \|x - w_i(t)\|^2 \; \forall i \tag{1}$$

where i^* is the index of the BMU neuron, x is the input vector, w_i is the weight vector of neuron i at the iteration $t \in \{I, \ldots, epoch\}$, and $\|x - w_i\|$ is the Euclidean norm calculated for all i. In general, the SOM neurons selected several times as winners during the various iterations represent the input clusters.

(5) Neighborhood definition. After selecting the BMU, a neighborhood is defined around the neuron itself. The neighborhood is defined by a neighborhood function $h_{(i^*,i)}$ which determines the intensity of the weight update for neuron i. The neighborhood function represents the influence of neuron i^* on neuron i and generally is a Gaussian function defined as follows [31]:

$$h_{i^*,i} = e^{\frac{-\|r_i - r_{i^*}\|}{2\sigma^2(t)}} \tag{2}$$

where r_i and r_{i^*} are the positions of neurons i and i^*, respectively, and $\sigma(t)$ is the neighborhood size at the training iteration t.

The neighborhood function is centered on the winning BMU neuron and decreases exponentially according to the distance between neurons on the map, as shown in Figure 3.

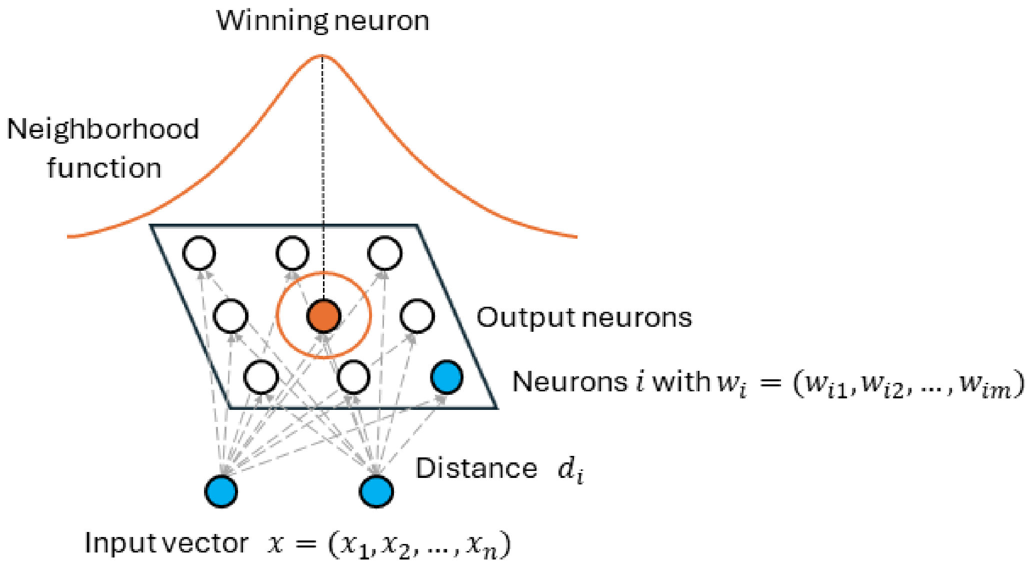

Winning neuron
Neighborhood function
Output neurons
Neurons i with $w_i = (w_{i1}, w_{i2}, \ldots, w_{im})$
Distance d_i
Input vector $x = (x_1, x_2, \ldots, x_n)$

Figure 3. Visual representation of the neighborhood function.

By determining the degree to which the weights of neurons close to the neuron activated by the input are adapted, this function promotes the organized and structured alignment of neurons in the topological map with the input data.

(6) Neighborhood shrinkage. The neighborhood width $\sigma(t)$ gradually decreases during training, allowing the map neurons to adapt to the input data and reach a stable state. The neighborhood amplitude is typically defined as follows:

$$\sigma(t) = \sigma_0 \mathrm{e}^{(-\frac{\mathrm{t}}{\tau})} \tag{3}$$

where σ_0 is the initial neighborhood width and τ is a time constant that regulates the speed of the neighborhood in the training iteration.

(7) Weight update. Once the neighborhood is defined, the weights of the neurons in the map are updated with the following learning rule [42,43]:

$$w_i(t+1) = w_i(t) + \Delta w_i(t) = w_i(t) + \eta(t)\, h_{(i^*,i)}(x - w_i(t)) \quad \forall i \in N_{i^*} \tag{4}$$

where w_i is the weight vector of node i at iteration t, $\eta(t)$ is the learning rate at iteration t, $h_{(i^*,i)}$ is the neighborhood function, x is the input vector, and N_{i^*} defines a neighborhood region.

(8) Learning rate update. The learning rate is a training parameter that controls the size of the vector of weights in SOM learning [44]. There are many functions of the learning rate, of which the most commonly used is the exponential function:

$$\eta(t) = \eta_{initial}\left(\frac{\eta_{final}}{\eta_{initial}}\right)^{\frac{t}{t_{max}}} \tag{5}$$

The values of $\eta_{initial}$ and η_{final} are chosen based on the problem being analyzed, and $t_{max} = epoch$ is the maximum iteration time set at the beginning of the algorithm. The value of the learning rate is gradually reduced during training to ensure stable convergence of the map.

(9) Repetition. Steps 2–8 are repeated for all input vectors in the training dataset.

(10) End of training. SOM training is stopped when a set number of iterations has been reached or when the map has reached a stable configuration and the data distribution on the map has reached a satisfactory level.

The SOM, and in particular its implementation in MATLAB, allows one to evaluate and visualize the network's output after finishing the training phase [45]. For example, a graphical representation of the SOM's topology (Figure 4a), neighbor weight distances (Figure 4b), and hits (Figure 4c) can be obtained.

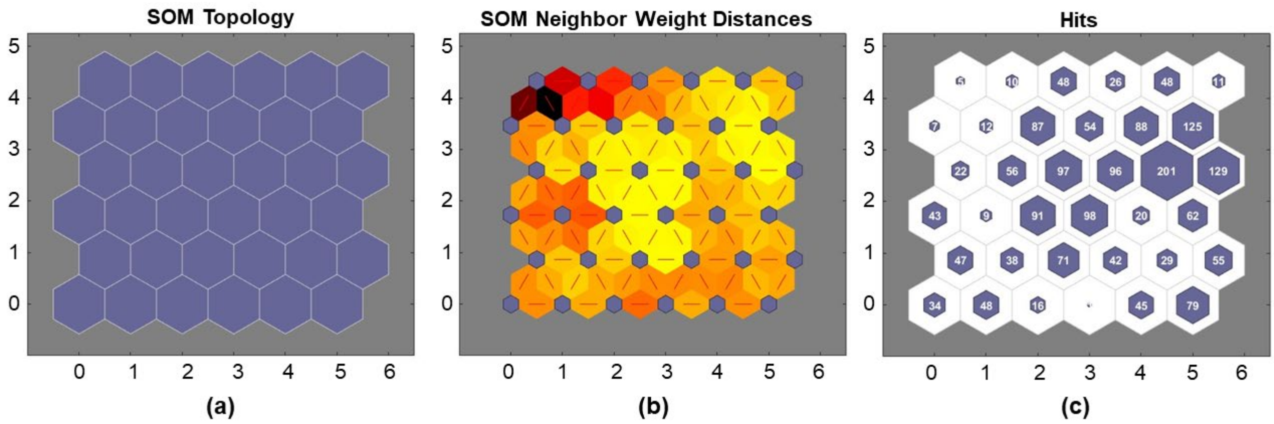


Figure 4. Graphical representation of the SOM after training. (**a**) Topology. (**b**) Neighbor weight distances. (**c**) Hits.

In Figure 4b, the blue hexagons represent neurons, and the boxes containing red lines connecting neurons have light colors for shorter distances and dark colors for longer distances. Figure 4c shows a histogram representing the number of times each neuron in the SOM map was selected as a winner during training.

Together with the previous graphical representation, each test produced a set of point cloud graphs corresponding to the number of clusters for that test (Figure 5). The "y"-axis presents a selection of the 150 top analyzed variables for each artifact, considered significant variables from an archaeological perspective. (The major categories listed on any cluster's "y"-axis do not represent all major technological and typological categories. The quantity exceeds the maximum number of variables that allow proper cluster reading. The variables listed must, therefore, be considered a selection of the major variables.) In contrast, the "x"-axis shows each artifact selected during the clustering. (The empty ranges signify either the absence of artifacts within a specific cluster (from artifact n° 0 to 950, and from artifact n° 1450 to 2950) or a lack of data (from artifact n° 951 to 1449). The database originally used for this study indeed contains additional sites belonging to different chrono-cultural phases as part of a wider project.)

