Concept-wise granular computing for explainable artificial intelligence

Antonio Luca Alfeo, Mario G. C. A. Cimino, Guido Gagliardi

Concept-wise granular computing for explainable artificial intelligence

Antonio Luca Alfeo[1,2], Mario G. C. A. Cimino[1,2] and Guido Gagliardi[1,3,4*]

[1]Dept. of Information Engineering, University of Pisa, Pisa, Italy.
[2]Research Center E. Piaggio, University of Pisa, Pisa, Italy.
[3]Dept. of Information Engineering, University of Florence, Florence, Italy.
[4]Dept. of Electrical Engineering, KU Leuven, Leuven, Belgium.

*Corresponding author(s). E-mail(s): guido.gagliardi@phd.unipi.it;
Contributing authors: luca.alfeo@unipi.it; mario.cimino@unipi.it;

Abstract

Artificial neural networks offer great classification performances, but their internal model works as a black box. This can prevent their outcomes to be employed in real-world decision-making processes, e.g. in smart manufacturing. To address this issue, the neural network should provide human-comprehensible explanations for their outcomes. This can be achieved by exploiting domain concepts and measuring their importance for the classification. To this aim, we implement an information granulation process via a neural network specifically trained to represent data instances featuring the same (different) concept's item close to (far away from) each other. By combining the representations for each concept, we obtain the so-called *conceptual space embedding*. The classification is obtained by processing it via a neural network classifier. The conceptual space embedding (i) organizes the data instances according to their concepts-wise proximity, resulting in a very informative data representation; this translates into greater classification accuracy if compared with a concept-wise approach from the state-of-the-art; and (ii) encodes each concept in one of its parts; this enables the measurement of the importance of one concept by manipulating the corresponding part of the conceptual space embedding. The proposed approach has been tested with real-world data from smart manufacturing.

Keywords: Granular Computing, Neural Networks, Representation Learning, Smart Manufacturing, Concept Importance, eXplainable Artificial Intelligence

1 Introduction and Motivation

Thanks to the adoption of the industry 4.0 paradigm, the approaches based on artificial neural networks (ANN) are more and more employed in manufacturing (Alfeo et al, 2020). If integrated with the decision-making processes, this technology can improve the quality assessments of both production and business processes (van Zelst et al, 2021), resulting in a remarkable improvement in productivity (İç and Yurdakul, 2021). Despite the unprecedented classification performances provided by state-of-the-art ANN architectures, their internal model works as a black box. As a result, the ANN classifications cannot be easily validated by domain experts (Alfeo et al, 2022b), and thus employed for the decision-making processes (Ahmed et al, 2022).

EXplainable Artificial Intelligence (XAI) approaches address this limitation by providing some explanations for ANNs' model (Alfeo et al, 2022a). Employing XAI approaches in the context of smart manufacturing can result in numerous benefits such as improved debugging of ANN-based systems, classification errors mitigation, and production cost reduction (Ahmed et al, 2022).

The explanations can feature one of the following three forms (Miller, 2019). (i) *Rule-based explanations* approximate the decision process of the ANN via a set of rules, e.g. associate classes to thresholds for some data attributes (van der Waa et al, 2021); (ii) *Instance-based explanations* associate an instance classified by the ANN to the prototype of the same class or to similar instances from different classes, to validate the classification results according to their similarities and differences (Delaney et al, 2021); and (iii) *feature importance explanations* generate a rank of all the data attributes by considering the importance of each attribute to the classification (Afchar et al, 2021).

The latter is the most used explanation form due to the many established publicly-available approaches providing a rank of the features' importance to the classification. Yet, the extensive use of feature-importance approaches has exposed their limitations. Feature importance approaches (i) may result in verbose explanations which are hardly interpretable by the decision-makers (Confalonieri et al, 2021); (ii) may be sensitive to dependencies and correlation between features, which can affect the importance score computation (Basu and Maji, 2022); and (iii) leave a significant interpretive burden to the decision-makers, that have to assess and motivate the importance of each data attribute to validate the classifications of the ANN model (Apicella et al, 2020).

XAI researchers are addressing these weaknesses by providing explanations in terms of a higher-level input representation, i.e. domain concepts (Kim et al, 2018). If properly chosen, domain concepts are supposed to be relevant for the classification, less numerous than data attributes, and clearly understandable by the decision-makers (Ghorbani et al, 2019). Concept-based approaches can be considered as an intersection between the great representation capability of opaque connectionist AI models (e.g., ANN) and symbolic AI models, that are characterized by lower generalization and scaling capabilities but are definitively easier to explain (Díaz-Rodríguez et al, 2022). As an example, the concept-based approach proposed in (Kazhdan et al, 2021) detects the occurrence of higher-level concepts in a data instance by analyzing the activation patterns of the ANN nodes. Other concept-wise approaches exploit images as input. This results in easy-to-validate explanations since the occurrence of a concept (e.g., a particular background) can be immediately recognized by a human observer (Kazhdan et al, 2021). However, in smart manufacturing, the data are usually available in tabular form. Some concept-wise approaches for tabular data may partially compromise recognition performances to obtain explanations based on concepts (Hitzler and Sarker, 2022). The approach proposed in this work addresses these challenges, resulting in improved classification performance and concept-importance computation for tabular data.

To exploit the methods used for the computation of the feature importance to obtain a measure of the *concept-importance*, it is required to transform the space of the analysis, from a feature space to a "conceptual space". This can be obtained via an information granulation process. Indeed, information granules are obtained by grouping data instances via a criterion of similarity or functionality (Song and Wang, 2016). For instance, this can be achieved via context-based clustering (Pedrycz, 1998). This approach represents the input space as information granules by clustering the data instances using a collection of predefined sets (i.e. the contexts) defined in the output space. Similarly to contexts in (Pedrycz, 1998), it is possible to define some domain-knowledge concepts and use a supervised training procedure (Alfeo et al, 2017) to learn a representation of the input space that allows granulating data instances according to their concepts-wise proximity (Qi et al, 2019).

To this aim, a representation learning approach can be employed. Representation learning approaches are designed to build manifolds by grouping similar items, e.g. in the latent space of a neural network. These approaches are used to transform the data instances into simpler representations in the latent space, which are more convenient to be employed in a classification task. Interested readers may refer to Bengio et al (2013).

