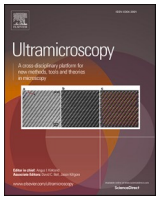
Ultramicroscopy 253 (2023) 113828

Contents lists available at ScienceDirect

Ultramicroscopy

journal homepage: www.elsevier.com/locate/ultramic



Comparative of machine learning classification strategies for electron energy loss spectroscopy: Support vector machines and artificial neural networks

Daniel del-Pozo-Bueno [a,b,*], Demie Kepaptsoglou [c,d], Francesca Peiró [a,b], Sònia Estradé [a,b]

[a] Departament d'Enginyeria Electrònica i Biomèdica, LENS-MIND, Universitat de Barcelona, Barcelona 08028, Spain
[b] Institute of Nanoscience and Nanotechnology (IN2UB), Universitat de Barcelona, Barcelona 08028, Spain
[c] SuperSTEM, Sci-Tech Daresbury Campus, Daresbury WA4 4AD, UK
[d] School of Physics, Engineering and Technology, University of York, Heslington YO10 5DD, UK

ARTICLE INFO

Keywords:
Electron energy loss spectroscopy
Machine learning
Support vector machines
Artificial neural networks
Transition metals
Oxidation state

ABSTRACT

Machine Learning (ML) strategies applied to Scanning and conventional Transmission Electron Microscopy have become a valuable tool for analyzing the large volumes of data generated by various S/TEM techniques. In this work, we focus on Electron Energy Loss Spectroscopy (EELS) and study two ML techniques for classifying spectra in detail: Support Vector Machines (SVM) and Artificial Neural Networks (ANN). Firstly, we systematically analyze the optimal configurations and architectures for ANN classifiers using random search and the tree-structured Parzen estimator methods. Secondly, a new kernel strategy is introduced for the soft-margin SVMs, the cosine kernel, which offers a significant advantage over the previously studied kernels and other ML classification strategies. This kernel allows us to bypass the normalization of EEL spectra, achieving accurate classification. This result is highly relevant for the EELS community since we also assess the impact of common normalization techniques on our spectra using Uniform Manifold Approximation and Projection (UMAP), revealing a strong bias introduced in the spectra once normalized. In order to evaluate and study both classification strategies, we focus on determining the oxidation state of transition metals through their EEL spectra, examining which feature is more suitable for oxidation state classification: the oxygen K peak or the transition metal white lines. Subsequently, we compare the resistance to energy loss shifts for both classifiers and present a strategy to improve their resistance. The results of this study suggest the use of soft-margin SVMs for simpler EELS classification tasks with a limited number of spectra, as they provide performance comparable to ANNs while requiring lower computational resources and reduced training times. Conversely, ANNs are better suited for handling complex classification problems with extensive training data.

1. Introduction

Scanning and Transmission Electron Microscopy (S/TEM) has witnessed an increase in imaging and spectroscopically generated data. Since the correctors were introduced in the 2000s, technological advancements (such as detectors, electron guns, monochromators, etc.) and an increase in computing power have enabled the application of techniques that were theoretically conceived but not yet feasible to implement, such as 4D STEM, Differential Phase Contrast (DPC), atomically resolved analytical and fast tomography, ultrafast TEM, or photon-induced near-field electron microscopy. Some of these techniques, such as 4D STEM, high resolution Electron Energy Loss Spectroscopy (EELS) or DPC, easily generate gigabytes per image. This issue has made the S/TEM community aware of the need to develop new analysis tools to process these vast amounts of data precisely and rapidly.

In this sense, Machine Learning (ML) has become one of the most promising fields for handling large amounts of data [1]. As a result, many ML algorithms have been applied to various Transmission Electron Microscopy (TEM) techniques for various purposes [2]. In order to gain a comprehensive understanding of how these algorithms can be utilized for analyzing data, we categorize ML algorithms based on their degree of supervision: unsupervised algorithms, semi-supervised algorithms, reinforcement algorithms and supervised algorithms. In this

* Corresponding author at: Departament d'Enginyeria Electrònica i Biomèdica, LENS-MIND, Universitat de Barcelona, Barcelona 08028, Spain.
E-mail address: dpb.523@gmail.com (D. del-Pozo-Bueno).

https://doi.org/10.1016/j.ultramic.2023.113828
Received 25 April 2023; Received in revised form 26 July 2023; Accepted 2 August 2023
Available online 3 August 2023

classification, the focus is on the type of data required to use them, i.e., supervised algorithms need known data with known labels in order to train the algorithm and so use it to analyze unknown data. Alternatively, we find unsupervised algorithms, which do not require known data (labels) to analyze the unknown data. In between them, there are semi-supervised and reinforcement algorithms, which require a minimum amount of known or labeled data for a good performance.

Following this classification, the algorithms applied to S/TEM are presented based on the type of data required. Historically, the first algorithms applied to TEM were unsupervised ones and they were mainly applied to process EELS data, since they present a larger complexity than other signals acquired from the TEM. In particular, the first algorithms were the dimensional reduction algorithms, i.e., those that are capable of reducing the dimensionality of the data to facilitate its interpretation. They have been applied for multiple tasks such as to eliminate noise or to identify signals in EELS Spectrum Images (SI). Such an implementation can be found in the first works of N. Bonnet et al. where Principal Component Analysis (PCA) and Independent Component Analysis (ICA) were introduced in the TEM community to extract relevant information from EEL spectra [3,4]. From this initial works plenty of research has been conducted by means of those algorithms [5,6] like those of F. del la Peña et al. where the ICA was used to map titanium oxides phases [7], the works of Yedra et al. where the PCA and ICA were used for noise reduction and compositional identification in EELS tomographic reconstruction for nanostructured materials [8,9], the work of S. Lichtert et al. where the PCA was used as a noise filter in SIs [10], the work of Eljarrat et al. where the PCA, blind source separation and ICA was applied to retrieve the electronic properties of silicon nanocrystals [11], the work of Martineau et al. and Åhes et al. where ICA and Non-Negative Matrix Factorization (NMF) were used as methods for identifying scanning precession electron diffraction data [12,13], the work of M. Pelaez-Fernandez et al. where NMF was used to identify plasmons [14], the work of J. Sunde et al. where PCA was used to map the orientation of 4D-STEM datasets [15] or the work of T. Blum for the identification of light elements by combining STEM and Energy Dispersive X-ray Spectroscopy (EDS) images using singular value decomposition and ICA [16]. Recently, another dimensionality reduction technique has been adopted by the community, the Uniform Manifold Approximation and Projection (UMAP), as found in the work of Li et al. where UMAP was applied to extract atomic-resolution defect information from 4D-STEM datasets [17].

Later, continuing with the unsupervised algorithms, our group also incorporated clustering algorithms as a tool for classifying EEL SIs, algorithms such as K-means, hierarchical clustering, or Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), which have demonstrated good performance identifying spectra with different characteristics, whether for composition identification or oxidation states [18,19]. Recently published works using unsupervised Artificial Neural Networks (ANN), specifically Convolutional Autoencoders (AEC), have proven to be a good strategy for reducing noise after rapid acquisition of EEL spectra [20], identifying defects in graphene and zincblende structures [21] or systematic exploration of 4D-STEM data by using rotational invariant Variational Autoencoder (VAE) approach [22].

In the other hand, ANNs have recently played a crucial role in the development of supervised and semi-supervised algorithms. The advancements that these strategies have brought to the TEM community are manifold, and these algorithms can be applied to conventional S/TEM 2D imaging modes, i.e., those where a 2-dimensional image is acquired, and to more complex data such as EELS SIs or 4D STEM images, which are formed by 3 and 4 dimensions, respectively. This division is really relevant, as the complexity of the algorithms applied is related to the complexity of the data.

In conventional S/TEM 2D imaging modes, such as High-Resolution TEM (HRTEM) or High-Angle Annular Dark Field (HAADF), ML strategies have been extensively applied for image identification, e.g., identifying features in nanostructured materials, NanoParticle (NP) shapes, atomic positions, and defects. For these tasks, Convolutional Neural Networks (CNN) serve as the primary strategy. They have been applied in numerous cases including the classification and identification of nanostructures through Bright Field (BF) and Dark Field (DF) images [23,24], metrology study of NPs in BF and DF images [25–27], atomic-scale detection on S/TEM [28,29], atomic-scale defects and distortions detection on S/TEM images [30–32], or even in-situ TEM videos where iron irradiation effects were identified [33]. Although supervised strategies offer many applications and benefits in the analysis of conventional S/TEM, they may be more effective when analyzing more complex data. In this regard, there are works for 4D STEM that also use CNNs for 2D phase retrieval and phase object reconstructions [34], or some works in EELS, where Artificial Neural Networks (ANN) and Support Vector Machines (SVM) are used for transition metal oxidation state identification [35,36].

In this study, we conduct a comparison between soft-margin SVMs and ANNs in classifying EELS spectra, in particular the oxidation state of transition metal oxides based on their characteristics. We aim to distinguish the oxidation state not only from the transition metal white lines, but also from the oxygen K edge, or from the entire spectrum encompassing both features. To achieve this comparison, we propose a set of ANN architectures, testing different types of layers and conducting a systematic search in the parameter space using Random Search (RS) and Tree-structured Parzen Estimator (TPE) algorithms. These search algorithms allow us to determine the most effective combination of parameters and architecture in constructing EELS classifiers. Both algorithms are also compared in terms of computational resource expenditure to determine which one offers the best performance. It is noteworthy that previous studies have already demonstrated the success of soft-margin SVMs and ANNs in classifying the oxidation state of Iron (Fe) and Manganese (Mn) oxides based on their white lines [35,36], including our previous work. Subsequently, the impact of normalization techniques on EELS spectra is analyzed through the utilization of the UMAP dimensional reduction algorithm. This algorithm has been previously demonstrated by our group to effectively reduce the complexity of EELS spectra while preserving their proximity structure [18]. In this sense, a new kernel strategy for the soft-margin SVMs is also proposed: the cosine kernel, which eliminates the need for normalizing the data in the training and classification of spectra, streamlining the preprocessing process. Finally, strategies to enhance the robustness of SVM algorithms versus energy shifts in the edges will be explored.

2. Theoretical background

2.1. Support vector machines

Support Vector Machines (SVMs) are a type of supervised ML algorithm used for classification and regression analysis. They were first introduced in the 1990s as a method for solving binary classification problems [37]. The main idea behind SVMs is to find the maximum margin hyperplane that separates the data into different classes. The margin is the distance between the hyperplane and the closest data points, referred to as support vectors. SVMs are effective in handling high-dimensional data, and can handle non-linear relationships between the input variables and the output class labels through the use of kernel functions. These functions transform the input data into a higher-dimensional space, where a linear separation can be achieved. The SVMs offer a versatile and robust approach to supervised learning, which have been applied in a wide range of applications, including image classification, bioinformatics, and natural language processing [38].

The version used in this article corresponds to the soft-margin SVM which is a variation of the standard algorithm, that allows for misclassified samples by introducing a cost parameter balancing the margin maximization and the number of misclassified samples. In these cases, a

hard-margin SVM, which seeks to find a hyperplane that perfectly separates the classes, may not be suitable as it is prone to overfitting the data. The soft-margin SVM addresses this issue by allowing for some misclassifications, referred to as slack variables, in exchange for a wider margin. This trade-off is controlled by a parameter called "C", which determines the balance between the margin width and the number of misclassified data points. A high value of C means that a more accurate classification is required (smaller margin), which in turn means that fewer misclassified data points are allowed. On the other hand, a low value of C means that less accurate classification is allowed (more relaxed margin), which in turn means that more misclassified data points are allowed in exchange for a wider margin. In other words, a high C is associated with a higher penalty for misclassified data points, while a low C is associated with a lower penalty.

When the SVM is trained, an optimization problem is solved to find the hyperplane that best separates the data into the desired classes. This optimization problem seeks to maximize the margin between the classes while minimizing the classification error, and is formulated as a Quadratic Programming (QP) problem. The objective function of this QP problem is to minimize the sum of the squared distances from the data points to the hyperplane, subject to the constraint that the margin is at least as large as the tolerance for misclassifications. The complexity of this optimization problem depends on the size and complexity of the data, as well as on the choice of the kernel function used to transform the data into a higher-dimensional space. To solve this QP problem, the optimization algorithm used is the Sequential Minimal Optimization (SMO) [39] which is designed to be efficient for large-scale datasets.

2.2. Artificial neural networks

ANNs, also referred to as Neural Networks (NN), are a family of ML algorithms inspired by the structure and function of biological neurons. The aim of these algorithms is to replicate the ability of the human brain to learn from experience and make predictions based on that learning. ANNs are comprised of interconnected artificial neurons, organized into layers, that process and transmit information through weighted connections. Each neuron receives inputs from other neurons, processes the inputs and the activation function to produce an output signal. These output signals are then passed on to the next layer until a final output prediction is produced.