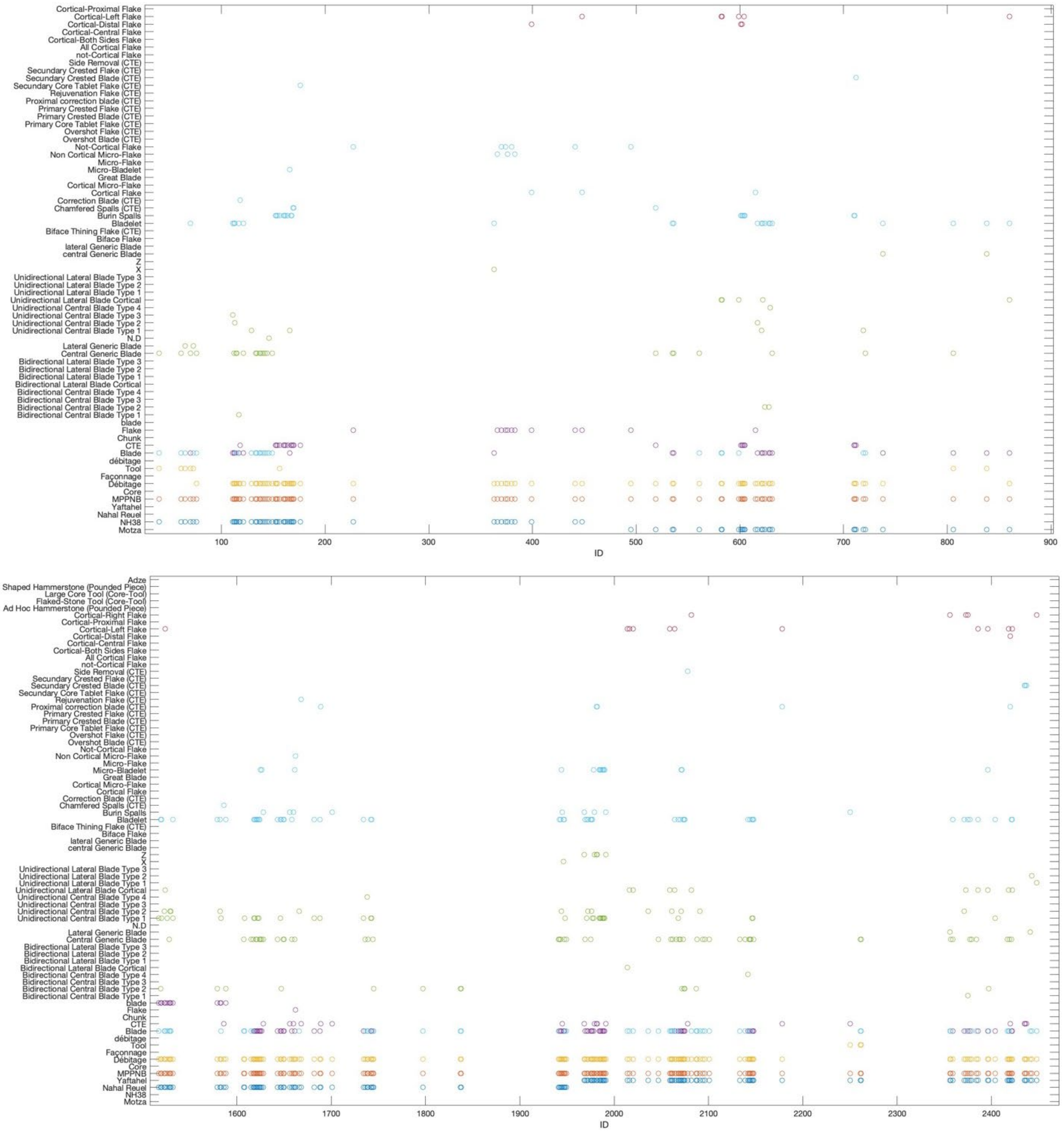


Figure 5. Example of clustering with SOM: some technological categories of artifacts from Nahal Yarmuth 38 and Motza (**top**) are grouped in the same cluster, as well as artifacts from Nahal Reuel and Yiftahel (**bottom**).

Notably, an artifact may contain a mutable number of variables. An artifact (listed on the "x"-axis with a numeric ID) often repeats itself in many variables, but some significant categories cannot be part of others. For example, an artifact that belongs to the "débitage category" can be either a flake, a blade/laminar artifact, or a CTE. However, it cannot be part of the "core", "façonnage", or "tool" categories. Moreover, for example, a blade/laminar blank can be a "great/large blade", a proper "blade", a "bladelet", or a "micro-bladelet", and therefore, it can belong to other sub-categories (such as cortical flake/blank, bidirectional or unidirectional blade/blank, etc.) depending on the specific characteristics of the artifact. Furthermore, each symbol in the graph represents a specific artifact, and the use of different colors helps improve exclusively the graph's readability.

In conclusion, the SOM is a powerful and flexible clustering technique that enables efficient data visualization and analysis. Indeed, this type of visualization can help identify groups of neurons that respond similarly to different inputs and identify map regions that represent clusters of similar inputs.

2.2. K-Means Clustering Algorithm

K-Means is one of the most widely used clustering algorithms for data analysis because of its simplicity and computational efficiency. It is a straightforward, fast, unsupervised, nondeterministic numerical method that assigns data to k different clusters iteratively until it converges to a local minimum [46]. Once the parameter k is chosen (a priori), the k cluster centers are randomly set, and iteratively, the dataset instances are assigned to the cluster with the nearest center. Specifically, at each iteration, the centroid values are recalculated as the average of the cases assigned to each cluster at the previous iteration until a fixed value of maximum iterations is reached or the local minimum of the criterion function, defined as follows:

$$E = \sum_{i=1}^{k} d_i = \sum_{i=1}^{k} d(x, x_i) \tag{6}$$

where d_i is the distance between the x_i data of the i-th cluster C_i and the average x of the clusters. Several metrics [47] exist for calculating the point-to-point distance between elements and centroids, including the following:

(1) Minkowski distance.

$$d(x, x_i) = \left(\sum_{x \in C_i} |x - x_i|^r\right)^{1/r} \tag{7}$$

This distance can be seen as a generalization of other metrics such as Manhattan distance and Euclidean distance.

(2) Manhattan distance. This defines the distance between two points as the sum of the absolute differences of their Cartesian coordinates. It is obtained by setting $r = 1$ in the Minkowski distance formula.

$$d(x, x_i) = \sum_{x \in C_i} |x - x_i| \tag{8}$$

(3) Euclidean distance. This is the minimum distance between two objects defined as the root of the quadratic error between them. It is obtained by setting $r = 2$ in the Minkowski distance formula.

$$d(x, x_i) = \left(\sum_{x \in C_i} |x - x_i|^2\right)^{1/2} \tag{9}$$

The strengths of the K-Means algorithm include its scalability, efficiency, and simplicity [48]. Most importantly, it is unsupervised, allowing inferences to be drawn from the dataset without any prior knowledge. This is an excellent advantage since labeled data are often expensive and complicated to obtain. The main disadvantage of this algorithm is the need to define the number k of clusters in advance, which is only sometimes so evident in real applications and for high-dimensional datasets.

Different approaches were selected within the K-Means-based system.

A test was performed using exclusively the K-Means algorithm, without any combination with other techniques. In this specific case, the K-Means algorithm chooses the initial centroids randomly between the given points in the dataset.

In particular, the algorithm randomly selects several points from the dataset and uses them as initial centroids for clusters. These centroids are then used as a starting point for the iterative process of the K-Means algorithm, which assigns each point to the nearest cluster and recalculates the centroids of each cluster.

The initial choice of centroids can affect the quality of clusters obtained by the K-Means algorithm. In some situations, a more sophisticated centroid selection strategy may be helpful. For example, a centroid selection algorithm that considers the density of the points in the dataset can be used to select centroids that represent the data structure well. However, the MATLAB K-Means function uses the initial random choice of centroids.

Another test was based on combining the K-Means algorithm and PCA (Principal Component Analysis). This combination is a common technique used in many analyses for dimensionality reduction first and then for data clustering. PCA is a dimensionality reduction technique that transforms data into a new coordinate system. The first significant component captures the maximum possible variance in the data, the second principal component captures the maximum remaining variance orthogonally to the first, etc. The main steps of PCA include standardization of the data if the characteristics have different scales, construction of a covariance matrix of the standardized data, calculation of eigenvalues and eigenvectors of the covariance matrix, selection of the main components based on the larger eigenvalues, and transformation of the original data in the space of the selected main components. Combining PCA and K-Means begins with data preprocessing, such as normalization. Next, PCA is applied to reduce the size of the data, reduce noise, and facilitate the visualization of the data in a small space. Finally, the K-Means algorithm on the dimensionally reduced data. An example in the Matlab_R2024a environment follows:

(1) Load the data from the dataset and assign the data matrix to a variable.
(2) Then, apply PCA to the data to obtain the main components. The data will be transformed into the space of the main components, and the values "eigen" represent the variance explained by each component.
(3) Next, calculate the cumulative variance explained by the main components and find the minimum number of principal components needed to define at least 95% of the total variance.
(4) Then, project the data into the space of the first selected main components, reducing the dimensionality. The K-Means algorithm runs on these dimensionally reduced data for a desired number of clusters.
(5) The results of clustering, including the assignment of clusters for each data point and the coordinates of the clusters' centroids, are displayed in a scatter chart. The centroids are indicated by a black 'x' symbol.

The test was performed by first processing the dataset with the PCA technique, then applying the standard K-Means and reporting the data to the natural size.

This combination of PCA and K-Means reduces data dimensionality, eliminates noise, and facilitates clustering. It is beneficial when working with high-dimensional data.

The main advantages of combining PCA and K-Means include reducing the noise in the data, improving the accuracy of clustering, reducing the computational time for the K-Means algorithm by reducing the dimensionality, and the ability to view data in 2D or 3D form, making it easier to interpret clustering results.

In summary, combining PCA and K-Means is a powerful data analysis technique that helps reduce dimensionality, eliminate noise, and improve understanding of clusters in data.

Lastly, a test was run based on the K-Means++ technique, a variant of the K-Means algorithm that improves the centroid initialization phase for better and more stable clustering results. K-Means++ tries to choose the initial centroids to reduce the probability of obtaining suboptimal solutions.

The K-Means++ algorithm only modifies the centroid initialization phase as follows, while the rest of the clustering process remains unchanged:

(1) It starts by randomly choosing the first center between the data points. The square distance from the nearest center already chosen is calculated for each data point.
(2) To choose the next center, a new point is selected based on a probability distribution proportional to the square distance previously calculated. This means that points farthest from the current centroids will likely be chosen as new centroids.

(3) This process is repeated until all the necessary centroids have been chosen.
(4) Once the initial centroids are chosen, the standard K-Means algorithm (assignment and update) is used until convergence.

K-Means++ offers several advantages. First, the initial centroids chosen with this method are more evenly distributed than those selected with simple random selection, leading to a higher probability of good-quality clustering. In addition, the algorithm reduces the likelihood of finishing in suboptimal local minima, improving the result's robustness and stability. Despite the more complex initialization, the algorithm generally converges more quickly and with better results than the standard K-Means.

K-Means++ is an improved technique for initializing centroids in the K-Means algorithm, which leads to greater efficiency and better clustering results. This variant is beneficial for data with complex structures or if it is necessary to reduce variability in clustering results.