The contributions of this research work can be summarized as follows:

- We propose a novel architecture employing supervised representation learning to perform concept-wise information granulation. The obtained data representation can be employed for the classification while enabling the evaluation of the importance of each concept for the ANN model.
- The proposed information granulation approach enables both the representation of the feature-wise and concepts-wise proximity between data instances. This results in a more informative input representation, and thus better classification performances if compared with other concept-based approaches.
- The importance of each concept for the ANN model is measured via a novel procedure that can also be exploited by other concept-based approaches.

The proposed architecture is tested using real-world data from smart manufacturing. Specifically, the production settings and the characteristics of the raw material are used to classify the quality level of the final product. The concepts are provided by domain experts as additional information (e.g. production operative conditions) for each production instance.

The paper is structured as follows. Section 2 presents the background and related works. Section 3 details the proposed approach. The case studies and the experimental setup are presented in Section 4. Finally, Section 5 details the obtained results, and 6 discusses results and outlines conclusions, respectively.

2 Related works

Information granulation approaches aim at abstracting the complexity of the data by re-arranging the instances into semantically structured clusters. This enables the decomposition of the original problem into more manageable sub-tasks (Song and Wang, 2016). For instance, hand-written symbols can be represented and recognized as groups of strokes (Lake et al, 2015). The academic community has recently focused on the emergence of these semantically relevant information granules in the latent space of ANNs to better explain their reasoning.

For instance, the authors in (Bau et al, 2020) indicate how artificial neurons can be explicitly triggered by human-understandable concepts, even if such networks were not explicitly trained to encode them. Lower-layer neurons can encode notions like textures and edges in image recognition tasks (Díaz-Rodríguez et al, 2022), whereas higher-layer neurons can encode notions like specific objects and abstract emotions (Zhou et al, 2015).

However, understanding if and which ANN nodes represent a human-understandable concept remains an open research question, as proved by different posthoc analyses on trained ANNs (Chen et al, 2020). For instance, the authors in (Zhou et al, 2018) show an alignment between high-level semantic concepts and some nodes in the neural network, but these nodes do not provably contain the network's full information about the concepts. In summary, rather than triggering a specific part of the neural network, the information about one concept could be scattered throughout the whole network.

To address this issue, instead of looking at individual nodes, their activation can be linearly combined to represent some predefined higher-level concepts (Kim et al, 2018). Concept Activation Vectors are vectors in the latent space of an ANN that are specifically chosen to align with predefined or automatically discovered concepts (Ghorbani et al, 2019). Still, most of those approaches work under the assumption that a trained ANN ”places” each concept in one easy-to-classify portion of its latent space. However, since the latent space was not explicitly built to have this property, there is no reason to believe that the above assumption holds (Chen et al, 2020). Rather than relying on these assumptions, it is convenient to explicitly constrain the latent space of an ANN during its training (Chen et al, 2020). As an example, the authors in (Koh et al, 2020) constrain the latent space of the ANN to represent semantically meaningful concepts while performing the classification.

Concept formation and learning is one promising granular computing research area, aimed at performing the classification by combining concepts isolated as information granules (Salehi et al, 2015). In this regard, different strategies can be implemented to learn or isolate the target concepts (Hu et al, 2014). To enable an information granulation process based on a concept-wise semantic, we exploit an approach based on supervised representation learning, which is specifically designed to organize the latent space of the ANN, i.e. aggregating (separating) instances belonging (not belonging) to the same concepts.

The proposed approach share some similarities with other existing research works. Similarly to (Kazhdan et al, 2021), the proposed approach combines concept-wise classification and representation learning; however, the approach presented in (Kazhdan et al, 2021) exploits an unsupervised AI approach designed for image data. Similarly to (Koh et al, 2020), the employed concepts are human-specified; however, using the approach presented in (Koh et al, 2020) each concept is represented via one single direction of the latent space, which may result in information loss.

To the best of our knowledge, this is the first attempt to obtain concept-wise explainability (i.e. the concept-importance) for an ANN model by implementing an information granulation process via a supervised representation learning approach.

3 Design

In this section, the design of the proposed architecture is detailed. It consists of three functional modules, i.e. the conceptual space projectors, the quality level classifier, and the concept importance computation procedure.

The Conceptual Space Projector (CSP) enables the information granulation process by grouping the instances according to one concept c. The CSP is a neural network whose latent space is constrained through a representation learning approach. This is explicitly designed to place instances characterized by the same concepts in proximity to each other in the latent space.

More specifically, the CSP consists of a fully connected multi-layer perceptron (MLP) trained by using the concept's items and the multi-similarity loss, as training labels and loss function respectively. The multi-similarity loss employs the inverse of the Euclidean distance ($d_{A,N}$ and $d_{A,P}$ in Fig. 1) as a measure of the similarity between 3 data points in the latent space, i.e. the anchor (A), the positive sample (P), and the negative sample (N). A is an instance characterized by the same concept item as P. Instead, A and N are characterized by different concept items. Via the multi-similarity loss (Eq. 1), the CSP is trained to provide a representation in which the similarity between A and P is greater than the similarity between A and N. Considering the whole training batch, the multi-similarity loss maximizes the similarity $S_{i,p}$ between each anchor i and all the positive samples $p \in P_i$ while minimizing the similarity $S_{i,n}$ between the anchor i and all the negative samples $n \in N_i$. α, β and λ are parameters of the loss function. Interested readers may find more details in (Wang et al, 2019). The implementation of the multi-similarity loss has been released by Google in September 2021 with the Python package *TensorFlow similarity*.

$$\begin{aligned} MSLoss = & \frac{1}{m}\sum_{i=1}^{m}\Big\{\frac{1}{\alpha}log[1+\sum_{p\in P_i} e^{-\alpha(s_{i,p}-\lambda)} \\ & +\frac{1}{\beta}log[1+\sum_{n\in N_i} e^{\beta(s_{i,n}-\lambda)}]\Big\} \end{aligned} \tag{1}$$

Once trained, the CSP projects the data instances in its latent space, which is ordered in a concept-wise fashion. However, not only the concept-wise similarity is represented in such a latent space. By using a supervised representation learning approach, the instances' representations are obtained as a nonlinear transformation of the feature space. As such, the CSP allows the similarity of the instances in the feature space to be represented as well. As shown in (Wang et al, 2019), adjusting the dimension of the latent space allows the proximity between instances in the feature space to be better represented. This can result in a more informative input for the subsequent classification tasks and thus in greater recognition performances.