To understand ANNs, it is crucial to consider the key components that play an important role in the correct application of the network to a specific problem. These components include the activation functions, loss functions, optimization algorithms, and network architecture. Activation functions introduce non-linearity into the model, allowing the network to learn complex relationships between inputs and outputs. Common activation functions include the sigmoid function, hyperbolic tangent (Tanh), Rectified Linear Unit (ReLU), leaky ReLU, Exponential Linear Unit (ELU), or Scaled Exponential Linear Unit (SELU). The choice of activation function depends on the task and type of network.

Loss functions measure the difference between the predicted output and actual output, with the goal of the training process being to minimize the value of the loss function. This is achieved by adjusting the weights of the connections between the neurons using optimization algorithms. The backpropagation algorithm is a widely used algorithm for training ANNs, which computes the gradient of the loss function with respect to the weights to compute gradient descent updates. This algorithm uses optimization algorithms to update the weights in the direction of the steepest descent of the loss function, with the learning rate determining the step size of the updates. Various optimization algorithms can be used in ANNs, including Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (ADAM), Root Mean Square Propagation (RMSProp), and others. The choice of optimization algorithm also depends on the specific problem and type of network being used.

The architecture of the network, including the number of hidden layers, the number of neurons in each layer, the type of transformation applied per each neuron or the way of connected the layers, also plays a crucial role in the network's performance. The optimal network architecture is often determined through trial and error and experimentation [40], with a common approach being to start with a simple architecture and gradually increasing complexity until performance plateaus.

In recent years, ANNs have become a widely used tool for solving a variety of problems, leading to the development of various type of networks, including Feedforward Neural Networks (FNN), CNNs, Generative Adversarial Networks (GANs), Autoencoders (AE), and Recurrent Neural Networks (RNN). The FNNs are the simplest type of ANN, with information flowing only in one direction from the input layer to the output layer without forming loops. The CNNs are a specific type of FNN specifically designed for image processing tasks and use convolutional layers to apply filters to the input data and extract relevant features. The AE are used for unsupervised learning tasks, where the goal is to encode and decode data in order to learn a compact representation. The GANs are used for generative tasks, with the goal of generating new data samples that are similar to the input data. Finally, the RNNs are designed for tasks where the input data are processed over time and the network state is influenced by past inputs. Regarding this work, we mainly use the FNN and CNN networks, which are the most commonly used for classification tasks.

As we use FNNs and CNNs throughout this work, we focus on these. From a mathematical standpoint, the operation carried out by them is as follows:

$$output = activation(W * input + b) \tag{1}$$

where:

- W is a weight matrix that connects the neurons of the current layer with the neurons of the previous layer.
- $input$ is the input vector coming from the previous layer.
- b is the bias vector for each neuron in the dense layer.
- $activation$ is an activation function, such as ReLU, Sigmoid, Tanh, etc. that is applied to each neuron.

ANNs have proven to be a powerful tool in the field of ML and have been applied in a wide range of applications, including computer vision, speech recognition, and natural language processing [41].

2.3. Uniform manifold approximation and projection for dimension reduction

The UMAP is a state-of-the-art graph-based dimensionality reduction algorithm that has been increasingly used in recent years for visualizing and analyzing high-dimensional data [42]. It is a non-linear technique that aims to preserve both the local and global structure of the data while reducing its dimensionality to a low-dimensional space that can be easily visualized and interpreted.

The algorithm begins by constructing a high-dimensional graph representation of the data, which is then optimized to be as structurally similar as possible to a low-dimensional graph. To construct the initial high-dimensional graph, UMAP builds essentially a weighted graph, where the edge weights represent the likelihood that two points are connected. To determine connectedness, UMAP extends a radius from each point and connects points when their radii overlap. The choice of radius is crucial, as too small a radius can lead to an isolated plethora of very small clusters, while too large a radius will result in the connecting of all points. To address this challenge, UMAP chooses the radius locally based on the distance to each point's nth nearest neighbor. The graph is then made "fuzzy" by decreasing the likelihood of connection as the radius grows. Finally, by ensuring that each point is connected to at least its closest neighbor, UMAP balances the preservation of local and global structure.

The algorithm is controlled by two parameters: the number of neighbors and the minimum distance. These parameters effectively determine the balance between the local and global structure in the final projection. The number of neighbors, which corresponds to the approximate number of nearest neighbors used to construct the initial high-dimensional graph, is the most important parameter. A low number of neighbors will result in UMAP focusing more on local structure by limiting the number of neighboring points considered when analyzing high-dimensional data, while a high number will lead to UMAP prioritizing the overall structure at the expense of fine details. The minimum distance parameter, on the other hand, determines the tightness of clustering of points in the low-dimensional space. Lower values will result in tightly packed embeddings, while higher values will result in a more relaxed grouping of points, with a focus on preserving the broad topological structure.

UMAP has been proven effective in preserving the structure of various data types, including images, text, and gene expression data. It has also shown to outperform other dimension reduction algorithms, such as t-SNE, in terms of speed, scalability, and accuracy.

3. Methodology

3.1. Dataset construction

In this study, the oxidation state of transition metal oxides, particularly Fe and Mn oxides, is examined. The study focuses on two features of these compounds in EELS: the transition metal white lines and the oxygen K-edge. In Fig. 1a, these features are visualized for two Fe oxides, wüstite (FeO) and magnetite (Fe_3O_4). The fine structure of the oxygen K-edges is seen to be primarily divided into 4 edges, with a pre-edge, a main edge, and two subsequent edges of lower intensity. The main difference between the two oxidation states is seen in the intensity of the pre-peak and a slight energy shift in the main peak [43]. The Fe white lines are composed by two edges: the first and most intense corresponding to the L_3 edge, with an onset value of approximately 708eV, and a less intense second edge known as L_2, with an onset value of around 721eV. It is important to note that both onset values depend on the oxidation state and the specific component under study [43,44]. Therefore, the main differences in these edges based on the oxidation state are the onset position and shape, with a clear variation in intensity of the second peak based on Fe ion oxidation state [43]. Similarly, in Fig. 1b the Mn white lines are presented, keeping the same structure of Fe white lines: first and most intense corresponding to the L_3 edge, with an onset value of approximately 640eV, and a less intense second edge known as L_2, with an onset value of around 653eV. In this case, they are also conditioned by their oxidation state, with Mn^{2+} having the lowest energy loss and Mn^{4+} having the highest energy loss [44].

To classify the oxidation state, ten spectral datasets were constructed, with a set of labels required for each dataset. The spectra employed to build the datasets were obtained from SIs of magnetite NanoCubes (NC) and Fe oxide core-shell NCs, previously studied in the works of Torruella et al. and del-Pozo-Bueno et al. [45–47], with only the signal from the nucleus corresponding to wüstite being used. Given that the NCs exhibit a core-shell structure, the wüstite signal obtained is influenced to some extent by the top and bottom magnetite shell layers. However, we utilized the spectra with the least influence from these magnetite shell layers, achieved through the application of UMAP and HDBSCAN algorithms. The extraction and labeling of these spectra were conducted using the UMAP and HDBSCAN algorithms to ensure the correct labeling of each spectrum [18]. The Mn white lines spectra were obtained from pure Mn oxides from the work of Chatzidakis et al. [36].

Simultaneously, an eleventh spectral dataset was constructed to assess the performance of the cosine kernel in classifying EEL spectra from NPs of varying sizes, resulting in different thickness, and so, intensity values. These NPs presented a circular morphology, with a range of diameters from 5 to 11 nm. These spectra were obtained from Fe_3O_4/$Mn_xFe_{3-x}O_4$ core/shell NPs, consisting of a magnetite oxide core surrounded by a Mn-ferrite shell [48]. As a result, each class within the dataset corresponds to either the Mn-ferrite shell or the magnetite core. Similar to the previous NCs, these classes were labeled using UMAP and HDBSCAN algorithms.

The preprocessing applied to the tenth initial datasets is summarized in Fig. 2, where the final datasets are marked by a square inscribed in circular nodes. A straightforward preprocessing procedure was applied, consisting of removing the background prior to the edges of interest and normalizing the resultant spectra. This was carried out using the HyperSpy module of Python [49].

All the spectra associated with Fe oxides come from intermediary Dataset W, which contains the oxygen K-edge and the Fe white lines for an energy range of 460–800 eV, corresponding to 1360 energy channels with an energy dispersion of 0.25 eV/ch. In addition, it is composed of 2744 spectra of wüstite and 1350 spectra of magnetite. From this dataset, intermediary Datasets K and L were constructed, containing the spectra with the oxygen K-edge and the Fe white lines, respectively. The

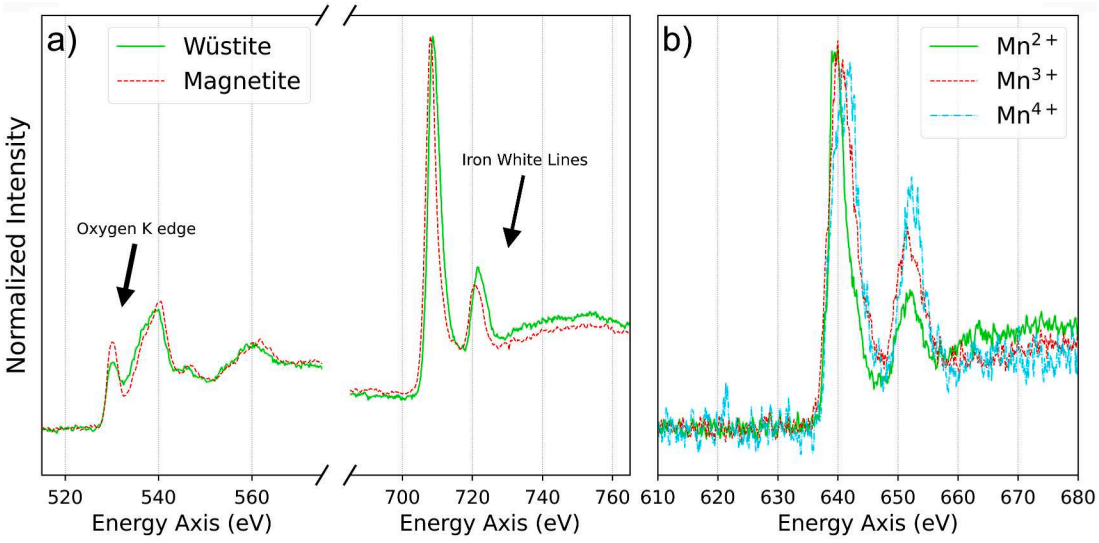


Fig. 1. (a) EELS Fe oxide spectra between 510 and 760 eV, containing the oxygen K-edge and Fe white lines. (b) EELS spectra of Mn oxides between 610 and 680 eV, containing Mn white lines.

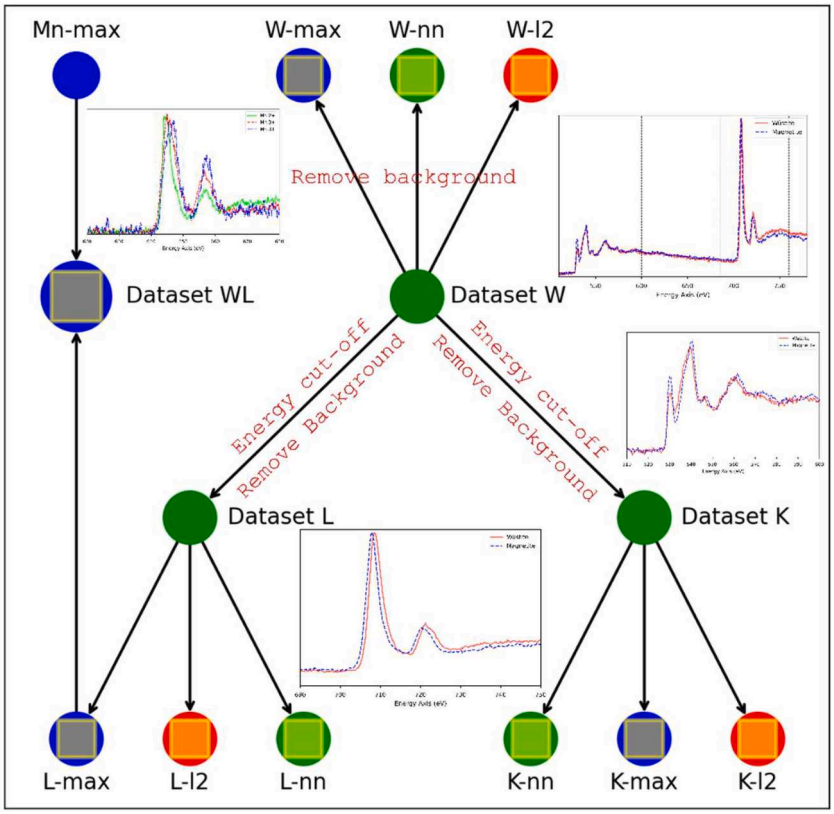


Fig. 2. Scheme of the datasets built for this work. The central nodes (Dataset W, L and K) correspond to the original spectra with an energy cut-off, and the pre-edge background removal depending on the feature in which we are interested (W = whole spectrum, L = white lines, K = oxygen K-edge). The nodes with a yellow square represent the tenth final datasets, where a normalization strategy is used, they are represented by colors: green → no-norm, red → L2-norm, blue → maximum-norm.

first one contains spectra of an energy range of 460–610 eV, corresponding to 600 energy channels, while the second contains spectra of an energy range of 600–800 eV, corresponding to 800 energy channels. The background was removed from the pre-edge of the interesting edge in the datasets W and K from 485 to 515 eV, while in the dataset L was applied from 665 to 695 eV. Three final datasets were constructed from each of these intermediary datasets according to the three normalization methods used: the L2-norm, maximum norm and no-normalization. Therefore, as an instance from the intermediary Dataset W three final datasets are obtained: one with no normalization applied (W-nn), one with L2-norm applied (W-l2), and the last one with maximum norm applied (W-max).