2.3. Hierarchical Clustering

Hierarchical clustering is an alternative approach that allows the model output to be visualized and interpreted using particular dendrograms. Its variants include (1) the partitioning approach (top-down), which, from a single cluster comprising the entire dataset, iteratively creates subdivisions into smaller-sized clusters containing a single instance [43], and (2) the agglomerative approach (bottom-up), dominant for constructing embedded classification schemes [49], which works in reverse to the previous one in that in the initial situation, each iteration belongs to a different cluster (there are as many clusters as there are data in the dataset) and iteratively joins neighboring clusters until there is only one large cluster or a user-chosen number of clusters left. In turn, hierarchical agglomerative clustering has two standard versions called "single-linkage" and "complete-linkage", using which a test was conducted. The former unites the u and v clusters for which the distances between members i and j are the smallest (see Equation (10)). In contrast, the latter, also called the Farthest Point Algorithm or Voor Hees Algorithm, uses dissimilarity between members as the union criterion (see Equation (11)).

$$d(u,v) = \min\left(dist\left(u_i, v_j\right)\right) \tag{10}$$

$$d(u,v) = \max\left(dist\left(u_i, v_j\right)\right) \tag{11}$$

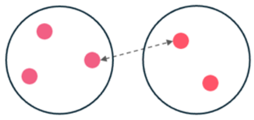
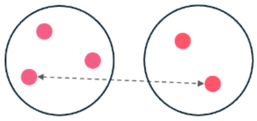
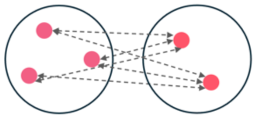
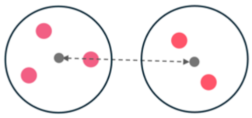
Other widely used types of linkage are average linkage, which joins pairs of clusters based on the minimum average distances between all members of the two groups (see Equation (12)), and centroid linkage, which joins groups based on the distance between centroids (see Equation (13)).

$$d(u,v) = \sum\nolimits_{ij} \frac{dist\left(u_i, v_j\right)}{|u| * |v|} \tag{12}$$

$$d(u,v) = \|c_u - c_v\|_2 \tag{13}$$

The approaches described above are summarized in Table 1.

Table 1. Summary of hierarchical linkage approaches.

Method	Single Linkage: Minimum distance between elements in clusters	Complete Linkage: Maximum distance between elements in clusters	Average Linkage: Average of the distance of all pairs	Centroid Linkage: Minimum distance between the centroids of two clusters
Visualization				

As previously introduced in the discussion of classical K-Means, there are several metrics for calculating the distance between dataset elements. In contrast, the metric generally used to measure similarity between instances is cosine similarity, defined as follows:

$$s(x_a, x_b) = \cos\theta = \frac{x_a \cdot x_b}{\|x_a\|\|x_b\|} = \frac{\sum_{i=1}^{n} x_{ai} x_{bi}}{\sqrt{\sum_{i=1}^{n} {x_a}_i^2}\sqrt{\sum_{i=1}^{n} {x_b}_i^2}} \tag{14}$$

All these methods follow a familiar pattern based on the following steps:

(1) Calculate the matrix of distances between all examples.
(2) Represent each instance as a single cluster or unite of all the cases as a unique global cluster (depending on the variant chosen).
(3) Unite cluster pairs or divide the whole cluster according to the selected method.
(4) Update the distance matrix.
(5) Repeat steps 2 to 4.

Generally, hierarchical bottom-up approaches are more accurate than top-down approaches but also computationally more expensive [50].

Another test was instead conducted using Ward's method. The Ward linkage method is a linkage version used in hierarchical agglomerative clustering. The main idea is based on minimizing the sum of the variances within the joined clusters. This method tends to produce clusters of a similar size.

The example that follows shows its MATLAB use:

(1) First, the data are loaded and standardized if necessary.
(2) Next, the hierarchical clustering algorithm using Ward's method is applied.
(3) The results are displayed in a dendrogram.

In MATLAB, the function "cluster" allows one to achieve a specific amount of clusters from the dendrogram, specifying the maximum number of desired clusters. Finally, the clustering results are displayed in a scatter plot, coloring the points based on the cluster to which they belong.

This method helps explore the data structure without specifying the number of clusters in advance. The resulting dendrogram visually represents the cluster hierarchy, making it easy to decide where to cut to obtain the desired number of clusters.

The clustering process begins by considering each data point as a separate cluster. Distances between each pair of clusters are calculated using the Ward distance (15), which measures the increase in the sum of the variances if the two clusters are joined together.

$$d_{Ward} \tag{15}$$

The Ward distance between two clusters, A and B, is calculated as the weighted sum of the internal variance of the two clusters, where the variance is measured as the square Euclidean distance between the cluster centroids. The formula is expressed as follows:

$$d_{Ward}\ (\mathrm{A},\ \mathrm{B}) = [(n_A\ n_B)/(n_A + n_A)]\ ||\mu_A - \mu_B\ ||^2 \tag{16}$$

where n_A e n_B are the point numbers in clusters A and B, respectively; μ_A e μ_B are the centroids of clusters A and B, respectively; and $||\mu_A - \mu_B||^2$ is the square Euclidean distance between the centroids of the two clusters.

The two clusters that lead to the most minor increase in the sum of the variances are joined. This process repeats until all points are merged into a single cluster or until the desired number of clusters is reached.

Ward's linkage method is advantageous because it tends to create clusters of similar size and minimizes variance within clusters, making clustering more meaningful and interpretable.

2.4. Silhouette Analysis

To quantify the quality of clustering, one can use silhouette analysis, a graphical tool that shows the level of clustering of instances in clusters. This analysis is based on calculating the silhouette coefficient of each example in the dataset, defined as follows:

$$s_i = \frac{b_i - a_i}{\max\{b_i, a_i\}} \tag{17}$$

where a_i is the cluster cohesion, calculated as the distance between each instance and all other instances belonging to the same cluster, and b_i is the separation of the cluster from the nearest cluster, calculated as the average distance between each example belonging to the i-th cluster and all examples contained in the nearest cluster.

The silhouette coefficient takes values in the range $[-1,1]$, where the upper extreme represents the ideal condition.

To explain in more detail what has just been stated, we present an example inspired by the one described in [51]. Starting with the unlabeled dataset shown in Figure 6a, whose actual partition is shown in Figure 6b, we wanted to apply a clustering algorithm with different values of parameter k to perform non-supervised clustering quickly and efficiently. We chose K-Means for simplicity, but this analysis applies to other clustering algorithms. The parameter k varied from 2 to 5; the distance metric chosen was Euclidean.

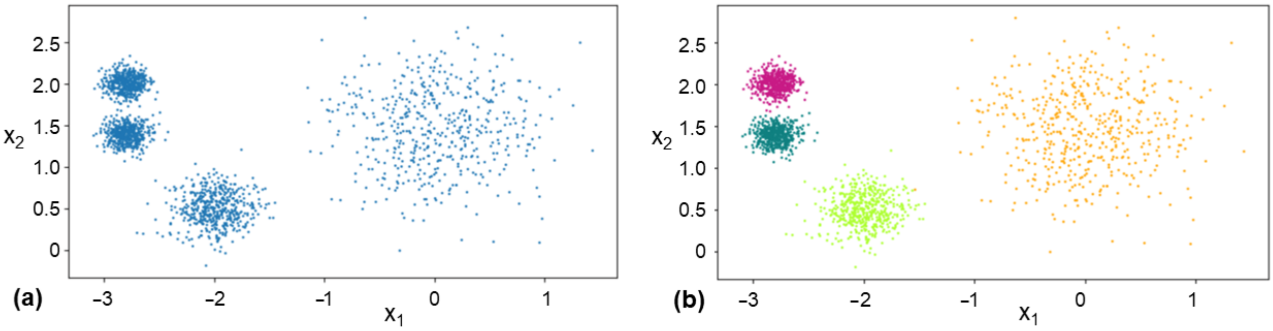


Figure 6. Data visualization. (**a**) Distribution of the dataset. (**b**) Real dataset partitioning.

In this simple case, the feature space is two-dimensional (features x_1 and x_2), so it is possible to visualize the distribution of the data and guess the most appropriate k number. This is not possible for datasets with dimensionality greater than 3, so the best solution is always to apply silhouette analysis. First, we graphically compared the results of the silhouette scores obtained by varying the parameter k (Figure 7).

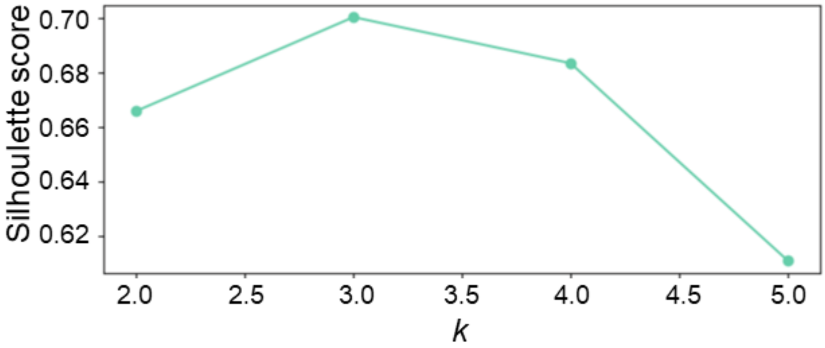


Figure 7. Silhouette scores for various values of k.

By comparing the silhouette scores for different cluster numbers, one could infer that k = 3 is the best choice; however, k = 4 is also quite good and certainly better than k = 2 or k = 5. A more informative visualization is provided by the silhouette diagram, which allows one to view the silhouette coefficients of each instance, organized according to the cluster to which they have been assigned and sorted by coefficient value. In such a diagram, each cluster is associated with a knife shape, whose height indicates the number of instances contained in the cluster. At the same time, the width represents the silhouette coefficients of the instances in the clusters. In the present case, the silhouette plots associated with the four different values of k are shown in Figure 8, where the vertical dashed red lines represent the average silhouette coefficients.