The Conceptual Space enCodings $CSC_{i,c}$ is the representation obtained via the CSP trained for concept c (CSP_c) for the input instance i. By concatenating the p-dimensional projections obtained with all the CSPs (Fig. 2), we obtain the $c \times p$-dimensional ***Conceptual Space Embedding*** (CSE) that is ordered according to all concepts and represents one of them in each of

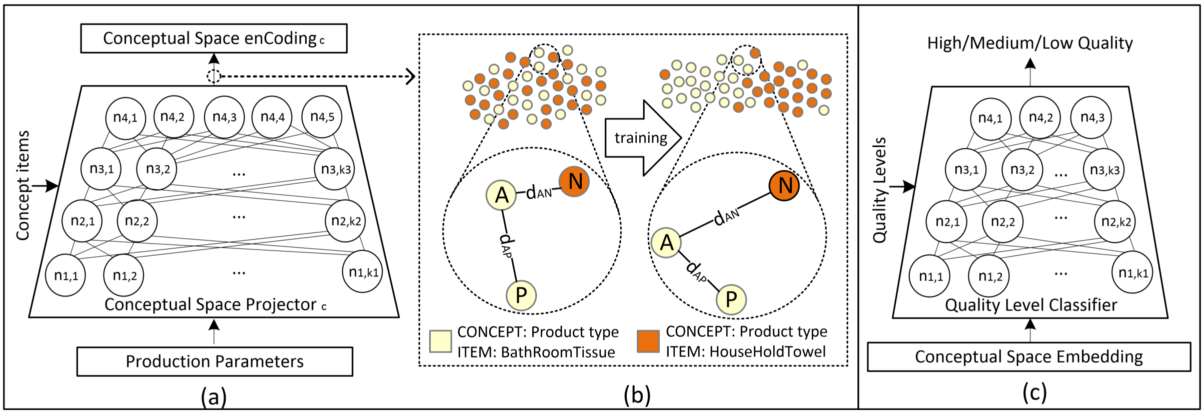

Fig. 1 (a) The architecture of the conceptual space projector, a multi layer perceptron featuring 3 hidden layers and trained via multi similarity loss. (b) The representations of the conceptual space evolution due to the training via multi-similarity loss; the distance between samples characterized by the same concept item (other concept item) is minimized (maximized) resulting in an ordered latent space. (c) The product quality level classifier, a multi layer perceptron classifier featuring 3 hidden layers and trained via the log loss function.

its parts. As represented in Fig. 2, the information granulation process is effectively provided by training the CSPs via multi-similarity loss and concatenating the representations learned for each concept. The CSE spatially organizes the instances according to their similarity in the feature space and their concepts-wise similarities.

The quality level classifier (QLC, Fig. 1) processes the CSE to classify the product quality level. It is implemented as a fully connected multi-layer perceptron trained via the log loss function (Eq. 2). In Eq. 2, m is the number of samples in the training batch, C is number of classes, x is binary indicator (0 or 1) if class label c is the correct classification for sample i, and p is the probability that sample i belongs to class c.

$$LogLoss = -\frac{1}{m}\sum_{i=1}^{m}\sum_{c=1}^{C} x_{i,j} log(p_{i,j}) \tag{2}$$

As for the CSPs, the QLC employs the rectified linear unit (*Relu*, Fig. 3 and Eq. 3) as the neurons' activation function. *Relu* is among the most widely used activation functions since it offers a great trade-off between computational complexity and recognition performance Stursa and Dolezel (2019).

$$Relu(z) = max(0, z) \tag{3}$$

Algorithm 1 shows the training procedure of the whole architecture. The procedure starts by training each CSP individually by using the *multi-similarity loss function* (Eq. 1). Once trained, the CSP_c can produce the Conceptual Space enCoding (CSC_c) for the training data. The Conceptual Space Embedding (CSE) for the training instances is obtained by concatenating the CSCs obtained with those instances for each concept. This is used to train the quality level classifier (QLC) via the *LogLoss function* (Eq. 2). As a result of the training procedure, the trained CSPs can produce a concept-wise representation of the input instances, and this can be processed by the trained QLC to classify the product quality level. Moreover, the CSE obtained with the training instances can be employed for the explanation procedure.

Once trained, the architecture can classify the product quality level by processing new instances via each component of the architecture. In Algorithm 2, $CSC_{c,t}$ is the representation obtained via each CSP_c for a new instance of the production parameters t. Those representations are concatenated to obtain the CSE_t for the new instance. The CSE_t is then provided as input to the QLC which recognizes the final product quality level q_t.

Both the trained QLC and the CSE obtained for the training instances are used to measure the importance of each concept for the classification. The concept importance computation procedure (CICP) is inspired by two explanation procedures from the state-of-the-art.