The WL dataset was constructed by combining intermediary Dataset L and Dataset Mn-max, which contains the Mn white lines of Mn oxides. This final dataset contains a total of 7187 spectra, composed of 2744 spectra of wüstite, 1350 spectra of magnetite, 992 spectra of Mn oxide, 841 spectra of Mn oxide III, and 1260 spectra of Mn oxide IV. The Fe oxide spectra span 700 energy channels from 612.5 to 787.5 eV with an energy dispersion of 0.25 eV/ch, while the Mn oxide spectra span 700 energy channels from 615 to 685 eV with an energy dispersion of 0.1 eV/ch. It is worth mentioning that the background removal and normalization procedures applied to the Mn oxide spectra were done in a similar manner to the Fe oxide spectra. The background was removed from 610 to 630 eV, and the spectra were normalized by the maximum norm.

The final dataset, known as the MF dataset, consists of 2458 spectra containing both Fe and Mn white lines, obtained from iron-Mn oxide core-shell NPs. The MF dataset covers the Mn and Fe white lines within an energy range of 580–784.7 eV. This range corresponds to 2048 energy channels, each with an energy dispersion of 0.1 eV/ch. Preprocessing applied to these spectra included PCA noise reduction and background removal from 590 to 620 eV. It is important to note that no normalization was performed.

Note that the energy resolutions from all the datasets have been measured from the Full-Width at Half Maximum (FWHM) of the Zero-Loss Peak (ZLP), after setting the energy dispersion in the spectrometer.

In summary, this study provides eleven final datasets comprising EEL spectra of Fe oxides (oxygen K edge and Fe white lines), Mn oxides (Mn white lines), and EEL spectra containing both Fe and Mn oxides (Mn and Fe white lines). The initial ten datasets are employed to assess the performance of supervised classification algorithms, specifically soft-margin SVMs and ANNs, in classifying the oxidation states of these transition metal oxides. The last MF dataset is used to evaluate the performance of the cosine kernel strategy for SVMs.

3.2. *Support vector machines classifiers*

From our previous work [35], the soft-margin SVM was established as an optimal classification method for detecting the oxidation state of transition metal oxides using the white lines. In this study, we introduce and assess the performance of the cosine kernel and employ it as a probabilistic classifier for SIs, as opposed to a binary one. Furthermore, we investigate various approaches to differentiate the oxidation state based on the transition metal white lines, the oxygen K edge, and the complete spectrum encompassing both features [50].

To evaluate which EEL feature is most reliable for determining the oxidation state, we adhere to the methodology outlined in our previous work [35]. First, the nine datasets constructed for this work are required, enabling us to assess the most suitable EEL feature for determining the oxidation state and the best normalization strategy for normalizing EEL spectra. Four kernel strategies are applied: linear, Radial Basis Function (RBF), sigmoid, and cosine kernel. For each dataset and model, the hyperparameters are optimized using the GridSearch algorithm, the classifiers are trained with 15% of the dataset and a 5-fold cross-validation. Finally, their performance is evaluated on the

test set, consisting of the remaining 85% of the spectra. These latter strategies aid in preventing overfitting during the training process.

The newly introduced kernel strategy, the cosine kernel, allows data normalization to be omitted. This kernel represents the cosine similarity used to compute the L2-normalized dot product of vectors. If X and Y are row vectors, then their cosine similarity, cos$D$, is represented as follows:

$$cosD(X,Y) = 1 - \cos(\theta) = 1 - \frac{X \cdot Y^T}{\| X \| \cdot \| Y \|} = 1 - \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} Y_i^2}\sqrt{\sum_{i=1}^{n} X_i^2}} \quad (2)$$

This estimator projects the vectors onto the unit sphere, and their dot product is the cosine of the angle between the vectors.

Subsequently, we examine the soft-margin SVM as a probabilistic classifier for EELS SIs, where the model output a probability score instead of a binary classification, thus providing more information about the confidence of the prediction. This probabilistic classifier is obtained by training the SVM using the Platt scaling method [51], which involves fitting a logistic regression model to the decision values produced by the SVM.

The employment of probabilistic classifiers is highly beneficial in EELS. When handling samples with compounds exhibiting mixed states, as is the case of having magnetite (Fe^{2+} and $2{\cdot}Fe^{3+}$) or Mn tetroxide (Mn^{2+} and $2{\cdot}Mn^{3+}$), the obtained spectra represent a combination of the individual spectra associated with each constituent, resulting in an intermediate state. It becomes a challenge to accurately classify the oxidation state to the corresponding spectra using traditional classification methods. Under these circumstances, probabilistic classifiers can offer significant advantages.

Probabilistic classifiers, however, are designed to manage such complexities by providing a probability distribution over the possible classes rather than a single, definitive classification. This approach allows for a more nuanced understanding of the data, as it enables the identification of the most probable classes and the corresponding confidence levels associated with each classification. By quantifying the uncertainty inherent in these intermediate states, probabilistic classifiers facilitate better decision-making and a more comprehensive interpretation of the results.

Ultimately, the cosine and probabilistic classifier are assessed, firstly, by the L-nn dataset and, in SIs of Fe oxide core/shell NCs, and secondly, by the MF dataset to demonstrate the effectiveness of the soft-margin SVM as EELS classifiers in more complex scenarios, where Fe and Mn white lines coexist within a single spectrum. In this context, the MF dataset is utilized to train a SVM classifier and directly classify EELS SIs. These SIs pertain to magnetite/Mn-Ferrite core/shell NPs of varying sizes, showcasing the effectiveness of the cosine kernel in handling intensity variability. In order to classify them, three preprocessing steps are applied. First, the PCA is used to reduce the noise in the spectra, subsequently, the NP signals and background noise are separated, resulting in the classification of only the NP signals. It is important to note that SVM is restricted by the classes, meaning that it will always assign a class for each point. Therefore, if the background is passed through, the algorithm will incorrectly assign a class to each spectrum from the background, which is nonsensical. Finally, the pre-edge background is removed for all NPs within an energy range of 580–620 eV.

3.3. Artificial neural networks classifiers

Shen et al. previous work has demonstrated that ANNs are an effective classification method for identifying the oxidation state of transition metal oxides [33], focusing on Mn oxides based on white line shapes. They evaluate popular ANN architectures and propose a fully convolutional one with better translation-invariance. In this study, we investigate various architectures for distinguishing oxidation states based on transition metal white lines, the oxygen K edge, and the full spectrum containing both features. We propose twenty-two different network architectures, including dense, convolutional, and hybrid models, to thoroughly assess the potential of ANNs as classifiers. To identify the most appropriate structure for classifying EEL spectra, we conduct an extensive parameter search using both RS and TPE methods [52,53]. Hyperparameter optimization is a critical step in the ML pipeline, enabling model parameter adjustments to improve performance. The RS method utilizes the Keras-Tuner module's RandomSearch function [54], while TPE employs the Hyperopt module [55]. The optimal architecture and hyperparameters are subsequently evaluated using Python and TensorBoard [56].

RS is a straightforward method for hyperparameter optimization, evaluating random parameter combinations to identify the best-performing set. Although efficient for low-dimensional parameter spaces, RS becomes less effective as the number of parameters and possible values increase. With a high number of parameters, the probability of finding the optimal set through random sampling diminishes significantly.

On the other hand, TPE is a more sophisticated optimization method that employs bayesian optimization to determine the next set of parameters to evaluate. The TPE algorithm constructs a probability distribution of the parameter space based on previous evaluations and uses this distribution to guide the search towards the most promising areas. This results in a more efficient search, as the algorithm can focus on parameter space regions most likely to contain the optimal set.

We explore various ANN architectures to optimize performance, categorized into two primary groups: those based on dense layers and those based on 1D convolutional layers.

For architectures based on dense layers:

0: Basic Dense Neural Networks (DNNs) consisting only of dense layers.

1–4: DNNs incorporating normalization layers, which help to stabilize and speed up the training process. These layers are incorporated in each structure as follows:

$$\begin{cases} 1: \textit{LayerNorm.} \\ 2: \textit{BatchNorm.} \\ 3: \textit{Alternanting LayerNorm. and BatchNorm.} \\ 4: \textit{Alternante BatchNorm. and LayerNorm.} \end{cases}$$

5–9: DNNs featuring dropout layers, which reduce overfitting by randomly ignoring a fraction of neurons during training, and normalization layers. These layers are incorporated in each structure as follows:

$$\begin{cases} 5: \textit{LayerNorm.} + \textit{Dropout} \\ 6: \textit{BatchNorm.} + \textit{Dropout} \\ 7: \textit{Dropout} + \textit{BatchNorm.} \\ 8: \textit{GaussianDropout} \\ 9: \textit{Dropout} \end{cases}$$

10–13: DNNs with dual branches, which are parallel pathways within the network, each one handling a different aspect of the input data, and containing BatchNormalization or Dropout layers. These layers are incorporated in each structure as follows:

$$\begin{cases} 10: \textit{LayerNorm.} \\ 11: \textit{Dropout} \\ 12: \textit{BatchNorm.} \\ 13: \textit{Basic} \end{cases}$$

For architectures based on 1D convolutional layers:

0: Convolutional Neural Networks (CNNs) with or without Pooling layers, which help to reduce the dimensions of the input data while preserving important features.
1–2: CNNs containing normalization layers, which are incorporated as follows:

$$\begin{cases} 1: \textit{BatchNorm.} \\ 2: \textit{LayerNorm.} \end{cases}$$

3: CNNs followed by dense layers, combining the advantages of both layer types.
4–6: Two-branch structures, each comprising CNN and DNN, and including normalization layers, which are incorporated as follows:

$$\begin{cases} 4: \textit{Basic} \\ 5: \textit{BatchNorm.} \\ 6: \textit{LayerNorm.} \end{cases}$$

7: Residual Neural Networks (ResNet), a type of CNN that uses shortcut connections to improve learning capabilities [57], consisting of 34 convolutional layers in our proposal.

The specific values and constraints are also presented for the various hyperparameters, which are the settings that can be adjusted to control the learning process, in Tables 1 and 2. Table 1 outlines parameters for dense structures, while Table 2 provides details for convolutional structures. Some parameters from Table 1 also appear in Table 2, as certain convolutional models incorporate dense layers within them.

Finally, once the optimal ANN models as EEL classifiers were identified, we were in a position to verify their effectiveness for classifying the oxidation state in the spectra. In order to evaluate the performance of the dense and convolutional-based architectures as classifiers with respect to the spectral region most useful for this task the 9 datasets built were used. Precautions were taken to avoid overfitting the model, particularly considering the limited amount of data available. Similarly, to the SVM, to prevent overfitting, we utilized a low number of epochs (one complete pass of training data through the algorithm), specifically 100 epochs, employed a 5-fold cross-validation to train the models, and reserved half of the data for testing the model.

3.4. Addressing energy shifts

In order to enhance the resistance of the EELS classifiers to energy shifts, we explore a method for improving the robustness of SVM and ANN models. This approach involves training the classifiers on spectra with energy shifts, ensuring that the models recognize that the onset value should not be the sole determining factor when identifying a spectrum. This strategy aims to overcome the experimental drift typically introduced in the acquired spectra as a result of the zero-loss peak displacement during SI acquisitions. To accomplish this, we have incorporated a total of 2.500 spectra, with an energy shift ranging from $[-1.75,\ 1.75]$ eV, into the dataset WL (Fe and Mn white lines). As a result, we have trained the classifiers using the shifted spectra dataset and compared their performance to the models trained solely with the WL dataset. In order to evaluate the performance of the classifiers, we have applied energy shifts to the spectra of WL dataset until reaching the desired energy shift and tested them at each energy step. This analysis aims to provide insights into the effectiveness of these classifiers in handling energy shifts, which is crucial for the accurate classification of EELS spectra in real-world applications. We must be aware that the Electron Energy-Loss Near-Edge Structure (ELNES) of transition metals white lines includes energy shifts of approximately 1 eV from one oxidation state to another [43,50].

Table 1
Hyperparameter space for dense structures in random search and TPE optimization. The table displays the minimum, maximum, and step values for the number of dense layers and the number of neurons. Additionally, it presents the options for the activation function, the output neuron activation, dropout, optimization algorithm, and learning rate.