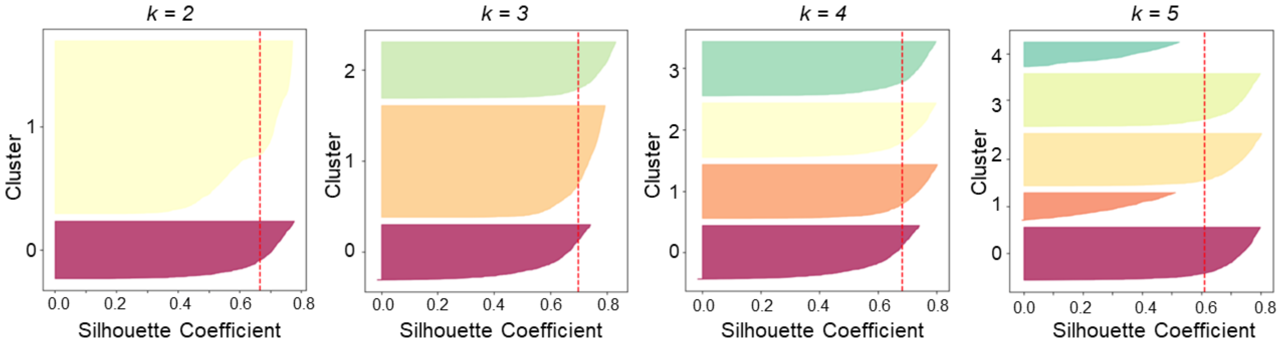


Figure 8. Silhouette diagrams for various values of k.

As can be seen, for k = 5, the silhouette values of some clusters are lower than the total mean value, indicating that the instances in those clusters are too close to the other clusters, and therefore, the clustering of the data is not good. In other cases, however, the instances exceed the mean value and may represent good choices. However, for k = 2 and k = 3, some clusters are very large (with an index of 1 in both cases), while for k = 4, all clusters have similar sizes. Therefore, although the total silhouette score for k = 3 is higher than that obtained for k = 4 (see Figure 7), the latter option seems the best.

Therefore, silhouette analysis is an excellent tool for assessing which k parameter is optimal for a specific application.

In this research paper, we applied the SOM, K-Means, and hierarchical clustering methods to analyze the archaeological data at our disposal. The results obtained by each method were thoroughly examined, and we discussed the most suitable approach for our specific application. We conducted multiple tests for each algorithm with varying cluster sets to enable a comprehensive comparative analysis. By testing the SOM, K-Means, and hierarchical algorithms with 36 and 20 clusters, we aimed to facilitate a thorough comparison. In particular, we applied different techniques for the K-Means and hierarchical algorithms to create distinct test scenarios. For instance, the K-Means algorithm was tested with 36 clusters, 20 using the PCA technique, and 20 using the K-Mean++ technique. On the other hand, the hierarchical algorithm was tested with 36 clusters using Ward's method and 20 clusters using the complete linkage technique. Furthermore, each test and setting was conducted between 5 and 10 times, and the results consistently showed no significant variation.

We supported each scenario with neighbor weight distance and hits analysis for the SOM, and with silhouette analysis for the K-Means and hierarchical algorithms. Furthermore, we evaluated the effectiveness and validity of each application through archaeological interpretation.

The comparative analysis facilitated the experts' selection of the most suitable approach for clustering analysis on a homogeneous dataset (Supplementary Material: Matrix). From an archaeological standpoint, it effectively demonstrated the potential to accurately replicate the archaeological interpretation of the technological and typological groups within the composition of the lithic assemblage. This research paper has opened up the opportunity to explore the inherent characteristics of archaeological datasets further, providing professionals with the capability to discern cultural aspects within homogeneous material cultures.

3. Results

The subsequent analyses present the results regarding the silhouette comparison between the K-Means and hierarchical algorithms and the outcomes of neighbor connection and distance concerning the SOM analysis. The efficiency results are then interpreted from an archaeological standpoint to enhance our comprehension of the various algorithms' capacity to decode archaeological datasets.

The SOM analysis was carried out on two different tests with 36 clusters (Figure 9) and 20 clusters (Figure 10). The first test showed different primary connections (Figure 9, left), and indeed, the number of clusters is relevant to the purpose of the archaeological research (Figure 9, right). Meanwhile, the test on 20 clusters showed fewer neighbors' weight distances (Figure 10, top) and significant hits (Figure 10, bottom).

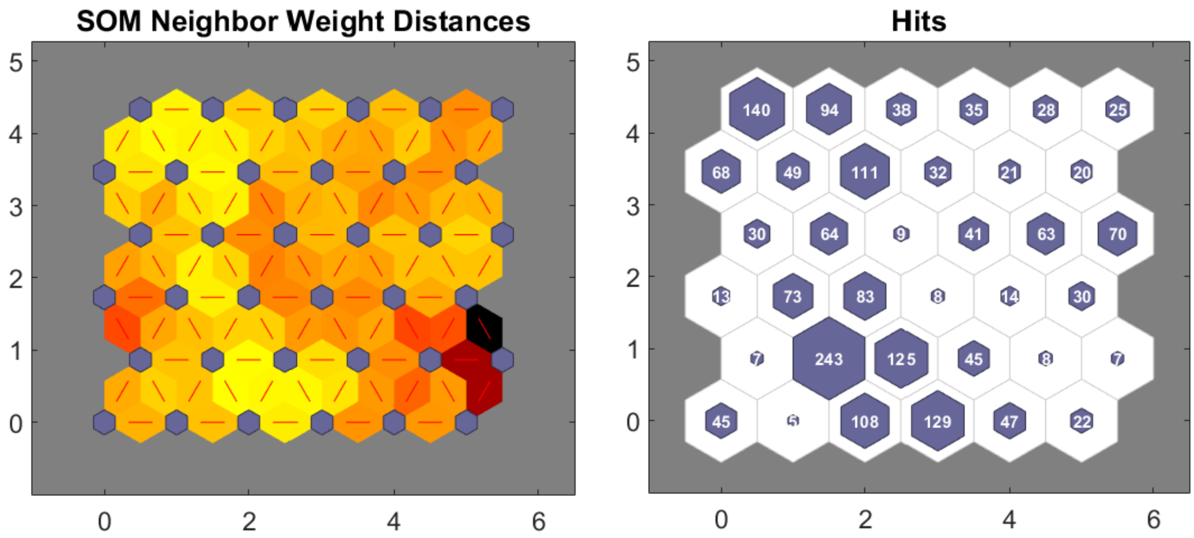


Figure 9. Neighbor weight distances (**left**) and hits (**right**). Test performed on 36 clusters (each graph must be read from the right to the left and from the bottom to the top of the grid).

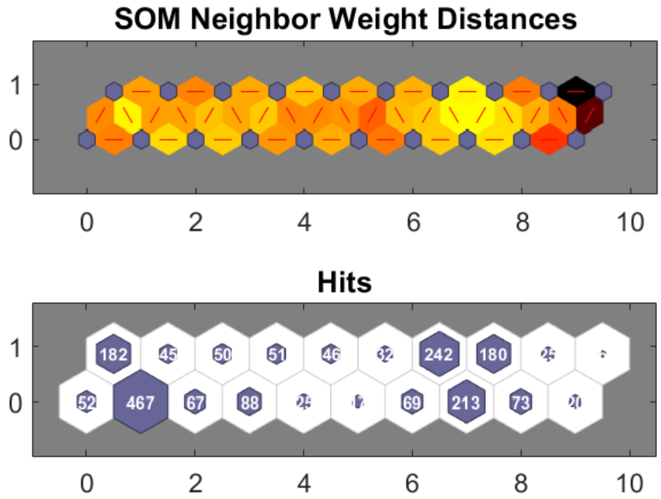


Figure 10. Neighbor weight distances (**top**) and hits (**bottom**). Test performed on 20 clusters^5.

The analysis of 36 clusters yielded significant insights from an archaeological perspective (Figure 9). The examination revealed the presence of technological groups of artifacts corresponding to specific stages in the operational sequence. For instance, cluster 3 comprised cortical artifacts (cortical flakes and blades) alongside non-cortical elements, representing the initial stages of core reduction. These artifacts, known as blanks, were identified across all analyzed sites, indicating a commonality in operational sequences across the Levant region (Supplementary Materials: Figure S1a,b).

Cluster 5 also proved to be pivotal, encompassing laminar and flake cores, crucial components in lithic production. This finding was consistent with expectations, reflecting the technological homogeneity observed across all sites. Notably, cores are fundamental elements in lithic production processes (Supplementary Materials: Figure S2a,b).

Furthermore, cluster 8 underscored a specific stage in production, primarily featuring bladelets, which are indicative of the laminar-core reduction process and were prevalent across all sites. These findings shed light on the shared operational sequences and technological practices among the sites under study. From a typological point of view, many clusters seem to reproduce a specific logic based on several parameters that were taken into account, showing positive results. Clusters 21 and 35 showed a group of tools (retouched artifacts) made on laminar blanks (blades and bladelets) shared by all sites. Retouching a laminar blank is, in fact, the main step in achieving the production of formal tools, such as arrowheads, denticulates, sickle blades, etc. Cluster 29 instead showed the group of retouched flakes and CTEs (core trimming elements), representing a secondary production. The latter products are used to make mostly scrapers, a cluster also highlighted in cluster 36.

In the analysis, most clusters demonstrated favorable outcomes regarding the SOM's capability to categorize and organize various clusters based on the technological and typological characteristics of the assemblage. This resulted in a consistent dataset of sites sharing similar technological traits. However, a few clusters exhibited divergence in typological terms, yielding significant preliminary findings. Notably, the arrowheads were grouped differently, revealing two distinct clusters. One cluster consisted exclusively of arrowheads from Nahal Reuel, designated as a unique cluster specific to this site (cluster 1). Conversely, arrowheads from other sites, such as Motza, Nahal Yarmuth, and Yiftahel, were grouped in a separate cluster (cluster 7). This differentiation holds archaeological significance as it suggests that, despite the shared technological aspects and numerous typological features among all sites, the arrowheads reflect distinct cultural preferences, potentially indicating sub-cultural distinctions (Supplementary Materials: Figure S3).

A few clusters that were produced showed a particular logic for the NN to read the archaeological dataset, which was understandable but incorrect in archaeological terms. In two clusters (clusters 11 and 12), a few cores were grouped with large-size flakes and hammerstones (Supplementary Materials: Figure S4a,b). Although the choice highlighted a grouping method based, in this specific case, mainly on the dimensions and weight of the artifacts, the grouping needed to be corrected from a technological point of view. Only one of the produced clusters was instead apparently nonsensical (cluster 14).

In assessing the test set comprising 20 clusters, the SOM consistently performed well in interpreting and decoding the archaeological dataset (Figure 10). However, the reduced number of clusters impacted the precision of the subdivision. For instance, cluster 1 effectively grouped cores from various sites, indicating a homogenous technological pattern in core reduction. Conversely, clusters 2 and 8 displayed a lack of specificity, encompassing the débitage category and flakes production from all sites without discerning distinctions (Supplementary Materials: Figure S5a,b).