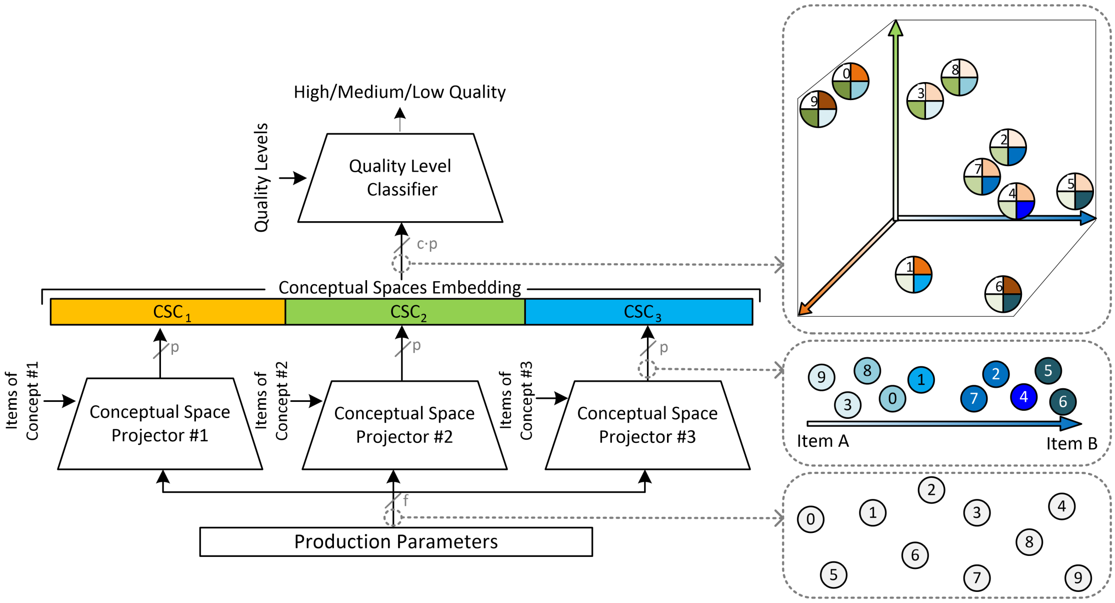

Fig. 2 Simplified representation of some instances being processed via the proposed architecture. The production parameters are the instances in the feature space (grey circles). Each *CSP* learns a representation (the *CSC*) of those instances in the latent space. Such latent space is ordered considering the items of one concept (colored circles). The concatenation of all the *CSCs*, i.e. the *CSE*, is a representation ordered according to all the concepts, allowing the data instance to cluster according to their concept-wise proximity (multi-colored circles). The *CSE* is used as an input for the quality level classifier.

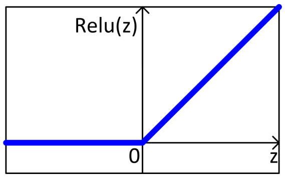

Fig. 3 The Relu activation function.

Algorithm 1 Training of the CSPs and the QLC

```
Require:
$P = \{p : \text{production parameters training samples}\}$
$Q = \{q_p : \text{quality level for } p\}$
$C = \{c : \text{concept to explain the classification}\}$
$I_c = \{i_{c,p} : \text{item of concept } c \text{ for } p\}$

Procedure:
for $c$ in $[1, \|C\|]$ do
    $CSP_c \Leftarrow MLP.train(P, I_c, MSLoss)$
end for
for each $p \in P$ do
    for $c$ in $[1, \|C\|]$ do
        $CSC_{c,p} \Leftarrow CSP_c.project(p)$
    end for
    $CSE_p \Leftarrow concat(CSC_{1,p}, ..., CSC_{\|C\|,p})$
end for
$QLC \Leftarrow MLP.train(CSE, Q, LogLoss)$
return $CSP$, $CSE$, $QLC$
```

Algorithm 2 Classification of the product quality q_t for production parameters t

```
Require:
$C = \{c : \text{concept to explain the classification}\}$
$CSP = \{CSP_c : \text{Conceptual Space Projector trained for concept } c\}$
$QLC \Leftarrow$ trained Quality Level Classifier
$t \Leftarrow$ instance of production parameters

Procedure:
for $c$ in $[1, \|C\|]$ do
    $CSC_{c,t} \Leftarrow CSP_c.project(t)$
end for
$CSE_t \Leftarrow concat(CSC_{1,t}, ..., CSC_{\|C\|,t})$
$q_t \Leftarrow QLC.classify(CSE_t)$
return $q_t$
```

The first procedure (Lucieri et al, 2020) is designed to explain the importance of concepts in an image recognition task. This approach evaluates how the recognition performance is affected by occluding the part of the image corresponding to each concept. In our case, the architecture employs tabular data (the *CSE*s are numeric arrays) thus the occlusion of each concept needs to be properly tailored, i.e. considering that the

CSE can be broken down into CSCs and that each CSC represents a concept.

The second procedure (Adadi and Berrada, 2018) is the well-known permutation importance approach. Given a trained model and a set of input instances, the permutation importance approach produces an importance score for each feature in the model. The importance is computed by randomly permuting the rows of one feature and considering how this affects the final classification performance. This procedure breaks the relationship between one feature and the target class. The drop in the performances represents how much the model depends on the permuted feature. One issue of the permutation importance approach is that it evaluates the impact of the permutations one feature at a time, and does not consider their joint contribution. For instance, if two important features are strongly correlated, permuting one of those does not necessarily remove their information from the model, and thus it does not translates into a performance drop. For this reason, more innovative feature importance approaches, e.g. SHAP (Lundberg and Lee, 2017), evaluate the importance of one feature f by considering the performances obtained with every subset of features including f.

We combine these strategies to compute the concepts importance. First, all the possible combinations of subsets of concepts are generated. Then, the concepts non included in the subset are "occluded" by permuting the CSCs corresponding to those concepts. Finally, those are concatenated to obtain the CSE and passed to the QLC to evaluate the impact on the classification performances.

More specifically, we generate a list of all possible concepts combinations called *conceptInclusionCombo*, with 2^c elements made of c booleans. Each boolean $k_{c_i,k=y}$ represents the inclusion (1) or the occlusion (0) of a concept i in the combination y. Eq. 4 represents the generalization of *conceptInclusionCombo* (CIC) whereas Eq. 5 represents an example of *conceptInclusionCombo* in the case with a number of concepts equal to 3.

$$CIC(C) = \begin{bmatrix} i_{c_1,k=1} & i_{c_2,k=1} & \cdots & i_{c_c,k=1} \\ i_{c_1,k=2} & i_{c_2,k=2} & \cdots & i_{c_c,k=2} \\ \cdots & \cdots & \cdots & \cdots \\ i_{c_1,k=2^c} & i_{c_2,k=2^c} & \cdots & i_{c_c,k=2^c} \end{bmatrix} \qquad (4)$$

$$CIC(C) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad with \; \|C\| = 3 \qquad (5)$$

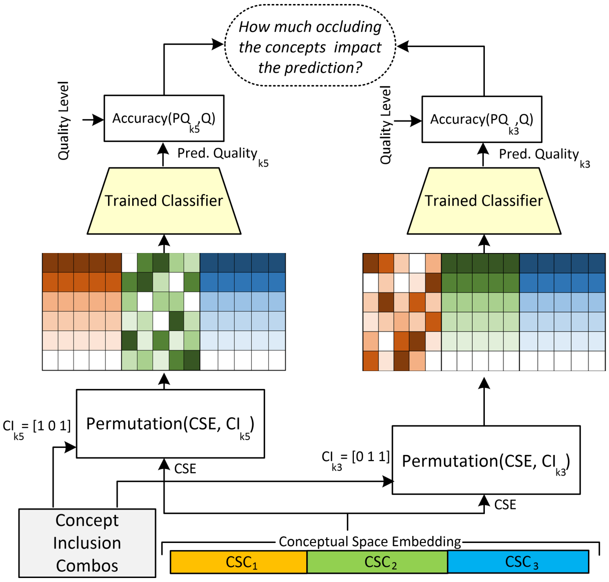

Fig. 4 Simplified representation of the concept importance computation procedure.