Hyperparameters for dense structures			
	Min.	Max.	Step
Dense layers	1	10	1
Neurons	16	1024	16
	Options		
Activation	ReLU, Sigmoid, Tanh, SELU, ELU, Exponential		
Algorithm	ADAM	SGD	RMSprop
Learning Rate	$1,\ 10^{-1},\ 10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}$		
Activation exit neuron	Sigmoid or SoftMax		
Dropout	0.2, 0.3, 0.5, 0.7, 0.9		

Table 2
Hyperparameter space for convolutional structures in random search and TPE optimization. The table displays the minimum, maximum, and step values for the number of convolutional layers, the number of convolution filters, the size of the convolution kernels, the pooling size, the number of dense layers, and the number of neurons. Additionally, it presents the options for the pooling type, activation functions, optimization algorithm, learning rate, and the activation function for the output neuron.

Hyperparameters for convolutional structures			
	Min.	Max.	Step
Convolution layers	1	5	1
Convolution filters	2	16	1
Convolution kernel size	2	6	1
Pooling size	2	3	1
Dense layers	1	12	1
Neurons	16	1024	16
	Options		
Pooling type	None, Max Pooling, Average Pooling		
Activations	ReLU, Sigmoid, Tanh, SELU, ELU, Exponential		
Optimization algorithm	ADAM	SGD	RMSprop
Learning Rate	$1,\ 10^{-1},\ 10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}$		
Activation exit neuron	Sigmoid or SoftMax		

4. Results and discussion

4.1. Assessing dataset formation with UMAP

Dimensionality reduction techniques, such as UMAP, in conjunction with clustering algorithms, like HDBSCAN [58,59], have been proven effective in accurately classifying composites in EELS spectra when addressing the thickness effect [18]. In this context, the term composites refer to distinct groups or clusters of spectra in the dataset, while classes correspond to the specific categories or labels assigned to these groups. The thickness effect arises when the intensity of a spectrum is proportional to the sample's thickness, leading intensity variations unrelated to the samples' chemical composition, which can lead to inaccurate results in ML algorithms. Other solution for this issue is the ML approach proposed by Dzyubachyk et al. [60] which addresses thickness-induced intensity variations in conventional TEM imaging, complementing dimensionality reduction techniques and clustering algorithms, specifically, UMAP and coherent local intensity clustering [61]. Finally, it is worth noting that thickness can also alter the ELNES of EELS spectra via plural scattering [62]. However, as we are working with small NPs of reduced sizes, this effect does not have a significant impact in the present study.

Given the advantages of UMAP as a dimensionality reduction technique for EELS, it was an appropriate method for evaluating the quality

of the constructed datasets. To assess the effectiveness of our datasets and examine the effect of normalization on EEL spectra, we employed the UMAP algorithm to reduce the high dimensionality of the datasets to 2D point for each spectrum, enabling their visualization in a 2D map. It is important to note that the parameters used in the UMAP algorithm for all the dimensional reductions correspond to a value of 0.5 for the minimum distance and 300 for the number of neighbors. This allowed us to observe whether the data was well-separated and originated from distinct composites. All datasets were reduced using the UMAP algorithm to determine if they were well-constructed in light of the intensity issue. The results were illustrated in Figs. 3 and 4, where each dataset was mapped.

In Fig. 3, it became apparent that, when analyzing the datasets without normalization (Fig. 3, 1st row), different classes in the datasets were corresponding to different composites, indicating well-constructed datasets. Even considering the thickness effect, which arose from the use of multiple samples with varying thicknesses to construct the datasets, it was evident that each composite was well-separated from the others. Conversely, when normalization was applied, a high degree of class overlap was observed for both the L2-norm and the maximum-norm, as seen in the 2nd and 3rd rows of Fig. 3, respectively. This suggest that normalization introduced a bias into our data and may impede classification and clustering algorithms to accurately classify or identify spectra.

From the results presented in Fig. 3 and drawing upon previous studies conducted by J. Blanco-Portals et al. [18,63], it is confirmed that UMAP has robust capabilities in dealing with the intensity effect without compromising its focus on ELNES. This dimensional reduction methodology considers intensity and ELNES as two interdependent variables and effectively manages to reduce dimensionality while preserving the importance of these characteristics. The results without normalization represented in the 1st row of Fig. 3 provide further support to this assertion, showing how UMAP even facilitates the correct separation of these two classes.

However, the application of normalization eliminates the variability associated with intensity. This phenomenon is clearly observed in Fig. 5, which displays the spectra pertaining to each class of the iron oxides (magnetite and wüstite), both normalized (using the L2 and max norm) and raw. In Fig. 5b, we observe that the intensity corresponding to wüstite is a bit higher than that of magnetite, complicating the observation of the iron white lines differences associated to each composite. On the other hand, in Fig. 5a, this intensity difference disappears in the spectra, where the displacement of the L3 onset becomes evident in combination of the L2 intensity difference. Therefore, the normalization improves visual distinction in the EELS edges for human visualization and some ML strategies. Yet, it does not necessarily enhance UMAP ability to distinguish between the spectra.

These observations lead us to realize that finding new strategies to analyze unnormalized spectra is crucial. While normalization might homogenize the spectra, thereby clarifying the EELS peak differences for human interpretation and a large quantity of ML strategies, it also confines all spectra to a strict intensity value range. This process may inadvertently discard valuable information that could assist in distinguishing spectra characteristics beyond thickness. Thus, a careful and

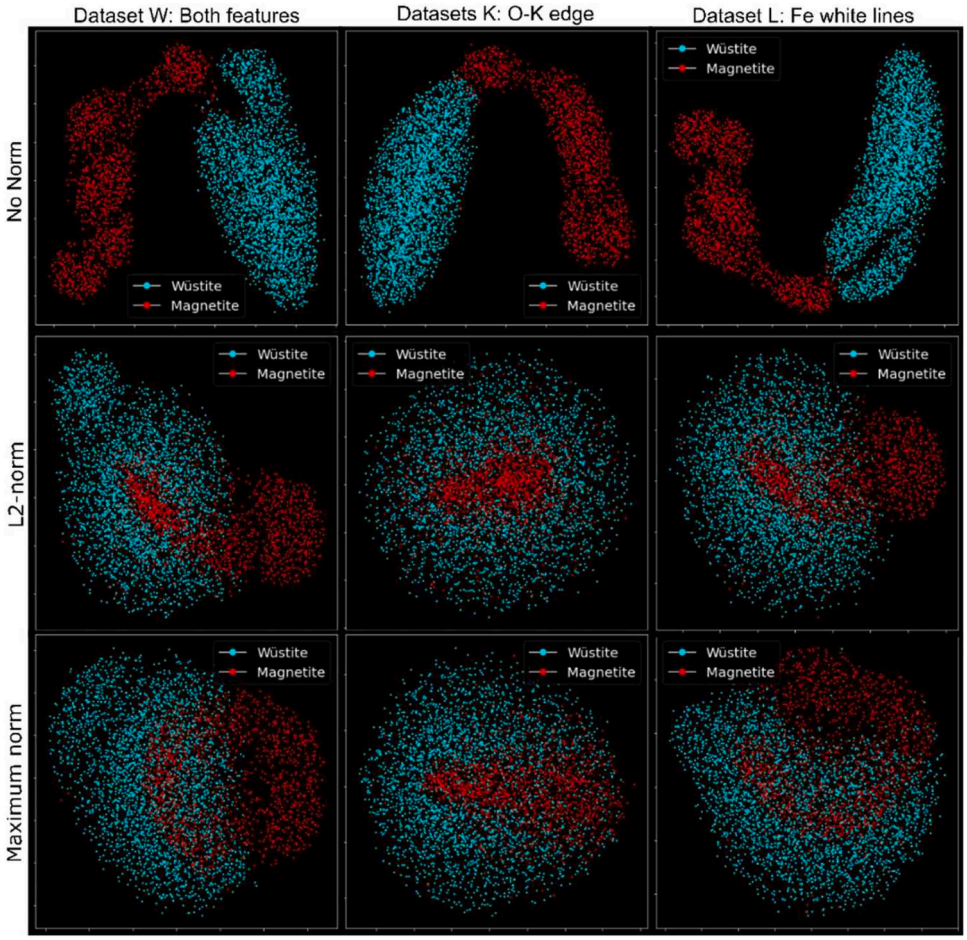


Fig. 3. 2D maps created using UMAP, a minimum distance between embedded points of 0.5 and a number of neighboring sample points of 300 are used for the nine datasets of Fe oxides used in this work. Each column represents a family of datasets (K, L and W) depending on the selected features (oxygen K edge, white lines or both), and each row indicates the normalization applied.

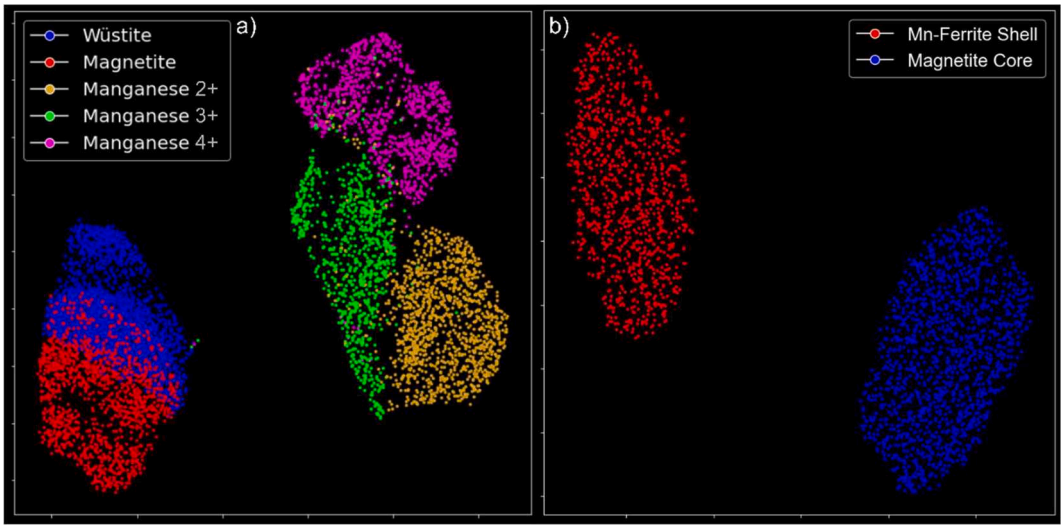


Fig. 4. 2D maps created using UMAP, a minimum distance between embedded points of 1.0 and a number of neighboring sample points of 300/150 are used for the a) WL dataset and b) MF dataset, respectively.

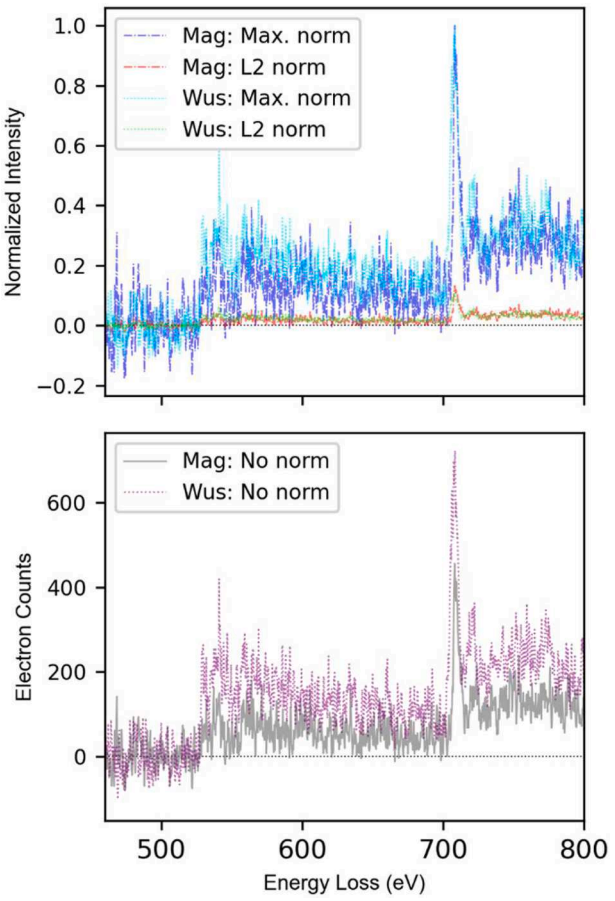


Fig. 5. EEL spectra for the Fe oxides classes, encompassing both the oxygen K-edge and the Fe white lines, and displaying the same set of spectra under different normalization conditions within the range of 460 to 800 eV: a) Normalized spectra (using both L2 and maximum norm), and b) Raw spectra (without normalization).

strategic approach to data normalization is essential in the field of EELS analysis.

Lastly, Fig. 4 depicts the UMAP embedding for WL dataset (Fig. 4a) and MF dataset (Fig. 4b). These maps unequivocally confirmed that both datasets were well-constructed, with all classes being well-separated. Comparing the 4a map to the one in the 3rd column and row of Fig. 3, corresponding to the L-max dataset, we observed that as more variability was added to the dataset, the UMAP algorithm was able to separate the Fe white lines more easily.