Similar issues were observed in the categorization of tools across clusters 7, 11, and 12, where different types of tools were conglomerated, both typologically and technologically. Despite the decreased precision in this analysis, a noteworthy finding was the identification of a cluster exclusively comprising arrowheads from Nahal Reuel (cluster 12), indicating a distinct typological divergence within a broader homogeneous material culture, reflective of a unique sub-cultural preference (Supplementary Materials: Figure S6).

Several tests were performed regarding silhouette analysis for each K-Means and hierarchical algorithm application. The comparison facilitates a better understanding of each application's capability to decode and read the archaeological dataset.

In the analysis, the K-Means algorithm was applied to a test set comprising 36 clusters and two test sets comprising 20 clusters each. Notably, the latter tests were conducted using two distinct techniques, namely K-Means PCA and K-Means++, to enhance the algorithm's effectiveness in interpreting and deciphering the dataset.

The analysis set on 36 clusters (Figure 11) yielded positive results from an archaeological standpoint in some cases; however, in others, the results were inconclusive, indicating lower effectiveness of the K-Means algorithm in deciphering the archaeological dataset.

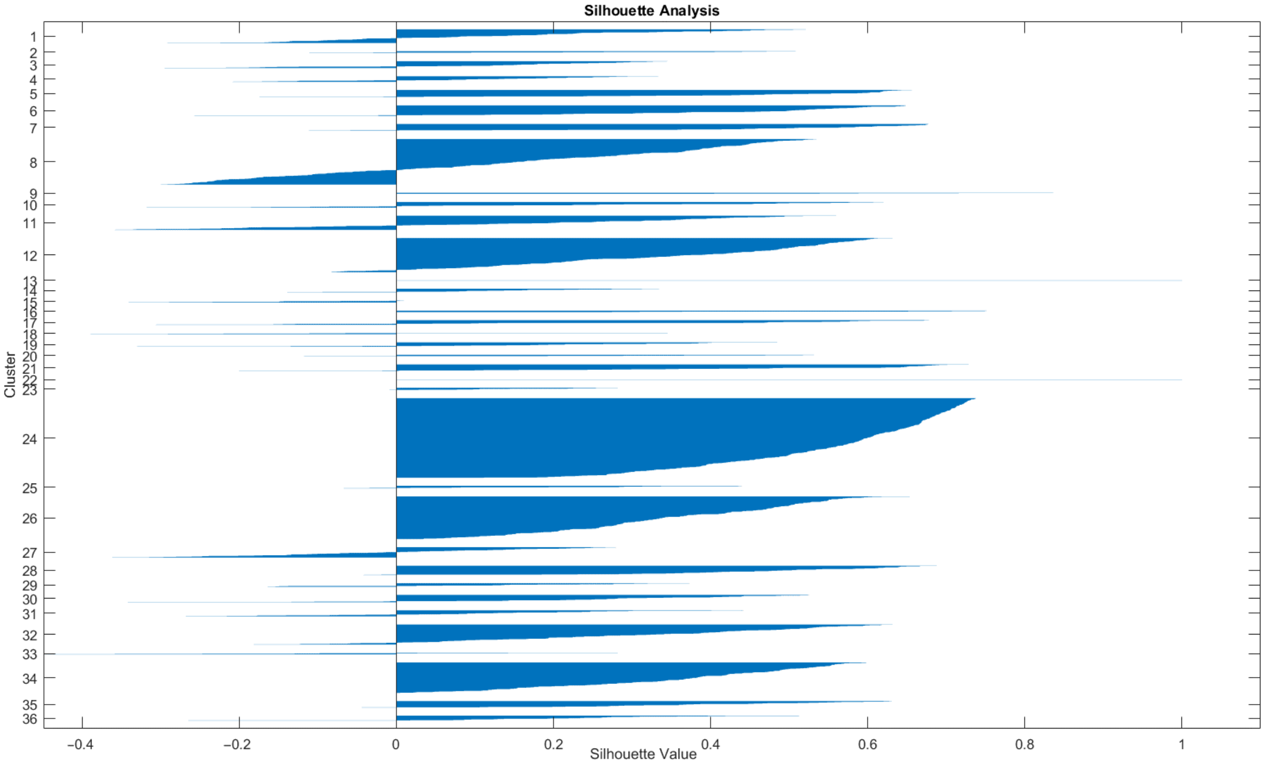


Figure 11. Silhouette analysis, K-Means algorithm. Test conducted on 36 clusters. Mean silhouette value = 0.3.

Cluster 10, for example, showed a group of primary elements from all sites that were expected as a first step of the chain of operations shared by all sites. Clusters 12 and 24 showed a further step, with fewer cortical and non-cortical flakes and laminar blanks from all sites instead. Cluster 8 went a step further, grouping all blades and bladelets from all sites and displaying the main production of any MPPNB site in the Levant: laminar blanks production. Nevertheless, some tools were grouped together with this category, lowering the efficiency of the cluster (Supplementary Materials: Figure S7a,b).

In specific clusters, the efficiency was notably lower. For example, cluster 11 included artifacts from all technological categories, and clusters 35 and 36 encompassed all typological categories. Lastly, clusters 13, 18, 25, 29, and 33 exhibited only a few groupings based on artifacts, indicating the algorithm's inability to determine proper clusters (clusters 13 and 33, Supplementary Materials: Figures S8 and S9).

Despite this lower efficiency in reading and decoding the archaeological dataset, the K-Means algorithm was able to detect a cluster based exclusively on arrowheads from Nahal Reuel (cluster 28), suggesting the possibility of a sub-cultural preference or group within the wider homogenous MPPNB material culture (Supplementary Materials: Figure S10) again.

The analysis of 20 clusters using the PCA technique showed that it was less precise and efficient than the previous analysis set on 36 clusters (Figure 12). The results indicated that some clusters were too general from an archaeological perspective, grouping together many categories without further differentiation. For instance, clusters 5 and 13 showed good results based on primary elements and generic retouched blades, and cluster 16 highlighted a shared technology by all sites, focusing on the group of cores (Supplementary Materials:

Figure S11a,b). However, many other clusters were more mixed. Cluster 15 combined all tool categories from all sites, cluster 7 mixed cores and tools, cluster 2 combined débitage (flakes, blades, and CTEs) and some tools together, and cluster 9 grouped together tools made on blades and hammerstones and adzes (Supplementary Materials: Figure S12). Additionally, some clusters contained only a minimal number of items, such as clusters 6, 17, and 19.

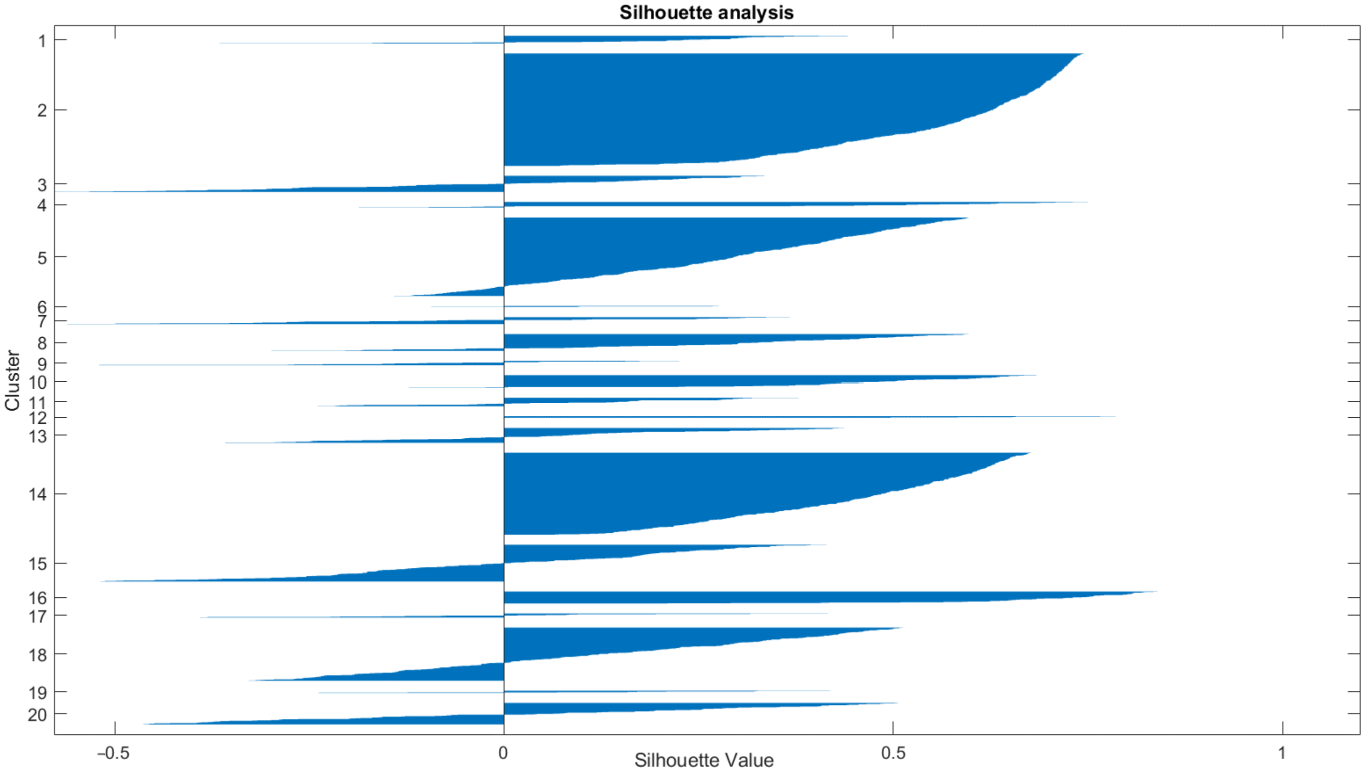


Figure 12. Silhouette analysis, K-Means algorithm (PCA technique). Test conducted on 20 clusters. Mean silhouette value = 0.3.

Despite the low efficiency in deciphering the dataset, the algorithm was able to identify a specific cluster of arrowheads from Nahal Reuel (cluster 10, Supplementary Materials: Figure S13).