Once all combinations of subsets of concepts have been generated, for each combination k, the rows of CSCs corresponding to the occluded concepts are (i) permuted element by element, (ii) concatenated to obtain the CSE_k, and (iii) passed as inputs to the QLC to compute the classification performance (Fig. 4). The importance of a concept is computed as the rescaled average of the classification performances obtained with all combinations in which the concept is included. The rescaling operation is needed to decouple the classification performance and the number of included concepts; those are indeed supposed to be correlated due to the greater informativeness of a CSE with fewer occluded parts. Specifically, the performances obtained with all the concepts combination is grouped according to the number of included concepts, then normalized via a Min-Max procedure (Eq. 6) and averaged considering the combinations in which the concept is included (Algorithm 3). The resulting concept importance measure is bounded between 0 (worst case) and 1 (best case).

$$MinMax(x, X) = \frac{x - min(X)}{max(X) - min(X)} \tag{6}$$

4 Case study and experimental setup

In this section, the experimental dataset and the experimental setup are described.

We employ a real-world dataset provided by a company that produces industrial machines for manufacturing tissue paper, i.e. the Koerber Tissue. Each machine consists of two main components: the *rewinder* and the *embosser*. The rewinder unwinds the reels of raw paper layers, stacks them, and passes them to the embosser. The embosser exploits rubber and steel rolls to press and glue those layers while imprinting a motif on the paper.

Each new machine is tested using different types of paper and production settings, e.g., by varying some parameters such as the rewinder speed or the embossing pressure. Each test is recorded in a table, considering both the production settings employed and some measurements taken on the final product. These measurements include different quality-related characteristics of the final product, such as paper bulk and resistance. These characteristics can be described via levels (i.e., high, medium, low). The company would be interested in recognizing the quality level of the final product according to the specific production setup.

As with many real-world dataset, the table provided by the company is characterized by a significant presence of missing values. To cope with this issue, the data is preprocessed via the procedure depicted in Fig. 5. Firstly, all the columns and rows with a percentage of missing values greater than 50% are removed. Then, the data instances are grouped considering the categorical features that do not present missing values. For each feature, the numerical missing value of one data instance is replaced with the median of its cluster.

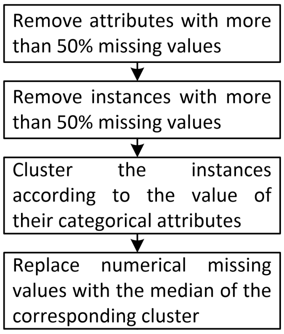
Remove attributes with more than 50% missing values
Remove instances with more than 50% missing values
Cluster the instances according to the value of their categorical attributes
Replace numerical missing values with the median of the corresponding cluster

Fig. 5 The steps of the preprocessing procedure used to handle the missing values in the data.

The resulting dataset consists of more than 650 instances and 11 informative features, as exemplified in Table 1. In Table 1 for confidentiality reasons, the specific information and names of the industrial components have been replaced with labels , e.g. A, B, and C. The features have the following meanings:

- *ID*, a unique identifier of each test measurements; it is not considered an informative feature and thus it is removed from the analysis
- *Bulk*, the quality-related characteristic of the final product; it consists of three levels (low, medium, and high)
- *Res Lon* (RLO) and *Res Lat* (RLA), the strength of the raw paper in the longitudinal and latitudinal directions; it is measured in Newtons per meter
- *Stretch* (STR), the percentage of elongation of the raw paper in the longitudinal direction, if wet
- *Paper weight* (PWE), the weight of the raw paper; it is measured in grams per square meter
- *Rubber roll hardness* (RRH), the hardness of the rubber roll used to imprint a motif on the paper; it is measured in Shore A
- *Tissue Layers Coupling* (TLC), the process aimed at coupling different tissue layers. Can be ”molded” (M), ”unmolded” (UM), or ”glued embossing” (GE).
- *Rewinder and Embosser model* (REW, EMB), a unique identifier for the models of these components

Algorithm 3 Concepts importance Computation procedure

```
Require:
$P = \{p$ : production parameters training samples$\}$
$Q = \{q_p$ : quality level for $p\}$
$C = \{c$ : concept to explain the classification$\}$
$CSE = \{CSE_p$ : Conceptual Space Embedding of $p$, separable in $CSC_{c,p}\}$
$QLC \Leftarrow$ trained Quality Level classifier

Procedure:
$CI \Leftarrow CIC(C)$
for $k$ in $[1, 2^{\|C\|}]$ do
    for $c$ in $[1, \|C\|]$ do
        if $CI_{c,k} == 0$ then
            $CSC_c \Leftarrow permutation(CSC_c)$
        end if
    end for
    $CSE_k \Leftarrow concat(CSC_1, CSC_2, ..., CSC_{\|C\|})$
    $classifications_k \Leftarrow QLC.classify(CSE_k)$
    $perfByCombo_k \Leftarrow accuracy(classifications_k, Q)$
    $inclByCombo_k \Leftarrow countIncludedConcepts(CI_{c,k})$
end for
for $n$ in $[1, \|C\|]$ do
    $PI_n = \{perfByCombo_k, k : inclByCombo_k == n\}$
    $scaledPBC_k \Leftarrow MinMax(perfByCombo_k, PI_n)$
end for
for $c$ in $[1, \|C\|]$ do
    $PC_c = \{scaledPBC_k, k : CI_{c,k} == 1\}$
    $conceptImportance_c \Leftarrow average(PC_c)$
end for
return $conceptImportance$
```

- *Bottom and Top Roll pattern* (BRP, TRP), a unique identifier of the motif characterizing the embosser rolls
- *Paper structure* (PSTR), a boolean indicating whether the raw paper is regular (No) or structured (Yes)