4.2. *Support vector machine classifiers*

The soft-margin SVM classifiers results for the 9 datasets built are presented in Table 3. The test accuracy outcomes of the SVM for classifying the oxidation state of transition metals using three EELS features—the oxygen K-edge (K), the white lines (L), and both features (W)—are shown as a function of the feature employed (column 1), the normalization approach (column 2), the kernel (column 3), and the optimal hyperparameters found (column 4). Three normalization strategies were employed: maximum counts, L2-norm, and no normalization. The latter, the absence of normalization, is associated with the introduction of a new kernel method, the cosine kernel, and is employed to simplify data preprocessing.

All classifier configurations in Table 3 achieved a minimum accuracy of 90%, making them effective models for the task at hand. However, some combinations were found to be more optimal. The highest accuracy was observed for the use of white lines (L) and the entire spectrum (W) that includes them, as opposed to using the oxygen K-edge (K) alone. This is a consequence of several factors. First, the ELNES of the oxygen K-edge contains more subtle features. Next, the white lines of the transition metal are less sensitive to the electron beam, providing more robust and accurate information about the crystal structure and bonding of the transition metal atoms in the sample, which is related to the oxidation state. Finally, the oxygen K-edge features have a lower signal-to-noise ratio compared to the white lines, making it more challenging to identify subtle differences related to oxidation states.

In this regard, it is important to clarify that the SVMs do not incorporate the information from the energy axis. Instead, the algorithm receives vectors of a fixed length, where the energy axis does not provide any energy information. Thus, the algorithm accurately classifies based on the shape of the edges.

As previously reported in our earlier work [35], the RBF kernel yielded the best results among the kernels used. In contrast, the sigmoid kernel exhibits the poorest performance, while other kernels achieve test accuracies up to 99% in some cases, the highest test accuracy achieved by the sigmoid kernel is 96% for the W-max L-l2 dataset. Notably, the cosine kernel demonstrates promising results and offers the advantage of

Table 3
Results of the test accuracy for different datasets using soft-margin SVM classifiers. The EELS feature column indicates the type of characteristic used, the normalization column indicates how the data was normalized, the kernel column indicates the kernel function considered, the C/Gamma/r column indicates the parameters used for each kernel, and the test accuracy column shows the accuracy of the classifier applied in the test set.

Dataset	EELS Feature	Normalization	Kernel	C / Gamma / r	Test Accuracy
W-max	Whole Spectrum	Maximum	Linear	0.1	0,97
			RBF	$10 / 10^{-2}$	0,97
			Sigmoid	$100 / 10^{-3} / 10^{-3}$	0,96
L-max	White Lines		Linear	0.1	0,97
			RBF	1 / 0.1	0,96
			Sigmoid	$1000 / 10^{-3} / 1$	0,97
K-max	Oxygen K-Edge		Linear	1	0,94
			RBF	10 / 0.1	0,96
			Sigmoid	$1000 / 10^{-3} / 0.1$	0,93
W-l2	Whole Spectrum	L2-Norm	Linear	0.1	0,95
			RBF	0.1 / 1.0	0,98
			Sigmoid	$100 / 10^{-3} / 10^{-3}$	0,96
L-l2	White Lines		Linear	1.0	0,96
			RBF	1 / 0.1	0,98
			Sigmoid	$100 / 10^{-3} / 10^{-3}$	0,90
K-l2	Oxygen K-Edge		Linear	100	0,93
			RBF	0.1 / 1	0,93
			Sigmoid	$100 / 10^{-3} / 10^{-3}$	0,90
W-nn	Whole Spectrum	No Normalization	Cosine	0.1	0,95
L-nn	White Lines		Cosine	1.0	0,99
K-nn	Oxygen K-Edge		Cosine	0.1	0,97

not requiring data normalization. As observed in Fig. 3, normalization has a substantial impact on EELS data, by skipping this step, the risk of altering the original data is reduced.

In Fig. 6, we evaluated the performance of the probabilistic soft-margin SVM classifier trained with the L-nn dataset and the cosine kernel on real EELS SIs of previously reported Fe oxide core-shell NCs [45–47]. The results indicate that the algorithm effectively distinguishes the magnetite-composed shell from the wüstite-dominated core. Furthermore, probabilistic classifiers such as this one does not depend on rigid labels, providing greater flexibility in the classification process. It is also important to mention that the only data preprocessing applied to these SIs consisted of removing the pre-edge background and cropping the energy region corresponding to the white lines. This underscores the simplicity and efficacy of the cosine kernel method for identifying EELS spectra.

In this sense, in Fig. 7, we reevaluate the cosine kernel to address a different problem involving the determination of core and shell, but this time not through the oxidation state of iron as in the case of Fig. 6, but through the relative intensities of the Mn and Fe white lines that are related to the composition of each transition metal. In this case, the spectra are classified by the intensity of each white lines, this allows us to assess the proper functioning of the cosine kernel more confidently. If we look at Fig. 7, it is worth noting that the spectra used to train the SVM classifier, i.e., to build the MF dataset, were obtained from SIs a) and b), with an energy dispersion of 0.1 eV/ch, equal to SI e), while SIs c) and d) were obtained at a dispersion of 0.5 eV/ch. In the case of these higher energy resolution images, the SVM model is trained by the same MF dataset as for the rest of the classifications, with the difference that their energy resolution was adapted by interpolation. Furthermore, the cosine kernel successfully overcomes the thickness effect. This is evident in

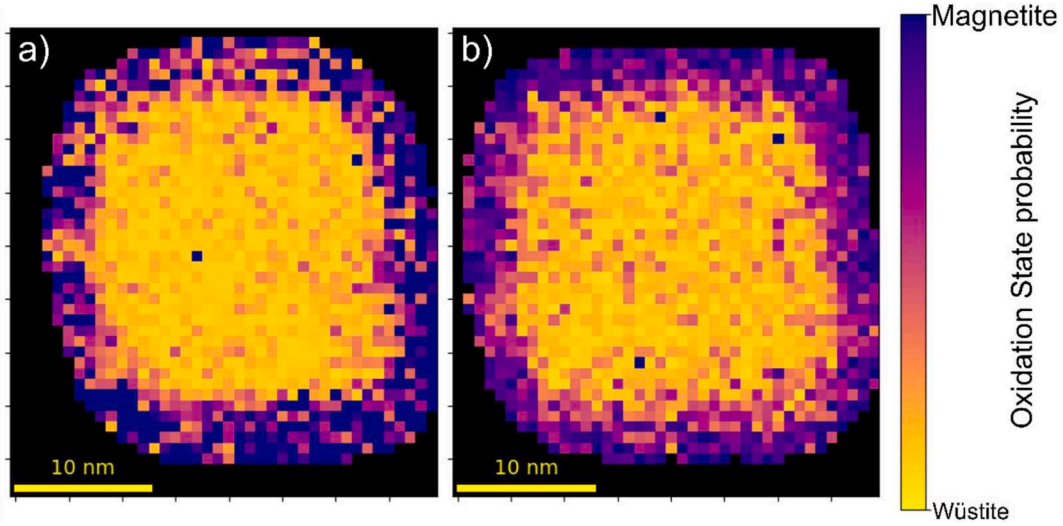


Fig. 6. Classification map of EELS SIs for two distinct core/shell NCs, obtained using a probabilistic cosine kernel SVM. The colors represent the probability of the oxidation states of magnetite and wüstite.

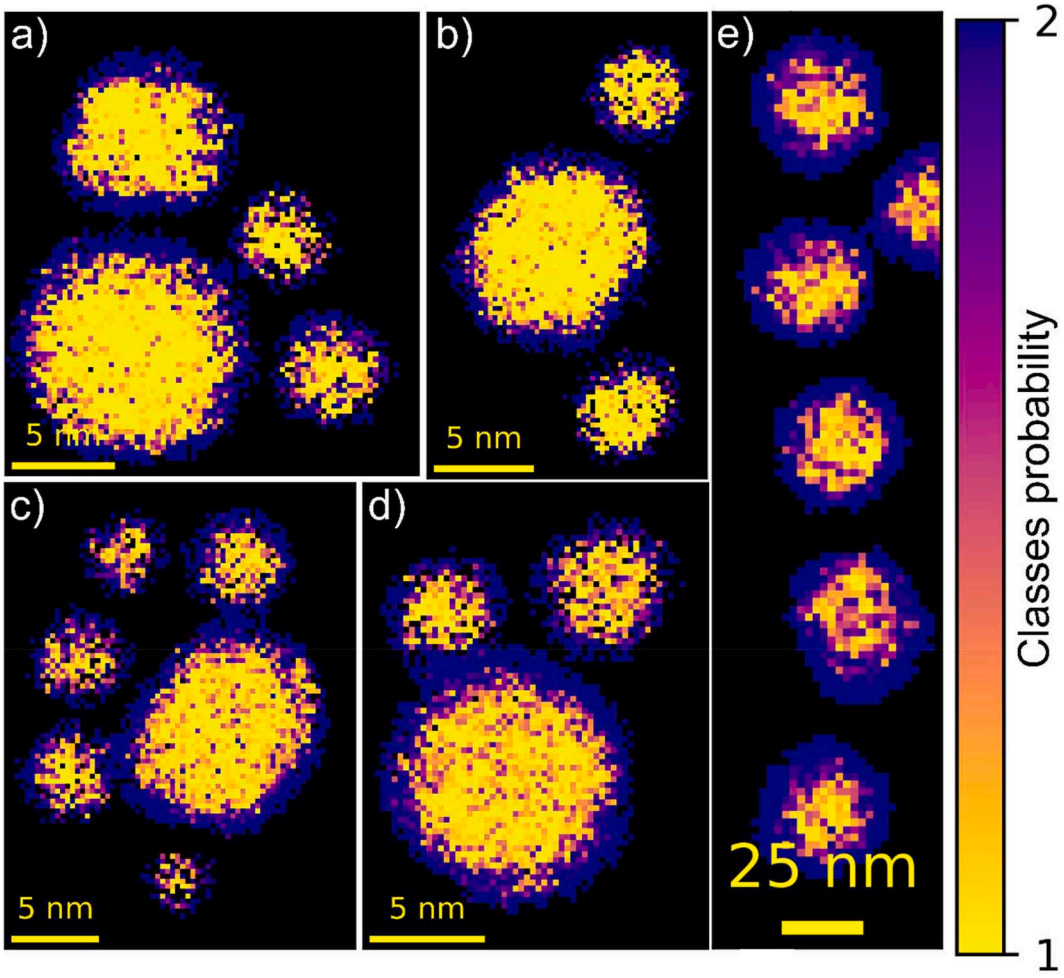


Fig. 7. Classification map of EELS SIs for five distinct Fe/Mn oxide core/shell NPs, generated using a probabilistic cosine kernel SVM. The colors represent the probability of a spectrum belonging to a shell (class 2) or a core (class 1).

Fig. 7, where the diameter of the NPs ranges from 5 to 56 nm, while the MF dataset was constructed from NPs with diameters ranging from 5 to 11 nm.

When examining the resulting probabilistic classification of all the SIs, we clearly see the good performance offered by this kernel without normalizing spectra, thereby reducing the bias that might be incorporated into them. Moreover, since these SIs have different energy resolutions, this demonstrates that if we have high-energy-resolution training spectra for the problem we want to tackle, any spectrum acquired at a higher energy resolution can be classified without major inconvenience. Therefore, Fig. 7 shows that SVMs can be used as EELS spectrum classifiers to solve a wide variety of problems while also demonstrating that using the cosine kernel makes unnecessary the normalization of our spectra.

Regarding this ability of the cosine kernel and SVMs to identify EELS spectra without the need for normalizing the spectra, it is crucial to note that this behavior is not coincidental or arbitrary. When examining the formulation of the kernel in Eq. (2), we observe that it measures the cosine of the angle between two vectors in an n-dimensional space, disregarding the magnitude of these vectors, i.e., the intensity. Thus, by definition, the cosine kernel inherently incorporates an internal normalization when computing the similarity between two vectors. Hence, we decided not to assess normalized data using the cosine kernel. It appeared redundant since this kernel already incorporates internal normalization. Furthermore, ML algorithms are significantly impacted by data intensity variability [64]. Thus, testing non-normalized data with these algorithms seemed impractical.

4.3. Artificial neural network classifiers

4.3.1. ANN optimization

The results of the ANN hyperparameters optimization, using the RS and TPE optimization methods are presented and discussed in this section. We start evaluating the results from the RS, then we proceed to analyze the results from the TPE optimization and finally we evaluate the performance of the best architecture as an ANN classifier for determining the oxidation state.

Regarding the RS results, due to the large number of hyperparameters and the computational cost of model selection, a total of approximately 475 trials were performed for each type of structure, dense and convolutional. The exact number of models searched was summarized in Table 4. During the RS, models were trained for 10 and 30 epochs. Given that we were aiming to identify the best model for EEL spectra, we used relatively small values for the number of training epochs. The criterion used to determine the success of a classification model is a test validation accuracy greater than 0.90, that is, 90%.

In order to determine the most effective ANN classifier for EELS spectra, we present Table 5. This table displays the success rate for different structures as a function of the number of epochs used during training. To account for the non-uniform distribution of sample sizes resulting from this RS, we have employed a bayesian approach, specifically a beta-binomial model, to estimate the success rate for each structure. This approach allowed for a reliable and informative determination of the most effective classifier based on the results obtained.