The analysis of 20 clusters created using the K-Means++ algorithm yielded interesting results (Figure 13). Some clusters were nearly empty or contained a mix of various technologies, while others were more precise in terms of typology.

For instance, clusters 2, 19, and 20 exhibited a combination of all technological categories of débitage and a few tools (cluster 2, Supplementary Materials: Figure S14a,b). Conversely, clusters 3 and 5 lacked sufficient qualitative and quantitative elements for an archaeological cluster. Cluster 4 exclusively featured cores, indicating a consistent technology across all sites without further subcategories within this group (Supplementary Materials: Figure S15a,b).

Only a few clusters demonstrated acceptable technological efficiency. Cluster 13 displayed primary elements and non-cortical flakes from all sites, representing the initial and second steps in the core reduction process. Finally, cluster 14 grouped retouched flakes and scrapers on flakes, highlighting similarities between these two categories.

The algorithm did, however, produce specific results regarding tool categories. For instance, cluster 9 aggregated specific arrowheads from Nahal Reuel, similar to the previous test, while cluster 17 grouped a few arrowhead types common across all sites (Supplementary Materials: Figure S16).

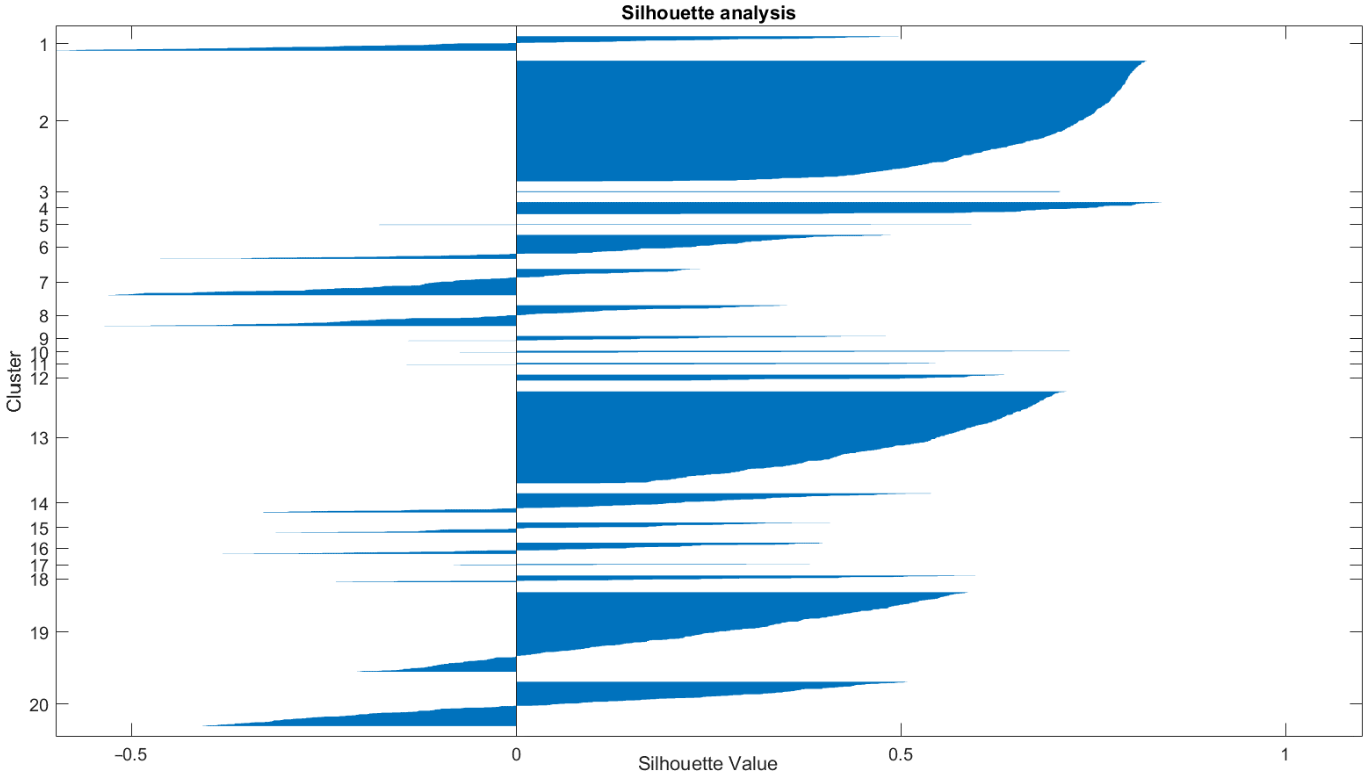


Figure 13. Silhouette analysis, K-Means algorithm (K-means++). Test conducted on 20 clusters. Mean silhouette value = 0.25.

In the analysis, the hierarchical algorithm was applied to a test set comprising 36 clusters, specifically based on Ward's method (Figure 14), and to another set of 20 clusters based on the complete linkage technique (Figure 15) to enhance the algorithm's effectiveness in interpreting and deciphering the dataset. The silhouette analysis was present in each test.

The analysis carried out on 36 clusters showed good results in decoding/reading the archaeological dataset. However, it achieved a lower precision than the SOM under the same conditions (Figure 14).

From a technological point of view, several clusters recognized different steps in the chain of operation during core reduction in all sites. For example, clusters 21 and 26 mainly showed primary elements and flakes with a high percentage of cortex coverage (the first products detached from the core during its reduction), highlighting an initial step of the chain (cluster 21, Supplementary Materials: Figure S17a,b). Clusters 16 and 22 showed the step that followed during the process: the realization of laminar blanks and CTEs in the first cluster and non-cortical flakes in the other. Lastly, cluster 17 recognized the category of retouched flakes together with scrapers, as these are often made on flakes.

From a typological point of view, the hierarchical clustering algorithm struggled more in identifying more precise clusters within the tools category. For example, while a few clusters followed a legitimate logic based on retouch characteristics, such as clusters 1 and 2 which respectively grouped denticulates and notches and denticulates and generic retouched blades (cluster 2, Supplementary Materials: Figure S18a,b), other clusters combined different tools with fewer characteristics in common, such as clusters 8, 9, 12, 14, 27, and 33 (cluster 8, Supplementary Materials: Figure S19a,b). As a matter of fact, these clusters appeared in the silhouette analysis with negative values.

Although a higher level of imprecision was noted during the analysis of the dataset's typological information, cluster 36 showed a unique cluster exclusively from Nahal Reuel, represented solely by arrowheads. This again highlights a possible sub-cultural preference within a wider shared cultural substratum.

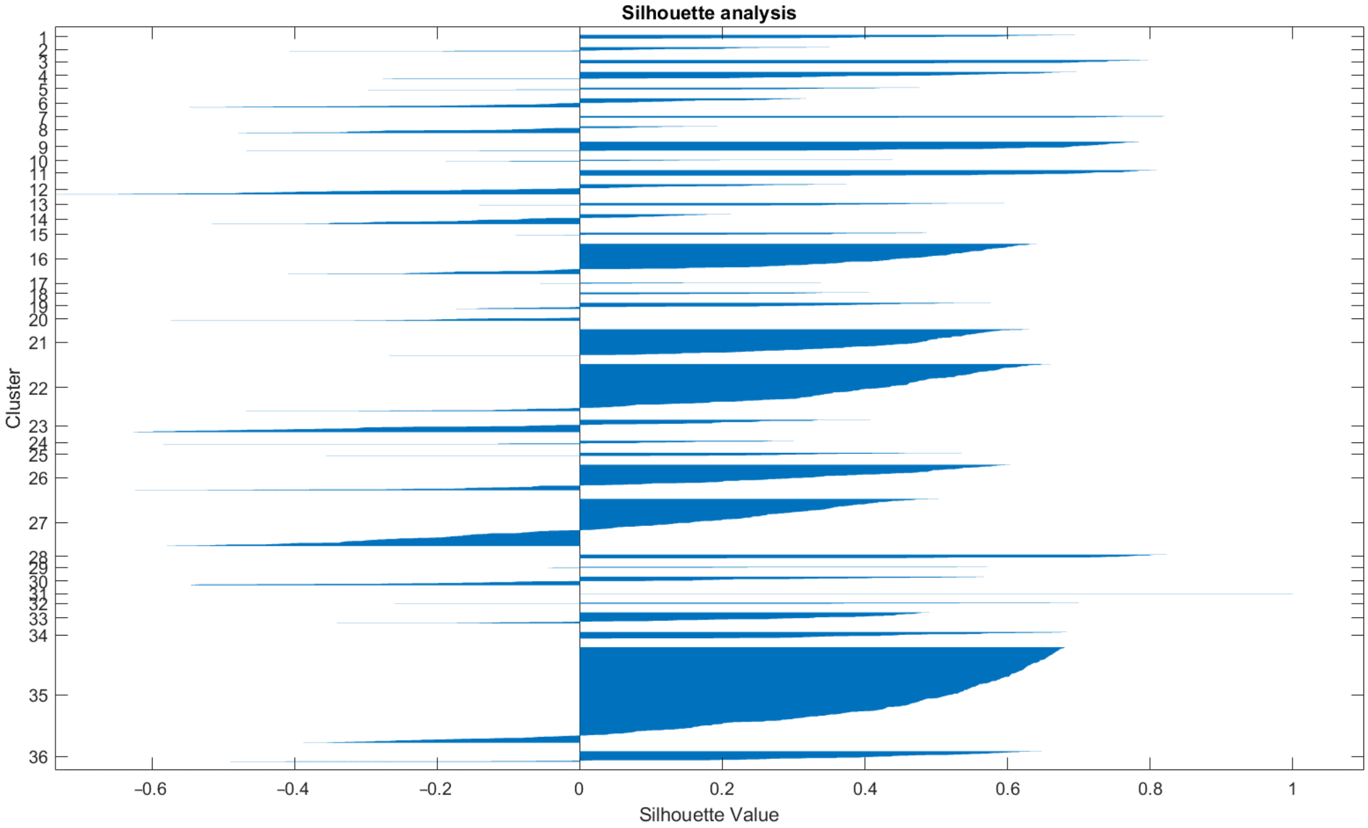


Figure 14. Silhouette analysis, hierarchical algorithm (Ward's method). Test conducted on 36 clusters. Mean silhouette value = 0.2.