The proposed concept-based architecture requires some domain knowledge to specify which concept is represented by each data instance. In our case study, the concepts are chosen with the support of different domain experts who specified both the concepts and their expected importance for the classification. This is critical for the analysis since it represents the ground truth for the model explanation. This can be employed for testing the system's ability to distinguish concepts characterized by different importance levels. Specifically, we have two concepts of major and two concepts of minor importance for our classification:

- *Type of Product*: the type of product being manufactured. The items for this concept are House Hold Towel and Bathroom Rolls Tissue. Major importance for the classification.
- *Tissue Layers*: the number of tissue layers in the final product. The items for this concept are integers ranging from 1 to 4. Major importance for the classification.
- *Stretch Lat*: the percentage of elongation of the raw paper in the latitudinal direction, if wet. The items for this concept are two levels, high and low. Minor importance for the classification.
- *Dry stretch ratio*: the ratio of the raw paper resistance in the longitudinal and latitudinal directions, if dry. The items for this concept are

Table 1 Example of the data used for this study. For confidentiality reasons, specific information and the names of the industrial components have been replaced with generic labels (A, B and C). Moreover, the values of the rows in each column are randomly swapped.

ID	Bulk	RLO	RLA	STR	PWE	RRH	REW	EMB	TLC	BRP	TRP	PSTR
f4505	LOW	61.67	122.34	4.63	20.72	58	A	A	UM	A	B	Yes
ac053	HIGH	66.31	122.34	5.41	20.72	59	B	B	M	B	A	Yes
45057	LOW	52.34	118.92	4.75	23.87	62	C	A	UM	A	C	Yes
45052	MED	66.31	112.94	6.29	22.82	61	B	A	UM	C	B	Yes
45054	MED	52.37	116.81	5.07	21.93	59	B	C	M	C	C	No
45058	LOW	65.03	116.81	5.06	21.92	55	A	B	GE	B	A	No
4d886	HIGH	95.18	124.38	3.86	20.36	62	A	B	GE	A	A	No
24d883	HIGH	88.31	100.64	3.68	20.12	62	A	B	M	C	C	Yes
952f38	LOW	58.73	118.17	4.62	20.61	59	C	C	GE	A	B	No
...	...	...	...	...	...	...	...	...	...	...	...	...
952f3a	MED	61.67	118.17	4.63	20.64	57	C	C	GE	C	C	Yes

two levels, high and low. Minor importance for the classification.

To be processed by an ANN, each numerical feature is rescaled between 0 and 1 via a min-max procedure (Formulae 6). Moreover, each categorical feature is processed via a one-hot encoding procedure, i.e. replacing categorical labels with binary encodings of their enumerates.

All the experiments are performed using a Monte Carlo 10 folds validation framework. The performances obtained are presented via their mean and variance. As the main performance measure for the classification, we use the accuracy (Formulae 7). In Formulae 7, C_i is one if the classification of instance i is correct, zero otherwise.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} C_i \tag{7}$$

5 Results and discussion

This section details the experimental results obtained with the proposed architecture, i.e. the Conceptual Space Embedding (CSE).

Firstly, we show the results obtained via a parametric exploration of the CPSs dimension. Then, the classification accuracy obtained with CSE is compared against Concept Activation Vector (Hitzler and Sarker, 2022), a concept-wise approach from the state-of-the-art. Finally, the importance of the concepts is measured via the concept importance computation procedure using both CSE and Concept Activation Vector.

The most important parameters of CSE is the size of the Conceptual Spaces enCoding. On one hand, a bigger encoding may correspond to a richer data representation (Wang et al, 2019). On the other hand, a bigger encoding can result in more complex distance evaluation within the training batch and thus, potentially, into a weaker concept-wise granulation. To understand how the size of the Conceptual Space enCoding affects the classification, we consider the accuracy of the QLC by varying the size of the CSCs from 3 to 6.

Table 2 classification accuracy (percentage) by varying the size of the CSCs, average and standard deviation obtained via a 10-cross fold validation.

CSCs size	$\%Accuracy \pm St.Dev.$
6	75.73 ± 3.64
5	$\mathbf{77.88 \pm 3.79}$
4	74.65 ± 4.22
3	72.97 ± 4.36

As shown in Table 2, the best classification accuracy is obtained with a CSC size equal to 5. The potentially weaker information granulation obtained with greater CSC sizes doesn't translate into better performances. For this reason, we consider 5 as the CSC size for the following experiments.

We test the proposed architecture against a state-of-the-art concept-wise approach, i.e. Concept Activation Vector (Hitzler and Sarker, 2022). For a fair comparison, both Conceptual Space Embedding and Concept Activation Vector (CAV) employ computationally similar components: a three-layers MLP for each concept, and a three-layers MLP for the final quality classification. Each Conceptual Space Projector produces a representation consisting of 5 elements. Considering 4 concepts, this results in an input for the QLC consisting of 20 elements. On the other hand, Concept Activation Vector (CAV) produces a representation consisting of all the membership scores for each concept. Those are obtained as the concatenation of the softmax activation of the last layer of the MLPs. Considering 3 binary concepts (i.e. type of product, stretch lat, dry stretch ratio) and one 4-class concept (i.e., tissue layers), it results in an input for the QLC consisting of 10 elements.

To have a fair performance comparison, the hyper-parameters of both approaches are independently optimized. Since all the MLPs used in these architectures feature the same structure, we employ the same hyper-parameters and value set for their optimization, as summarized in Table 3.

Table 3 Values considered for the optimization of the architectures' hyper-parameters.

Hyper-parameter	Values
batch size	8, 16, 32
early stopping patience	4, 8, 12
first MLP layer	128, 64, 32
second MLP layer	64, 32, 16
third MLP layer	16, 8

The optimization of each architecture component is carried out on its own. Each *CSP* is optimized considering the accuracy obtained via a 3-NearestNeighbors classifier employing the representations provided by the *CSP*. With all the other MLPs, the accuracy obtained while classifying their target label (i.e. quality level or concept) is considered. Table 4 shows the results obtained via the hyper-parameters optimization; the classification performances are in terms of percent accuracy (average and standard deviation). According to the results shown in Table 4 CSE is significantly more accurate than CAV.