This bayesian model was constructed with the number of successful

Table 4
Results of a RS for ANN models using two different structures (dense and convolutional) over two different sets of epochs (10 and 30). The table displays the total number of models attempted, the number of successful models (with accuracy greater than or equal to 0.9), and the number of failed models (with accuracy less than 0.9).

Random search		Total	Successful model (Accuracy $\geq$ 0.9)	Failed model (Accuracy $<$ 0.9)
10 epochs	Dense	175	77	98
	Conv	162	101	61
30 epochs	Dense	300	180	120
	Conv	306	206	100

Table 5
Success ratio of the models per structure. The table includes the results for dense and convolutional structures. The first and last columns indicate the structure's number and the central columns the success ratio for each structure and epochs used in the training. Then, the rows show the different structures, the ones marked in purple correspond with most successful structures.

	Success ratio per structure (%)				
Dense Structures	10 epochs	30 epochs	10 epochs	30 epochs	Conv. Structures
0	47	54	48	45	0
1	42	65	57	74	1
2	50	74	48	64	2
3	55	78	52	62	3
4	61	72	44	49	4
5	42	51	69	85	**5**
6	54	59	75	78	**6**
7	29	40	88	80	**7**
8	33	62			
9	19	38			
10	50	65			
11	36	30			
12	67	78			
13	50	63			

attempts and total attempts for each structure considered. The prior distribution for the success rate of each structure was a beta distribution, which reflects our uncertainty regarding the true success rate. Using a binomial distribution, we modeled for each structure the probability of observing a certain number of successful attempts given the total number of attempts and the success rate. Using bayesian inference and the beta-binomial model, we computed the success ratio per structure.

From the results in Table 5, we found that the best performing dense structures were numbers 3, 4, and 12, while the top convolutional structures were numbers 5, 6, and 7. Dense structures 3 and 4 combine dense and normalization layers, while structure 12 have a two-branch architecture. Convolutional structures 5 and 6 are two-branch configurations with normalization layers, and structure 7 is based on the ResNet architecture.

After identifying the most suitable structures for EELS classification, the remaining hyperparameters for these structures were evaluated using Fig. 8. As shown in this figure, optimal models were found across the entire range of each parameter, indicating that for the majority of the investigated hyperparameters, there were no clear preferences. The only metric that demonstrated a distinct preference was the learning rate. For dense models, small rates ($\leq 10^{-3}$) were preferred, while rates between 10^{-2} and 10^{-4} were optimal for convolutional models. Although no specific parameter configuration was observed, Tables 6 and 7 list the parameters for the models with the highest absolute accuracy.

After discussing the RS results, we examine the results obtained through TPE optimization. The TPE algorithm was applied to evaluate a total of 250 and 500 models for both type of structures, each trained for 30 epochs. The TPE optimization process resulted in a selection of the 10th structure for dense layers and 5th structure for convolutional layers. As for the RS, we have presented the hyperparameters of the best models in Tables 6 and 7. Although for the convolutional layer we have shown that the 7th structure is also convenient, their complexity is higher than the 5th one, thus, we decided to pick this first one.

From these tables, it is observed that in most of the dense and convolutional structures, the preferred optimization algorithm is SGD or RMSprop. It is also observed that the number of dense layers and the number of filters in the convolution layers vary significantly between the optimal structures. However, in general, no clear pattern is observed in terms of the number of convolution layers or kernel size. Based on these results, it appears that the tanh activation function is a preferred choice for the dense structures, as it is used in several of the models with the highest validation accuracy. Additionally, for the convolutional structures, ReLU, SELU and tanh activation functions are used in models with high validation accuracy. It is noteworthy that for 10 epochs, the simplest architecture achieved the highest accuracy due to having fewer parameters to optimize. However, as the number of epochs increased, the accuracy of larger networks improved.

Based on these findings, we conclude that determining optimal parameters for an ANN classifier is a challenging task, as the weights are constantly updated during training to determine the optimal weight configuration for the selected hyperparameters [65]. Furthermore, we have observed that all the successful architectures contain normalization layers, thus, it is recommended to include them in ANN classifiers built for EELS data [66]. With regards to the optimization problem, we are attempting to solve, our data complexity is low. Therefore, with a sufficient number of training iterations, a wide range of configurations can be properly optimized and thus, can be effective classifiers. This highlights that this type of parameter optimization may not be necessary for data with low complexity [67]. In addition, it is important to note that the complexity of a model should be matched to the complexity of the data [67].

4.3.2. *ANN classification*

After discussing the most appropriate ANN architecture for EELS data from both optimization strategies, we have selected one configuration each from Tables 6 and 7 and trained the ANN models using the datasets constructed to evaluate their performance in a realistic scenario. For the dense structures, we chose configuration 10, while for the convolutional case, we selected configuration 5. In this regard, the convolutional classifiers comprised approximately 5000 parameters, and the dense ones contained around 3 million parameters. The test accuracy results obtained are presented in Table 8.

From Table 8, we derived several insights regarding the classification accuracy of different datasets. The test accuracies were consistently high 99% for the whole spectrum (W) and white lines (L) features across all normalization methods. This indicated that both dense and convolutional classifiers excel in handling these features, irrespective of the normalization technique employed. In contrast, the oxygen K-edge (K) feature yielded lower test accuracies compared to the whole spectrum and white lines, with convolutional classifiers consistently outperforming their dense counterparts. This suggests that convolutional structures may be better suited for these particular data.

However, the difference in the number of parameters between the

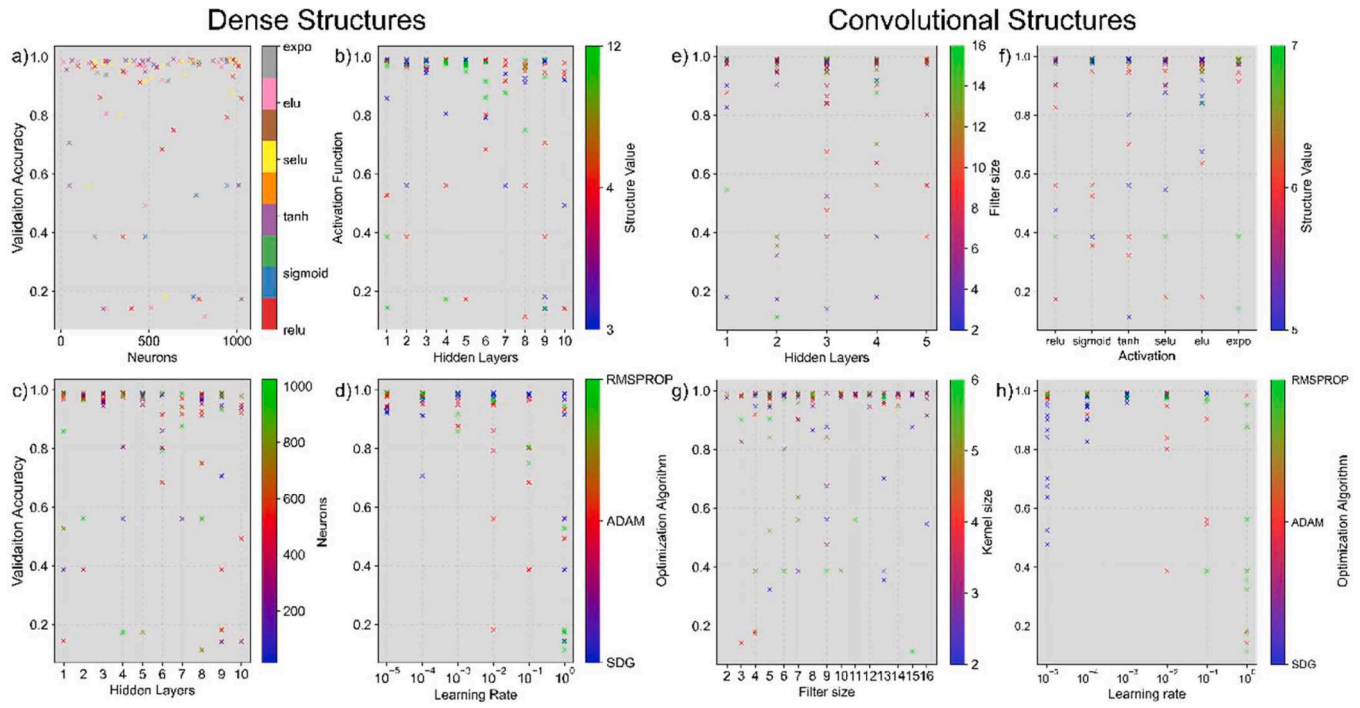


Fig. 8. Representation of hyperparameter for RS based on validation accuracy of the most successful ANN structures found in Table 5. The sub-figures (a)–(d) depict the parameters for dense structures, while sub-figures (e)–(h) show the parameters for convolutional structures. Sub-figure (a) illustrates the activation functions and number of neurons, sub-figure b) displays the number of hidden layers and structure value, sub-figure c) showcases the number of hidden layers and neurons, sub-figures (d) and (h) depict the learning rate and optimization algorithm, sub-figure e) illustrates the number of hidden layers and filter size, sub-figure (f) displays the activation functions and structure values, and sub-figure g) shows the filter and kernel size.

Table 6
Hyperparameters for the most successful dense ANN structures obtained from the RS and the TPE optimization.

Hyperparameters for dense structures				
	RS		PTE	
	10 Epochs	30 Epochs	30 Epochs	30 Epochs
Tries	175	300	250	500
Validation Accuracy	0.993	0.992	0.995	0.995
Dense layers	1	9	5	9
Neurons	640	944	890	725
Structure	3	12	10	10
Activation	Tanh	Tanh	Tanh	SELU
Algorithm	RMSprop	RMSprop	SGD	SGD
Learning Rate	10^{-5}	10^{-5}	10^{-1}	10^{-1}
Activation exit neuron	SoftMax	SoftMax	Sigmoid	Sigmoid

Table 7
Hyperparameters for the most successful convolutional ANN structures obtained from the RS and the TPE optimization.

Hyperparameters for convolutional structures				
	RS		PTE	
	10 Epochs	30 Epochs	30 Epochs	20 Epochs
Tries	162	306	250	500
Validation Accuracy	0.993	0.993	0.995	0.995
Structure	5	5	5	7
Convolution layers	1	3	2	2
Convolution filters	15	16	11	6
Convolution kernel size	5	3	4	5
Pooling size	–	–	2	3
Dense layers	1	8	1	9
Neurons	672	720	648	448
Pooling type	None	None	Average	Average
Activations	ReLU	Tanh	ELU	SELU
Optimization algorithm	ADAM	RMSprop	SGD	SGD
Learning Rate	0.001	0.0001	0.01	0.01
Activation exit neuron	Sigmoid	SoftMax	Sigmoid	Sigmoid

CNN and DNN models might impact their performance. The CNN model, with fewer parameters, leaded to a more simplified model that generalizes better and potentially results in improved performance on test data. Since both models were trained for the same number of epochs, the more complex DNN model, with its higher number of parameters, might not have enough time to converge to an optimal solution. Consequently, the simpler CNN model exhibited better performance in this scenario. It was also crucial to consider that CNNs were inherently designed to exploit local correlations and spatial hierarchies in the data, which can be advantageous when working with spectral data like EELS. This inherent structure in CNNs contributed to their superior performance compared to DNNs.

4.4. Energy shifts

In this section, we present the results of resistance of SVM and ANN models to energy shifts in combination with the results when they were trained with energy-shifted spectra. In Fig. 9, we present three plots displaying test accuracy as a function of the energy shift values applied in the spectra for three classifiers: (a) linear SVM, (b) RBF SVM, and (c) convolutional ANN classifier. Each classifier is represented in a separate plot, displaying two curves: one representing the model trained with spectra without shift applied (WL dataset) in blue, and a second curve in orange representing the model trained with randomly shifted spectra up to a maximum of 1.75 eV for both sides. For each curve, two vertical lines indicate the corresponding energy shift values at which the accuracy drops below 90% of its highest value (when no shift is applied) for each side, these values are also numerically provided in the legend.

The first clear result emerged when examining the blue curves, where we observed that the classifier offering the best resilience to

Table 8
Results of the classification accuracy for the different datasets using the ANN classifiers. The table shows the test accuracy for different EELS features and normalization methods for both dense and convolutional ANN structures.