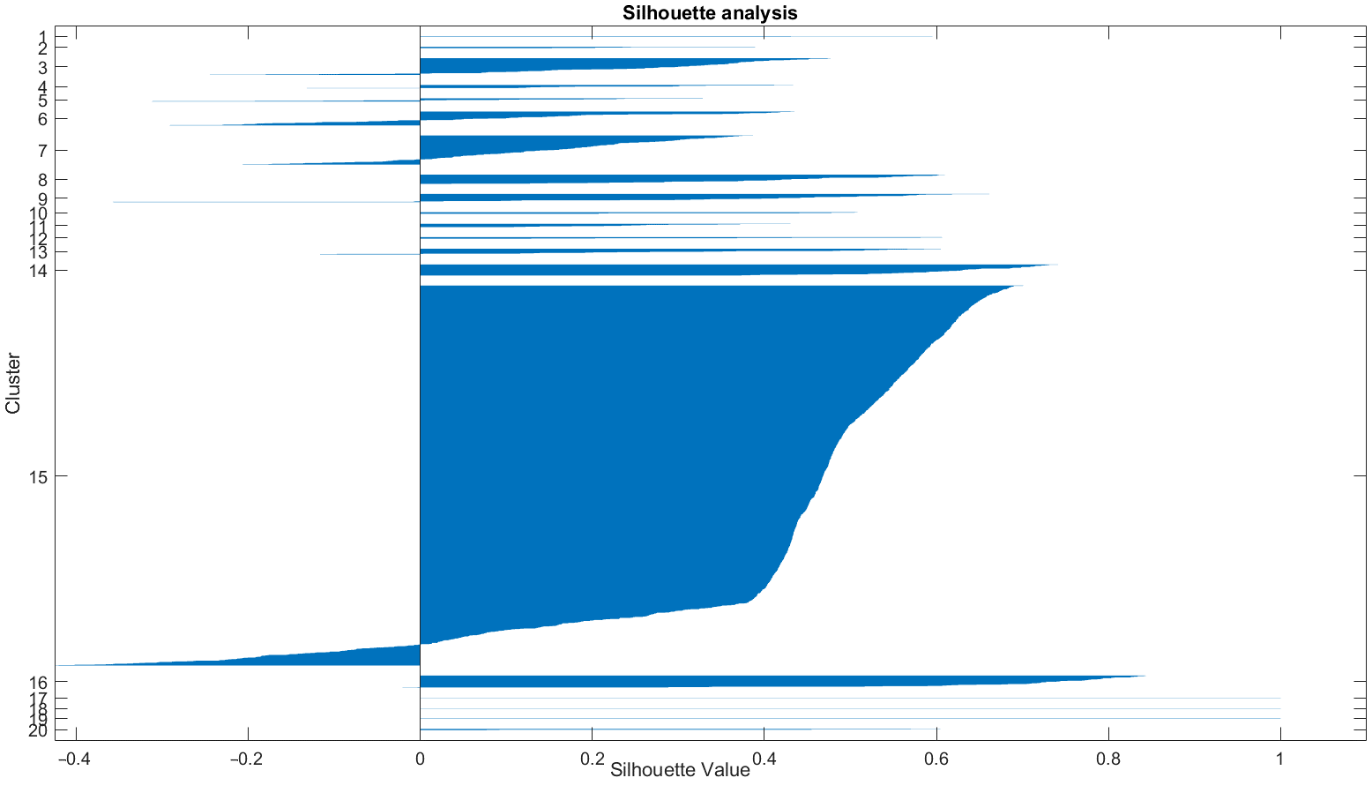


Figure 15. Silhouette analysis, hierarchical algorithm (complete linkage technique). Test conducted on 20 clusters. Mean silhouette value = 0.3.

The analysis of 20 clusters revealed similar efficiency issues as the SOM analysis conducted using the same parameters (Figure 15).

Clusters 3, 4, 6, 7, and 12 showed combined artifacts from different tool categories (cluster 3, Supplementary Materials: Figure S20a,b), while clusters 17, 18, and 19 each showed only a single item (cluster 17, Supplementary Materials: Figure S21). Only a few clusters, such as 14, 15, and 16, were coherent with the subdivision. While cluster 16 showed cores from all sites and cluster 15 showed the main débitage production in all sites, highlighting the main technological similarities, cluster 14 exclusively contained arrowheads from Nahal Reuel.

Notably, all algorithms set to either 36 or 20 clusters identified a unique cluster made of specific arrowheads from Nahal Reuel, suggesting the possibility of a local cultural preference within a wider homogenous material culture which all sites belong to as part of the Middle Pre-Pottery B cultural koiné.

4. Discussion

The utilization of various algorithms and techniques yielded promising results. The SOM demonstrated superior accuracy and precision compared to the K-Means and hierarchical algorithms. Notably, the SOM test set with 36 clusters outperformed both the tests with 20 clusters and the other algorithms. Overall, the tests with 20 clusters yielded less precise and efficient results, except for the K-Means++ application, which presented contrasting outcomes. However, all tests with 36 or 20 clusters successfully identified technological and typological clusters to varying degrees.

Our study observed that different algorithms displayed varying proficiency levels in decoding the archaeological dataset. It is crucial to analyze and describe their performance in detail. We found that the algorithms set on 36 clusters could identify distinct, coherent clusters more accurately, whereas the algorithms set on 20 clusters tended to combine different classes of elements. This consistent behavior was observed across multiple runs, indicating an inherent characteristic in the algorithms' approach to interpreting the dataset. It became apparent that more clusters were essential for effectively deciphering the homogeneous complete lithic assemblage.

Additionally, it is noteworthy that each test conducted using a specific algorithm and set of clusters consistently produced groups comprising the same elements, with only minor and insignificant exceptions. Moreover, when we subdivided the dataset into major categories and conducted separate tests, the algorithms demonstrated greater precision and efficiency, even when set at fewer clusters. This recurring behavior was observed across all algorithms and parameters, indicating a consistent pattern in the clustering results.

In conclusion, the SOM technique demonstrated an impressive capacity to comprehend the intricate steps involved in lithic production, including the chain of operations, technological nuances, and typological characteristics. Given its exceptional proficiency in deciphering homogeneous archaeological datasets, we highly recommend using the SOM technique in such contexts. It is important to note that the optimal number of clusters may vary depending on the specific context, anticipated clusters, and the nature of the data. Moreover, the efficacy of the algorithms mentioned above hinges significantly on the number of variables (archaeological parameters/features) assessed for each artifact.

We therefore suggest analyzing the entire assemblage with the SOM method first and then running different tests for each subdivision made, depending on the type of dataset and its possible internal subdivisions. This approach may yield promising results by achieving greater clarity and precision.

5. Conclusions

In conclusion, the comparative analysis of various SOM, K-Means, and hierarchical algorithms and their respective techniques has yielded noteworthy and promising findings. Each approach was systematically evaluated and interpreted within an archaeological framework to elucidate the decoding process inherent to each algorithm. Overall, the tests demonstrated varying degrees of efficacy and precision in comprehending a homogenous archaeological dataset. Of the algorithms tested, the SOM algorithm emerged as the most effective and precise, capable of not only providing a comprehensive overview of the technological and typological characteristics of the MPPNB lithic industries of Nahal Yarmuth 38, Motza, Yiftahel, and Nahal Reuel but also delving deeper to yield more coherent and precise clusters. It successfully differentiated the principal technological and typological groups within the lithic assemblages into distinct sub-categories. Additionally, a unique cluster exclusively comprising arrowheads from one of the selected sites was identified in each application, suggesting a distinct cultural preference or even a sub-cultural aspect within the Levantine MPPNB cultural koiné.

Notably, the SOM exhibited the fewest decoding errors compared to the hierarchical and K-Means algorithms and their associated applications, such as Ward's method, complete linkage, K-Means++, and the PCA technique, demonstrating lower overall efficiency and precision levels. These results underscore the potential of the SOM algorithm in decoding and analyzing archaeological datasets, particularly those pertaining to lithic industries.

This study may enable the exploration of new frontiers in archaeology.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics13142752/s1.

Author Contributions: Conceptualization, E.N.; methodology, M.T., E.N. and F.G.; formal analysis, M.T. and E.N.; results, E.N.; data collection, E.N.; writing—original draft preparation, M.T., E.N. and F.G.; visualization, F.M. and M.M.; supervision, C.C.B. and F.F.; project administration, C.C.B. and F.F.; funding acquisition, F.F. All authors have read and agreed to the published version of the manuscript.

Funding: F.F. acknowledges the partial funding of the research activity by the European Union-The National Recovery and Resilience Plan (NRRP)–Mission 4 Component 2 Investment 1.4-NextGeneration EU Project-Project "National Centre for HPC Big Data & Quantum Computing"-CN00000013-CUP B83C22002940006-Spoke 6.

Data Availability Statement: Data are shown within the text and in the Supplementary Materials.

Acknowledgments: We thank Avi Gopher for his remarkable suggestions and sage guidance and for letting us analyze samples from Nahal Yarmuth 38. We would also like to thank Hamoudi Khalaily, Kobi Vardi, and the Israel Antiquities Authority for making Motza's samples available and Ofer Marder for allowing us to study materials from his fieldwork at Yiftahel. Finally, we thank Natalia Gubenko for her availability and support in providing the samples at the Israel Antiquities Authority center. The Nahal Reuel assemblage was excavated by the late A. Ronen.

Conflicts of Interest: The authors do not have any conflicts of interest to declare.