Table 4 Conceptual space embedding and Concept Activation Vector after the optimization of the hyper-parameters. classification accuracy (average and standard deviation) with a Montecarlo 10 fold validation.

Method	$\%Accuracy \pm Std$
Conceptual Space Embedding	78.19 ± 3.62
Concept Activation Vector	70.97 ± 2.68

By considering the optimal hyper-parameters for both CSE and CAV, we measure the importance of each concept via the concept importance computation procedure (Algorithm 3). We compare this measure with the ground truth about the importance of each concept, as provided by the domain experts. The obtained results are presented in Table 5.

Table 5 The concept importance values obtained with CSE and CAV, the resulting concept importance rank, and the ground truth provided by the domain experts.

Method	Concepts	Import.	Rank	G. Truth
CSE	product type	58.02	1	MAJOR
	tissue layers	55.70	2	MAJOR
	stretch lat	52.74	4	MINOR
	dry stretch r.	54.66	3	MINOR
CAV	product type	61.52	2	MAJOR
	tissue layers	74.91	1	MAJOR
	stretch lat	40.74	4	MINOR
	dry stretch r.	47.38	3	MINOR

According to the results in Table 5, CSE and CAV result in similar concept importance ranks and are both able to distinguish between concepts characterized by major and minor importance. Both ranks represent a correct solution for the concept importance evaluation. However, the ranking obtained with the two architectures do differ on the most important concepts.

Moreover, by comparing the concept importance measures obtained with CSE and CAV, the latter allows a clearer distinction between concepts of different importance. More specifically, the concept importance measures obtained with CAV are characterized by a lower variability among concepts of the same importance level (i.e. major or minor) and higher variability among concepts of different importance levels.

Indeed, since CSE allows a richer representation of the data, when a concept is occluded the

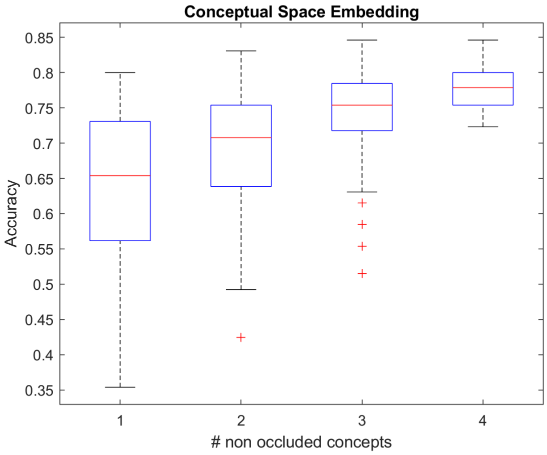

Fig. 6 Accuracy by number of non occluded concepts with Conceptual Space Embedding.

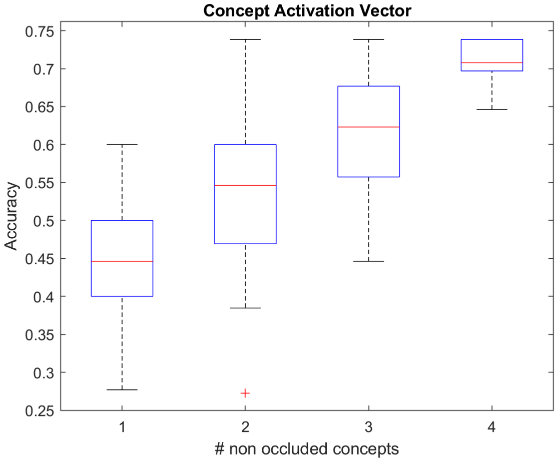

Fig. 7 Accuracy by number of non occluded concepts with Concept Activation Vector.

QLC can still rely on the information from the input space obtained via the other CSCs. For this reason, the performances of QLC are less affected by concept occlusion with CSE.

These considerations are confirmed by the box plots shown in Fig. 6 and Fig. 7. Considering the accuracy obtained with the QLC by varying the number of non-occluded concepts, CSE is less affected by the number of occluded concepts if compared to CAV. Indeed, the accuracies obtained with CAV grow almost linearly with respect to the number of non-occluded concepts. On the other hand, CSE results in a more asymptotic accuracy growth. This also suggests that CSE needs fewer concepts to achieve the same or greater results with respect to CAV, thanks to the richer data representation learned by the CSPs.

6 Conclusion

In this work, a novel explainable artificial intelligence architecture is proposed, i.e., the Conceptual Space Embedding (CSE).

CSE employs an information granulation process to effectively decompose the target problem in a concept-wise fashion. The information granulation is provided via a supervised representation learning approach aimed at projecting the data instances according to their concept-wise proximity. The projections obtained with each concept are concatenated to obtain the conceptual space embedding, which organizes the instances according to all concepts and represents one of them in each of its segments.

This allows using an occlusion-like explanation procedure to measure the importance of each concept for the classification based on the CSE.

The considered case study addresses the recognition of the quality level of the final product in a real-world smart manufacturing. The domain experts provided the concepts employed in this study, together with their expected importance levels (major and minor) for the classification.

To the best of our knowledge, this is the first architecture providing concept-wise explainability (i.e. the concept importance measure) by implementing an information granulation process via supervised representation learning.

The proposed approach is compared against a state-of-the-art concept-wise approach, i.e. Concept Activation Vector. The obtained results confirm that CSE enables a more informative data representation, and thus significantly better classification accuracy.

Finally, the importance of each concept for the classification is measured via a novel procedure that can also be exploited by other concept-based approaches. Indeed, this is employed to measure the importance of each concept both for CAV and CSE. By considering the ground truth for the importance of each concept, both CSE and CAV result in correct concepts' importance rank.

Moreover, by considering how the classification accuracy changes according to the number of non-occluded concepts, CSE results to be less sensitive to concept occlusion if compared to CAV. This suggests that CSE can obtain greater classification performances by employing fewer concepts. Indeed, the median value of the accuracy with one non-occluded concept is 44.62% for CAV and 65.39% for CSE. On the other hand, the greater sensitivity of CAV to concept occlusion results in a clearer separation between different concepts' importance levels.

Future developments will focus on improving the concept importance computation procedure, to allow different importance levels to be more clearly separated.