Dataset	EELS feature	Normalization	Test Accuracy	
			Dense	Conv.
W-max	Whole Spectrum	Maximum	0,99	0,99
L-max	White Lines		0,99	0,99
K-max	Oxygen K-Edge		0,95	0,97
W-l2	Whole Spectrum	L2-Norm	0,99	0,99
L-l2	White Lines		0,99	0,99
K-l2	Oxygen K-Edge		0,93	0,95
W-nn	Whole Spectrum	No Normalization	0,99	0,99
L-nn	White Lines		0,96	0,99
K-nn	Oxygen K-Edge		0,92	0,98

energy shifts was the SVM using the RBF kernel, closely followed by the convolutional classifier. Subsequently, when evaluating the effect of training the models with shifted spectra, we found that all models indeed improved their resilience, with the initially most resilient models showing the most significant improvement. Specifically, the SVM using the RBF kernel and the ANN classifier exhibit the greatest improvement in resilience compared to the initial model, while, in the linear SVM classifier the improvement was relatively small. In addition, examining all the models, we noticed that their robustness was not symmetric with respect to the side of the energy shift applied (negative or positive shifts). We observed that the slope of the accuracy curves was less pronounced for positive energy shifts than for negative ones, suggesting better resistance to positive shifts. This observation was also supported by the values of the accuracy drops indicated in the legends of Fig. 9.

4.5. Discussion

Based on the results obtained from the SVM and ANN classifiers, it was evident that white lines serve as the most reliable features for determining the oxidation state. Convolutional classifiers have been demonstrated to outperform dense classifiers, which was consistent with their lower complexity and therefore faster training times. This performance advantage can largely be attributed to the convolutional layers' ability to account for the spatial structure of the input data through the application of convolutional filters that scan the input spectrum to extract characteristics. Furthermore, the utilization of convolutional layers enables the network to learn spatial hierarchies of characteristics, where lower layers discern simple and local elements and higher layers identify more complex and abstract factors by combining lower-level features, thereby enhancing the network's capacity to generalize to

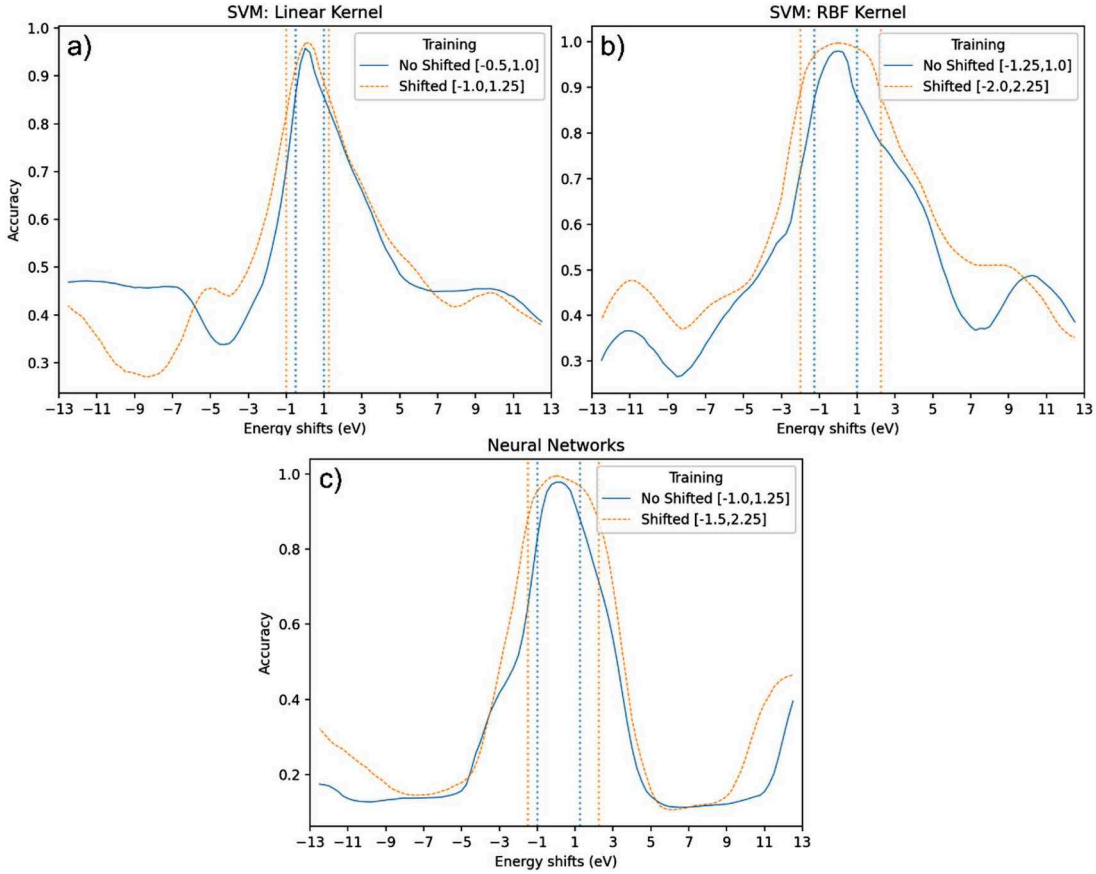


Fig. 9. Test accuracy as a function of energy shift values (in eV) for (a) linear SVM, (b) RBF SVM, and (c) convolutional ANN classifier. Blue and orange curves represent models trained on unshifted and shifted spectra, respectively. Vertical lines indicate energy shift values at which accuracy drops below 90%, with numerical values provided in the legend.

novel data.

Regarding the SVM classifiers, they exhibited comparable performance to ANN classifiers and inherently possess greater resistance to energy shifts, which can be further improved by training the model with shifted spectra. The SVM employing the RBF kernel emerged as the optimal classifier for handling energy shifts, while employing the cosine kernel proved that the normalization could be unnecessary for classifying EEL spectra Additionally, SVM models generally have faster training times than those based on ANN, as they typically require fewer parameters to be adjusted. These findings lead us to conclude that employing ANN as a classifier for EEL spectra, contingent upon the feature under investigation or the volume of available data, may be an impractical, overly complex, and resource-intensive methodology.

Consequently, for the classification of specific and individual features of EELS spectra, where data complexity is low, we strongly recommend for the use of SVMs. However, this does not imply that there are no scenarios in which ANN is a significantly more practical tool, if the aim is to train a classification model of transition metal oxidation states based on their white lines for a large number of compounds, the data complexity in this case would be sufficiently high to guarantee the use of ANN classifiers as a more convenient option.

5. Conclusions

Through the comparison carried out in this work, we conclude that both soft-margin SVM and ANN classifiers are highly effective for EELS spectra classification, specifically for discerning the oxidation state of transition metals. We have observed that SVMs perform better for classifying EELS spectra when using the RBF kernel, and that convolutional networks provide the best performance. Notably, the cosine kernel has shown excellent performance in classifying EELS spectra without the need for normalization.

The most effective feature for classifying oxidation states in transition metals was their white lines, although the oxygen K-edge also provided good results. Since the oxygen K-edge has more complex ELNES compared to white lines, it presented more challenges for both classifiers studied. In addition, the SVMs offered greater robustness when dealing with energy shifts, and in both cases, for ANNs and SVMs, training the models with shifted spectra further improved their resistance to them, providing accurate classifications for shifts up to 2.5 eV.

Thanks to the UMAP algorithm, we have observed the effects of normalization on dimensionality reduction across datasets constructed in this study. In this regard, we noticed that normalization significantly alters the data structure, which is often the price that have to be paid for enabling algorithms to effectively analyze data. In this context, we propose an alternative to the normalization procedure by utilizing SVMs with the cosine kernel. This approach demonstrated excellent performance in classifying EELS spectra without the need for normalization.

In conclusion, we recommend the use of SVMs over ANNs for simple EELS classification problems, where there is a small number of spectra to train the models, as these classifiers can offer the same performance as ANNs at a lower computational cost and shorter training and optimization times. On the other hand, the use of ANNs is recommended when dealing with large volumes of training data and, therefore, facing more complex classification problems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by the Spanish Project PDC2021-121366-I00 financed by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. The authors also acknowledge funding from MICIIN under the project PID2019-106165GB-C21, the support received from the ELECMI - ICTS Electron Microscopy for Materials Science and the funding from Generalitat de Catalunya under project 2021SGR00242 and the 2020 FI-SDUR 00035 grant from the AGAUR agency of the Generalitat de Catalunya. SuperSTEM is the UK National Research Facility for Advanced Electron Microscopy, funded by the Engineering and Physical Sciences Research Council (EP/W021080/1). The authors would like to thank Prof Vlado Lazarov (University of York) and Prof Sara A. Majetich (Carnegie Mellon University) for the use of data from Mn-ferrite NP samples.

References

[1] S. Raschka, J. Patterson, C. Nolet, Machine learning in python: main developments and technology trends in data science, machine learning, and artificial intelligence, Information 11 (4) (2020).
[2] S.V. Kalinin, C. Ophus, P.M. Voyles, R. Erni, D. Kepaptsoglou, V. Grillo, A. R. Lupini, M.P. Oxley, E. Schwenker, M.K.Y. Chan, J. Etheridge, X. Li, G.G.D. Han, M. Ziatdinov, N. Shibata, S.J. Pennycook, Machine learning in scanning transmission electron microscopy, Nat. Rev. Methods Primers 2 (1) (2022) 11, https://doi.org/10.1038/s43586-022-00095-w.
[3] N. Bonnet, N. Brun, C. Colliex, Extracting information from sequences of spatially resolved EELS spectra using multivariate statistical, analysis 77 (1999), https://doi.org/10.1016/S0304-3991(99)00042-X.
[4] N. Bonnet, D. Nuzillard, Independent component analysis: a new possibility for analysing series of electron energy loss spectra, Ultramicroscopy 102 (4) (2005) 327–337, https://doi.org/10.1016/j.ultramic.2004.11.003.
[5] X. Hu, Y. Sun, J. Yuan, Multivariate statistical analysis of electron energy-loss spectroscopy in anisotropic materials, Ultramicroscopy 108 (5) (2008) 465–471, https://doi.org/10.1016/j.ultramic.2007.07.005.
[6] N. Dobigeon, N. Brun, Spectral mixture analysis of EELS spectrum-images, Ultramicroscopy 120 (2012) 25–34, https://doi.org/10.1016/j.ultramic.2012.05.006.
[7] F. de la Peña, M.H. Berger, J.F. Hochepied, F. Dynys, O. Stephan, M. Walls, Mapping titanium and tin oxide phases using EELS: an application of independent component analysis, Ultramicroscopy 111 (2) (2011) 169–176, https://doi.org/10.1016/j.ultramic.2010.10.001.
[8] L. Yedra, A. Eljarrat, R. Arenal, E. Pellicer, M. Cabo, A. López-Ortega, M. Estrader, J. Sort, M.D. Baró, S. Estradé, F. Peiró, EEL spectroscopic tomography: towards a new dimension in nanomaterials analysis, Ultramicroscopy 122 (2012) 12–18, https://doi.org/10.1016/j.ultramic.2012.07.020.
[9] L. Yedra, A. Eljarrat, R. Arenal, L. López-Conesa, E. Pellicer, A. López-Ortega, M. Estrader, J. Sort, M.D. Baró, S. Estradé, F. Peiró, Electron energy-loss spectroscopic tomography of Fe_xCo(3−x) O $_4$ impregnated Co $_3$ O $_4$ mesoporous particles: unraveling the chemical information in three dimensions, Analyst 141 (16) (2016) 4968–4972, https://doi.org/10.1039/C6AN00562D.
[10] S. Lichtert, J. Verbeeck, Statistical consequences of applying a PCA noise filter on EELS spectrum images, Ultramicroscopy 125 (2013) 35–42, https://doi.org/10.1016/j.ultramic.2012.10.001.
[11] A. Eljarrat, L. López-Conesa, J. López-Vidrier, S. Hernández, B. Garrido, C. Magén, F. Peiró, S. Estradé, Retrieving the electronic properties of silicon nanocrystals embedded in a dielectric matrix by low-loss EELS, Nanoscale 6 (24) (2014) 14971–14983, https://doi.org/10.1039/C4NR03691C.
[12] B.H. Martineau, D.N. Johnstone, A.T.J. van Helvoort, P.A. Midgley, A.S. Eggeman, Unsupervised machine learning applied to scanning precession electron diffraction data, Adv. Struct. Chem. Imaging 5 (1) (2019), https://doi.org/10.1186/s40679-019-0063-3.
[13] H.W. Ånes, I.M. Andersen, A.T.J. van Helvoort, Crystal phase mapping by scanning precession electron diffraction and machine learning decomposition, Microsc. Microanal. 24 (S1) (2018) 586–587, https://doi.org/10.1017/s1431927618003422.
[14] M. Pelaez-Fernandez, B. Majérus, D. Funes-Hernando, R. Dufour, J.L. Duvail, L. Henrard, R. Arenal, Toward laser-induced tuning of plasmonic response in high aspect ratio gold nanostructures, Nanophotonics 11 (16) (2022) 3719–3728, https://doi.org/10.1515/nanoph-2022-0193.
[15] J.K. Sunde, C.D. Marioara, A.T.J. van Helvoort, R. Holmestad, The evolution of precipitate crystal structures in an Al-Mg-Si(-Cu) alloy studied by a combined HAADF-STEM and SPED approach, Mater. Charact. 142 (2018) 458–469, https://doi.org/10.1016/j.matchar.2018.05.031.
[16] T. Blum, J. Graves, M. Zachman, R. Kannan, X. Pan, M. Chi, Machine learning for challenging EELS and EDS spectral decomposition, Microsc. Microanal. 25 (S2) (2019) 180–181, https://doi.org/10.1017/s1431927619001636.
[17] X. Li, O.E. Dyck, M.P. Oxley, A.R. Lupini, L. McInnes, J. Healy, S. Jesse, S. V. Kalinin, Manifold learning of four-dimensional scanning transmission electron microscopy, NPJ Comput. Mater. 5 (1) (2019), https://doi.org/10.1038/s41524-018-0139-y.