References

1. Guyot, A.; Lennon, M.; Lorho, T.; Hubert-Moy, L. Combined Detection and Segmentation of Archeological Structures from LiDAR Data Using a Deep Learning Approach. *J. Comput. Appl. Archaeol.* **2021**, *4*, 1. [CrossRef]
2. Scotland, A.; Øivind, T.; David, C.; Waldeland, U.A. Using deep neural networks on airborne laser scanning data: Results from a case study of semi-automatic mapping of archaeological topography. *Archaeol. Prospect.* **2018**, *26*, 165–175. [CrossRef]
3. Caspari, G.; Crespo, B. Convolutional neural networks for archaeological site detection—Finding "princely" tombs. *J. Archaeol. Sci.* **2019**, *110*, 104998. [CrossRef]
4. Davis, D.S.; Caspari, G.; Lipo, C.P.; Sanger, M.C. Deep Learning Reveals Extent of Archaic Native American Shell-Ring Building Practices. *J. Archaeol. Sci.* **2021**, *132*, 105433. [CrossRef]
5. Küçükdemirci, M.; Sarris, A. Deep learning-based automated analysis of archaeo-geophysical images. *Archaeol. Prospect.* **2020**, *27*, 107–118. [CrossRef]
6. Trier, Ø.D.; Reksten, J.H.; Løseth, K. Automated mapping of cultural heritage in Norway from airborne lidar data using faster R-CNN. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *95*, 102241. [CrossRef]
7. Cole, K.E.; Yaworsky, P.M.; Hart, I.A. Evaluating statistical models for establishing morphometric taxonomic identifications and a new approach using Random Forest. *J. Archaeol. Sci.* **2022**, *143*, 105610. [CrossRef]
8. Eberl, M.; Bell, C.S.; Spencer-Smith, J.; Raj, M.; Sarubbi, A.; Johnson, P.S.; Rieth, A.E.; Chaudhry, U.; Aguila, R.E.; McBride, M. Machine Learning–Based Identification of Lithic Microdebitage. *Adv. Archaeol. Pract.* **2023**, *11*, 152–163. [CrossRef]

9. Gualandi, M.L.; Gattiglia, G.; Anichini, F. An Open System for Collection and Automatic Recognition of Pottery through Neural Network Algorithms. *Heritage* **2021**, *4*, 140–159. [CrossRef]
10. Troiano, M.; Nobile, E.; Mangini, F.; Mastrogiuseppe, M.; Conati Barbaro, C.; Frezza, F. A Comparative Analysis of the Bayesian Regularization and Levenberg–Marquardt Training Algorithms in Neural Networks for Small Datasets: A Metrics Prediction of Neolithic Laminar Artefacts. *Information* **2024**, *15*, 270. [CrossRef]
11. Nobile, E.; Conati, C.B. The Standardisation of the PPNB Lithic Industry from Er-Rahib. *Orig. Rev. Prehistory Protohistory Anc. Civiliz.* **2022**, *46*, 7–28.
12. Reeler, C. Neural networks and fuzzy logic analysis in archaeology. In *Archaeology in the Age of the Internet. CAA9, Proceedings of the 25th Anniversary Conference, University of Birmingham (BAR International Series 750, CD-ROM), Birmingham, UK, 10–13 April 1997*; Dingwall, L., Exon, S., Gaffney, V., Laflin, S., van Leusen, M., Eds.; Computer Applications and Quantitative Methods in Archaeology; Archaeopress: Oxford, UK, 1999.
13. Nicolucci, F.; D'Andrea, A.; Crescioli, M. Archaeological Applications of Fuzzy Databases. In *Computing Archaeology for Understanding the Past. CAA 2000. Computer Applications and Quantitative Methods in Archaeology, Proceedings of the 28th Conference, Ljubljana, Slovenia, 18-21 April 2000*; Stančič, Z., Veljanovski, T., Eds.; BAR International Series 931; Archaeopress: Oxford, UK, 2001; pp. 107–116.
14. Niccolucci, F.; Hermon, S. A fuzzy logic approach to reliability in archaeological virtual reconstruction. In Proceedings of the CAA 2004, Prato, Italy, 13–17 April 2004; Archaeolingua: Budapest, Hungary, 2004; pp. 28–35.
15. Baxter, M.J. A Review of Supervised and Unsupervised Pattern Recognition in Archaeometry. *Archaeometry* **2006**, *48*, 671–694. [CrossRef]
16. Baxter, M.J. Archaeological Data Analysis and Fuzzy Clustering. *Archaeometry* **2009**, *51*, 1035–1054. [CrossRef]
17. Horr, C.; Lindinger, E.; Brunnet, G. Machine learning based typology development in archaeology. *ACM J. Comput. Cult. Herit.* **2014**, *7*, 2. [CrossRef]
18. Parisotto, S.; Leone, N.; Schönlieb, C.-B.; Launaro, A. Unsupervised clustering of Roman potsherds via Variational Autoencoders. *J. Archaeol. Sci.* **2022**, *142*, 105598. [CrossRef]
19. Qubaa, A. Al-Hamdani, S. Detecting abuses in archaeological areas using k-mean clustering analysis and UAVs/drones data. *Sci. Rev. Eng. Environ. Sci.* **2021**, *30*, 182–194.
20. El-Hajj, H. Interferometric SAR and Machine Learning: Using Open Source Data to Detect Archaeological Looting and Destruction. *J. Comput. Appl. Archaeol.* **2021**, *4*, 47–62. [CrossRef]
21. Cicchitelli, G.; D'urso, P.; Minozzo, P. *Statistica: Principi E Metodi, 3 ed.*; Pearson: Milano, Italy, 2017.
22. Cochran, W.G. *Sampling Techniques*; Harvard University, John Wiley & Sons: Hoboken, NJ, USA, 1977.
23. Kish, L. *Survey Sampling*; Wiley: Hoboken, NJ, USA, 1965.
24. Gopher, A. *Arrowheads of the Neolithic Levant: A Seriation Analysis*; Eisenbrauns: Winona Lake, IN, USA, 1994.
25. Rollefson, G.O. The Late Aceramic Neolithic of the Levant: A Synthesis. *Paléorient* **1989**, *15*, 168–173. [CrossRef]
26. Kozlowski, S.; Aurenche, O. Territories, Boundaries and Cultures in the Neolithic Near East Archaeopress. In *Maison de l'Orient et de la Méditennanée*; BAR-International Series 1362; British Archaeological Reports: Oxford, UK, 2005.
27. Shea, J.J. *Stone Tools in the Palaeolithic and Neolithic Near East: A Guide*; Cambridge University Press: New York, NY, USA, 2013.
28. Barket, T.M. *The Tool Kit of Daily Life: Flaked-Stone Production at the Household Level at the Neolithic Site of 'Ain Ghazal, Jordan*; University of California: Riverside, CA, USA, 2016.
29. Arimura, M. *The Neolithic Lithic Industry at Tell Ain El-Kerkh*; Archaeopress Archaeology: Oxford, UK, 2020.
30. Arzarello, M.; Fontana, F.; Peresani, M. *Manuale di Tecnologia Litica Preistorica*; Carocci Editore: Roma, Italiy, 2015.
31. Tixier, J. *Typologie De L'epipaleolithique Du Maghreb*; Arts et Metiers Graphiques: Paris, France, 1963.
32. Boeda, E. Techno-logique & Technologie Une Paléo-Histoire des Objets Lithiques Tranchants: Prehistoire au Present Archeo-Edtions. 2013. Available online: https://www.decitre.fr/livres/technologique-technologie-9782364610033.html (accessed on 6 June 2024).
33. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 59–69. [CrossRef]
34. Kohonen, T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480. [CrossRef]
35. Kohonen, T. Essentials of the self-organizing map. *Neural Netw.* **2013**, *37*, 52–65. [CrossRef] [PubMed]
36. Morimoto, H. Hidden Markov models and self-organizing maps applied to stroke incidence. *Open J. Appl. Sci.* **2016**, *6*, 158–168. [CrossRef]
37. Hazrati Yadkoori, S.; Datta, B. Adaptive surrogate model based optimization (ASMBO) for unknown groundwater contaminant source characterizations using self-organizing maps. *J. Water Resour. Prot.* **2017**, *9*, 193–214. [CrossRef]
38. Huneiti, A.; Daoud, M. Content-based image retrieval using SOM and DWT. *J. Softw. Eng. Appl.* **2015**, *8*, 51. [CrossRef]
39. Upadhyay, P.K.; Sinha, R.K.; Karan, B.M. Predicting heat-stressed EEG spectra by self-organising feature map and learning vector quantizers—SOFM and LVQ based stress prediction. *J. Biomed. Sci. Eng.* **2010**, *3*, 529. [CrossRef]
40. Vesanto, J.; Himberg, J.; Alhoniemi, E.; Parhankangas, J. Self-organizing map in Matlab: The SOM Toolbox. *Proc. Matlab DSP Conf.* **1999**, *99*, 16–17.
41. Silva, L.A.; Pazzinato, B.; Coelho, O.B. Image Representation Using the Self-Organizing Map. In Proceedings of the Advances in Self-Organizing Maps: 9th International Workshop, WSOM 2012, Santiago, Chile, 12–14 December 2012; Springer: Berlin/Heidelberg, Germany, 2013; pp. 135–143.

42. Kumar, D.I.; Kounte, M.R. Comparative study of self-organizing map and deep self-organizing map using MATLAB. In Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6–8 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1020–1023.
43. Vesanto, J.; Alhoniemi, E. Clustering of the self-organizing map. *IEEE Trans. Neural Netw.* **2000**, *11*, 586–600. [CrossRef]
44. Natita, W.; Wiboonsak, W.; Dusadee, S. Appropriate learning rate and neighborhood function of self-organizing map (SOM) for specific humidity pattern classification over Southern Thailand. *Int. J. Model. Optim.* **2016**, *6*, 61. [CrossRef]
45. Dragomir, O.E.; Dragomir, F.; Radulescu, M. Matlab application of Kohonen self-organizing map to classify consumers' load profiles. *Procedia Comput. Sci.* **2014**, *31*, 474–479. [CrossRef]
46. Na, S.; Xumin, L.; Yong, G. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2–4 April 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 63–67.
47. Ghazal, T.M. Performances of k-means clustering algorithm with different distance metrics. *Intell. Autom. Soft Comput.* **2021**, *30*, 735–742. [CrossRef]
48. Kapil, S.; Chawla, M. Performance evaluation of K-means clustering algorithm with various distance metrics. In Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 4–6 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–4.
49. Murtagh, F.; Contreras, P. Algorithms for hierarchical clustering: An overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2*, 86–97. [CrossRef]
50. Nazari, Z.; Kang, D.; Asharif, M.R.; Sung, Y.; Ogawa, S. A new hierarchical clustering algorithm. *Int. Conf. Intell. Inform. Biomed. Sci.* **2015**, *201*, 148–152.
51. Géron, A. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.