Moreover, we will employ an information granulation process consisting of different granulation levels, resulting in a hierarchical concept-wise approach. A first granulation level can represent concepts very related to the feature space. The projection obtained via the first level of *CSP*s can be passed to the next information granulation level. The latter can represent higher-level abstraction concepts, i.e. specific to the domain knowledge. The resulting architecture should be able to further improve the classification performances and explain the ANN model at different abstraction levels.

Declarations

Fundings. This work has been partially supported by: (i) the company Koerber Tissue in the project “Data-driven and Artificial Intelligence approaches for Industry 4.0”; (ii) the University of Pisa, in the project PRA_2022_101 project “Decision Support Systems for territorial networks for managing ecosystem services”; (iii) the Tuscany Region in the framework of the SecureB2C project, POR FESR 2014-2020, Project number 7429 31.05.2017; (iv) the Italian Ministry of University and Research (MUR), in the framework of the ”Reasoning” project, PRIN 2020 LS Programme, Project number 2493 04-11-2021, and of the FISR 2019 Programme, under Grant No. 03602 of the project “SERICA”.

Conflict of Interest. The authors declare that they have no conflict of interest, and that an ethical statement is not applicable to this research.

Author Contribution. All authors contributed to study conception and design, material preparation, data collection and analysis, as well as experiments and manuscript writing. All authors read and approved the final manuscript.

Data Availability. Due to confidentiality agreements, supporting data can only be made available to bona fide researchers subject to a non-disclosure agreement.

References

Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE access 6:52,138–52,160

Afchar D, Guigue V, Hennequin R (2021) Towards rigorous interpretations: a formalisation of feature attribution. In: International Conference on Machine Learning, PMLR, pp 76–86

Ahmed I, Jeon G, Piccialli F (2022) From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. IEEE Transactions on Industrial Informatics 18(8):5031–5042

Alfeo AL, Cimino MGC, Egidi S, et al (2017) Stigmergy-based modeling to discover urban activity patterns from positioning data. In: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, Springer, pp 292–301

Alfeo AL, Cimino MG, Manco G, et al (2020) Using an autoencoder in the design of an anomaly detector for smart manufacturing. Pattern Recognition Letters 136:272–278

Alfeo AL, Cimino MG, Vaglini G (2022a) Degradation stage classification via interpretable feature learning. Journal of Manufacturing Systems 62:972–983

Alfeo AL, Cimino MGC, , et al (2022b) Automatic feature extraction for bearings' degradation assessment using minimally pre-processed time series and multi-modal feature learning. In: Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics (IN4PL 2022)

Apicella A, Isgrò F, Prevete R, et al (2020) Middle-level features for the explanation of classification systems by sparse dictionary methods. International Journal of Neural Systems 30(08):2050,040

Basu I, Maji S (2022) Multicollinearity correction and combined feature effect in shapley values. In: Australasian Joint Conference on Artificial Intelligence, Springer, pp 79–90

Bau D, Zhu JY, Strobelt H, et al (2020) Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences 117(48):30,071–30,078

Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35(8):1798–1828

Chen Z, Bei Y, Rudin C (2020) Concept whitening for interpretable image recognition. Nature Machine Intelligence 2(12):772–782

Confalonieri R, Weyde T, Besold TR, et al (2021) Using ontologies to enhance human understandability of global post-hoc explanations of blackbox models. Artificial Intelligence 296:103,471

Delaney E, Greene D, Keane MT (2021) Instance-based counterfactual explanations for time series classification. In: International Conference on Case-Based Reasoning, Springer, pp 32–47

Díaz-Rodríguez N, Lamas A, Sanchez J, et al (2022) Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case. Information Fusion 79:58–83

Ghorbani A, Wexler J, Zou JY, et al (2019) Towards automatic concept-based explanations. Advances in Neural Information Processing Systems 32

Hitzler P, Sarker M (2022) Human-centered concept explanations for neural networks. Neuro-Symbolic Artificial Intelligence: The State of the Art 342(337):2

Hu H, Pang L, Tian D, et al (2014) Perception granular computing in visual haze-free task. Expert systems with applications 41(6):2729–2741

İç YT, Yurdakul M (2021) Development of a new trapezoidal fuzzy ahp-topsis hybrid approach for manufacturing firm performance measurement. Granular Computing 6(4):915–929

Kazhdan D, Dimanov B, Terre HA, et al (2021) Is disentanglement all you need? comparing concept-based & disentanglement approaches. arXiv preprint arXiv:210406917

Kim B, Wattenberg M, Gilmer J, et al (2018) Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International conference on machine learning, PMLR, pp 2668–2677

Koh PW, Nguyen T, Tang YS, et al (2020) Concept bottleneck models. In: International Conference on Machine Learning, PMLR, pp 5338–5348

Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. Science 350(6266):1332–1338

Lucieri A, Bajwa MN, Dengel A, et al (2020) Explaining ai-based decision support systems using concept localization maps. In: International Conference on Neural Information Processing, Springer, pp 185–193

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. Advances in neural information processing systems 30

Miller T (2019) Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence 267:1–38

Pedrycz W (1998) Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE transactions on neural networks 9(4):601–612

Qi J, Wei L, Wan Q (2019) Multi-level granularity in formal concept analysis. Granular Computing

4(3):351–362

Salehi S, Selamat A, Fujita H (2015) Systematic mapping study on granular computing. Knowledge-Based Systems 80:78–97

Song M, Wang Y (2016) A study of granular computing in the agenda of growth of artificial neural networks. Granular Computing 1(4):247–257

Stursa D, Dolezel P (2019) Comparison of relu and linear saturated activation functions in neural network for universal approximation. In: 2019 22nd International Conference on Process Control (PC19), IEEE, pp 146–151

van der Waa J, Nieuwburg E, Cremers A, et al (2021) Evaluating xai: A comparison of rule-based and example-based explanations. Artificial Intelligence 291:103,404

Wang X, Han X, Huang W, et al (2019) Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5022–5030

van Zelst SJ, Mannhardt F, de Leoni M, et al (2021) Event abstraction in process mining: literature review and taxonomy. Granular Computing 6(3):719–736

Zhou B, Khosla A, Lapedriza À, et al (2015) Object detectors emerge in deep scene cnns. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR)

Zhou B, Bau D, Oliva A, et al (2018) Interpreting deep visual representations via network dissection. IEEE transactions on pattern analysis and machine intelligence 41(9):2131–2145