[18] J. Blanco-Portals, F. Peiró, S. Estradé, Strategies for EELS data analysis. Introducing UMAP and HDBSCAN for dimensionality reduction and clustering, Microsc. Microanal. 28 (1) (2022) 109–122, https://doi.org/10.1017/S1431927621013696.

[19] P. Torruella, M. Estrader, A. López-Ortega, M.D. Baró, M. Varela, F. Peiró, S. Estradé, Clustering analysis strategies for electron energy loss spectroscopy (EELS), Ultramicroscopy 185 (2018) 42–48, https://doi.org/10.1016/j.ultramic.2017.11.010.

[20] C.M. Pate, J.L. Hart, M.L. Taheri, RapidEELS: machine learning for denoising and classification in rapid acquisition electron energy loss spectroscopy, Sci. Rep. 11 (1) (2021) 19515, https://doi.org/10.1038/s41598-021-97668-8.

[21] M.P. Oxley, M. Ziatdinov, O. Dyck, A.R. Lupini, R. Vasudevan, S.V. Kalinin, Probing atomic-scale symmetry breaking by rotationally invariant machine learning of multidimensional electron scattering, NPJ Comput. Mater. 7 (1) (2021) 65, https://doi.org/10.1038/s41524-021-00527-3.

[22] K.M. Roccapriore, O. Dyck, M.P. Oxley, M. Ziatdinov, S.V. Kalinin, Automated experiment in 4D-STEM: exploring emergent physics and structural behaviors, ACS Nano (2021), https://doi.org/10.48550/arXiv.2112.04479.

[23] C. Wang, Q. Luo, E. Holm, Hierarchically structured classification of carbon nanostructures from TEM images by machine learning and computer vision, Microsc. Microanal. 27 (S1) (2021) 448–449, https://doi.org/10.1017/s1431927621002105.

[24] Z.H. Wu, J.J. Bai, D.D. Zhang, G. Huang, T.B. Zhu, X.J. Chang, R.D. Liu, J. Lin, J. A. Sun, Statistical analysis of helium bubbles in transmission electron microscopy images based on machine learning method, Nuclear Sci. Techn. 32 (5) (2021) 54, https://doi.org/10.1007/s41365-021-00886-y.

[25] S. Lu, B. Montz, T. Emrick, A. Jayaraman, Semi-supervised machine learning workflow for analysis of nanowire morphologies from transmission electron microscopy images, Digit. Discov. 1 (6) (2022) 816–833, https://doi.org/10.1039/D2DD00066K.

[26] H. Wen, J.M. Luna-Romera, J.C. Riquelme, C. Dwyer, S.L.Y. Chang, Statistically representative metrology of nanoparticles via unsupervised machine learning of TEM images, Nanomaterials 11 (10) (2021) 2706, https://doi.org/10.3390/nano11102706.

[27] H. Wen, X. Xu, S. Cheong, S.C. Lo, J.H. Chen, S.L.Y. Chang, C. Dwyer, Metrology of convex-shaped nanoparticles via soft classification machine learning of TEM images, Nanoscale Adv. 3 (24) (2021) 6956–6964, https://doi.org/10.1039/D1NA00524C.

[28] S. Akers, E. Kautz, A. Trevino-Gavito, M. Olszta, B.E. Matthews, L. Wang, Y. Du, S. R. Spurgeon, Rapid and flexible segmentation of electron microscopy data using few-shot machine learning, NPJ Comput. Mater. 7 (1) (2021) 187, https://doi.org/10.1038/s41524-021-00652-z.

[29] R. Lin, R. Zhang, C. Wang, X.Q. Yang, H.L. Xin, TEMImageNet training library and atomsegnet deep-learning models for high-precision atom segmentation, localization, denoising, and deblurring of atomic-resolution images, Sci. Rep. 11 (1) (2021) 5386, https://doi.org/10.1038/s41598-021-84499-w.

[30] P. Cho, A. Wood, K. Mahalingam, K. Eyink, Defect detection in atomic resolution transmission electron microscopy images using machine learning, Mathematics 9 (11) (2021) 1209, https://doi.org/10.3390/math9111209.

[31] A. Leitherer, A. Ziletti, L.M. Ghiringhelli, Robust recognition and exploratory analysis of crystal structures via bayesian deep learning, Nat. Commun. 12 (1) (2021) 6234, https://doi.org/10.1038/s41467-021-26511-5.

[32] M. Ziatdinov, C. Nelson, R.K. Vasudevan, D.Y. Chen, S.V. Kalinin, Building ferroelectric from the bottom up: the machine learning analysis of the atomic-scale ferroelectric distortions, Appl Phys Lett 115 (5) (2019), 052902, https://doi.org/10.1063/1.5109520.

[33] M. Shen, G. Li, D. Wu, Y. Yaguchi, J.C. Haley, K.G. Field, D. Morgan, A deep learning based automatic defect analysis framework for *in-situ* TEM ion irradiations, Comput. Mater. Sci. 197 (2021) 110560, https://doi.org/10.1016/j.commatsci.2021.110560.

[34] T. Friedrich, C.P. Yu, J. Verbeeck, S. Van Aert, Phase object reconstruction for 4D-STEM using deep learning (2022), https://doi.org/10.1093/micmic/ozac002.

[35] D. del-Pozo-Bueno, F. Peiró, S. Estradé, Support vector machine for EELS oxidation state determination, Ultramicroscopy 221 (2021), 113190, https://doi.org/10.1016/j.ultramic.2020.113190.

[36] M. Chatzidakis, G.A. Botton, Towards calibration-invariant spectroscopy using deep learning, Sci. Rep. 9 (1) (2019) 2126, https://doi.org/10.1038/s41598-019-38482-1.

[37] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, https://doi.org/10.1007/BF00994018.

[38] Y. Ma, G. Guo, Support vector machines applications, Ma, Y., Guo, G.. Engineering, Engineering (R0), 1st ed., Springer International Publishing, Cham, 2014 https://doi.org/10.1007/978-3-319-02300-7.

[39] J.C. Platt, Sequential minimal optimization: a fast algorithm for training support vector machines. https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/.

[40] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828, https://doi.org/10.1109/TPAMI.2013.50.

[41] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.

[42] L. McInnes, J. Healy, N. Saul, L. Großberger, UMAP: uniform manifold approximation and projection, J. Open Source Softw. 3 (29) (2018) 861, https://doi.org/10.21105/joss.00861.

[43] C. Colliex, T. Manoubi, C. Ortiz, Electron-energy-loss-spectroscopy near-edge fine structures in the iron-oxygen system, Phys. Rev. B 44 (20) (1991) 11402–11411, https://doi.org/10.1103/PhysRevB.44.11402.

[44] H. Tan, J. Verbeeck, A. Abakumov, G. Van Tendeloo, Oxidation state and chemical shift investigation in transition metal oxides by EELS, Ultramicroscopy 116 (2012) (2012) 24–33, https://doi.org/10.1016/j.ultramic.2012.03.002.

[45] P. Torruella, R. Arenal, F. de la Peña, Z. Saghi, L. Yedra, A. Eljarrat, L. López-Conesa, M. Estrader, A. López-Ortega, G. Salazar-Alvarez, J. Nogués, C. Ducati, P. A. Midgley, F. Peiró, S. Estradé, 3D visualization of the iron oxidation state in FeO/Fe_3O_4 core–shell nanocubes from electron energy loss tomography, Nano Lett. 16 (8) (2016) 5068–5073, https://doi.org/10.1021/acs.nanolett.6b01922.

[46] D. del-Pozo-Bueno, M. Varela, M. Estrader, A. López-Ortega, A.G. Roca, J. Nogués, F. Peiró, S. Estradé, Direct evidence of a graded magnetic interface in bimagnetic core/shell nanoparticles using electron magnetic circular dichroism (EMCD), Nano Lett. 21 (16) (2021) 6923–6930, https://doi.org/10.1021/acs.nanolett.1c02089.

[47] J. Muro-Cruces, A.G. Roca, A. López-Ortega, E. Fantechi, D. del-Pozo-Bueno, S. Estradé, F. Peiró, B. Sepúlveda, F. Pineider, C. Sangregorio, J. Nogues, Precise size control of the growth of Fe_3O_4 nanocubes over a wide size range using a rationally designed one-pot synthesis, ACS Nano 13 (7) (2019) 7716–7728, https://doi.org/10.1021/acsnano.9b01281.

[48] S.D. Oberdick, A. Abdelgawad, C. Moya, S. Mesbahi-Vasey, D. Kepaptsoglou, V. K. Lazarov, R.F.L. Evans, D. Meilak, E. Skoropata, J. van Lierop, I. Hunt-Isaak, H. Pan, Y. Ijiri, K.L. Krycka, J.A. Borchers, S.A. Majetich, Spin canting across core/shell $Fe_3O_4/Mn_xFe_{3-x}O_4$ nanoparticles, Sci. Rep. 8 (1) (2018) 3425, https://doi.org/10.1038/s41598-018-21626-0.

[49] F. de la Peña, E. Prestat, V.T. Fauske, P. Burdet, J. Lähnemann, P. Jokubauskas, T. Furnival, M. Nord, T. Ostasevicius, K.E. MacArthur, D.N. Johnstone, M. Sarahan, J. Taillon, T. Aarholt, pquinn-dls, V. Migunov, A. Eljarrat, J. Caron, C. Francis, H. W. Ånes, Hyperspy/Hyperspy: Release v1.7.3, Zenodo, 2022, https://doi.org/10.5281/zenodo.7263263.

[50] M. Varela, M.P. Oxley, W. Luo, J. Tao, M. Watanabe, A.R. Lupini, S.T. Pantelides, S. J. Pennycook, Atomic-resolution imaging of oxidation states in manganites, Phys. Rev. B Condens. Matter Mater. Phys. 79 (8) (2009) 1–14, https://doi.org/10.1103/PhysRevB.79.085117.

[51] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Microsoft Research (2000).

[52] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 24, Curran Associates, Inc, 2011.

[53] X. Xiao, M. Yan, S. Basodi, C. Ji, Y. Pan, Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm (2020), doi:10.48550/arXiv.2006.12703.

[54] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, others. Keras Tuner. 2019.

[55] J. Bergstra, D. Yamins, D.D. Cox, Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures, in: Proceedings of the 30th International Conference on Machine Learning, 2013.

[56] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: large-scale machine learning on heterogeneous systems. doi:10.48550/arXiv.1603.04467.

[57] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[58] R.J.G.B. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical density estimates for data clustering, visualization, and outlier detection, ACM Trans Knowl Discov Data 10 (1) (2015), https://doi.org/10.1145/2733381.

[59] L. McInnes, J. Healy, S. Astels, Hdbscan: hierarchical density based clustering, J. Open Source Softw. 2 (11) (2017) 205, https://doi.org/10.21105/joss.00205.

[60] O. Dzyubachyk, R.I. Koning, A.A. Mulder, M. Christina Avramut, F.G. Faas, A. J. Koster, Intensity correction and standardization for electron microscopy data, in: Proceedings of the Machine Learning Research, 2021, pp. 148–157.

[61] Z.X. Ji, Q.S. Sun, D.S. Xia, A modified possibilistic fuzzy C-means clustering algorithm for bias field estimation and segmentation of brain MR image, Comput. Med. Imaging Graph. 35 (5) (2011) 383–397, https://doi.org/10.1016/j.compmedimag.2010.12.001.

[62] R.F. Egerton, Electron Energy-Loss Spectroscopy in the Electron Microscope, Springer Science, 2011, https://doi.org/10.1007/978-1-4419-9583-4.

[63] J.B. Portals, F.P. Martínez, S.E. Albiol, 2023 Advanced computational tools for EELS data reduction and clustering, Quantitative Analysis and 3D Reconstructions. https://diposit.ub.edu/dspace/handle/2445/185702.

[64] J.IG Heaton, Y. Bengio, A. Courville, Deep learning, Genet Progr. Evolvable Mach. 19 (1–2) (2018) 305–307, https://doi.org/10.1007/s10710-017-9314-z.

[65] Y. Bengio, M. Grégoire, G.B. Orr, M. K.-R., Practical recommendations for gradient-based training of deep architectures. Neural Networks: Tricks of the Trade, 2nd ed.,

Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 437–478, https://doi.org/10.1007/978-3-642-35289-8_26.
[66] S. Ioffe, C. Szegedy, F. Bach, D. Blei, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, Lille, France 37, PMLR, 2015, pp. 448–456. Proceedings of Machine Learning Research.
[67] C.M. Bishop, M. Jordan, J. Kleinberg, B. Schölkopf, Pattern Recognition and Machine Learning, Springer, New York, 2006.