Technical Report

IRI-TR-20-02

Fuzzy clustering application on failure rate prediction in Water Distribution Networks

Joan Vendrell Gallart
Congcong Sun
Vicenç Puig
Gabriela Cembrano

June, 2020

Institut de Robòtica i Informàtica Industrial

Abstract

In this report a new approach of failure rate prediction is presented based on Fuzzy Clustering technic for a more deterministic and accurate implementation of neuro-fuzzy systems. This technique is compared with two benchmark methods: Artificial Neural Networks (ANN) and Adaptative Neuro-Fuzzy Inference Systems (ANFIS). Furthermore, an analysis of the necessary inputs is carried out with the goal of defining the useful information needed for the models. All these methods are applied to real data of Barcelona water distribution system and the models predictions are compared with calculated pipe failure rate.

Institut de Robòtica i Informàtica Industrial (IRI)
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona, Spain

Tel (fax): +34 93 401 5750 (5751)
`http://www.iri.upc.edu`

Corresponding author:

Congcong Sun
`csun@iri.upc.edu`
`http://www.iri.upc.edu/staff/csun`

1 Introduction

Water is a basic necessity, for this reason, even that sometimes it slip by unseen, Water Distribution Systems (WDS) are a key element for a good development of a modern society. The system complexity grow significantly when we talk about urban WDS were more factors have influence over more elements in the network. That kind of systems are expected to grow even more in complexity due to the growth of urban regions. In not so far future, the most of habitants in the world will live in a city, actually, the concept of *megacities* is growing progressively more.

In WDS, an interesting problem is to understand the relationship between the network parameters and pipes wear. So, we will be able to predict future problems in the systems and be able to introduce manteinance operation, reparations or replacing of the damaged pipes before breakdowns. It is important to notice that unexpected breakdowns involve economic and capital losses for the society. Furthermore, they supose a temporal interruption of water service and that can be a big problem for sensitive entities such as industry or hospitals.

There is no analytical formula to calculate the *failure rate* of the pipes. That is the reason why it is important to find the relationship between the parameters of the system. In a distribution network, a lot of different parameters take part into its behaviour, from more intuitive ones such as *Diameter*, *Lenght* or *Pressure*, to others such as *corrosion level* or *the usage of the pipes*. Some of these factors are not measurable and, moreover, some of them are not relevant for the calculation of *failure rate*. So, it is imporant to understand which inputs are influential. To reduce the number of inputs involve a reduction in the number of factors that have to be measured by sensors, therefore, it means a lower expenditure in the monitorization of the system. At the same time, it involves a faster and efficient predictive model. That is the reason why in this report there is an analysis of the necessary inputs in the model, in order to decide which are the most important parameters and avoid using superfluous ones.

The relationship between parameters is non-linear, for this reason, Machine Learning methods have acquired an strong relevance in this kind of analysis, just as in other predictive problems in other engineering fields. In particular, Neural Networks is the benchmark method in predictive problems. It has overcome over all other Machine Learning methods. Focusing on WDS management, the two most used Data-Driven Models are *Artificial Neural Networks* (ANN) and *Adaptative Neuro-Fuzzy Inference Systems* (ANFIS). It is important to understand that this kind of models have no way of assuring convergence and stability, so it is very important to tune correctly the parameters of the models.

ANN are an open model with lots of hyperparameters to tune such as number of neurons per layer, number of layers or activation functions, that means a big dimensionality of possibilities and, therefore, a more difficulty in finding the best combination. With regard to ANFIS, this kind of models are based on adding previous knowledge of the system in the model throught fuzzy logic and that is the reason why we can consider that they are more enclosed than ANN. However, there is also an important, and difficult, step in defining the *membership functions* to work with fuzzy logic. The experience of this research shows that *membership functions* are the most important parameters to get good performance with this model. In this report, an exhaustive analysis of this two methods is done.

Furthermore, as an improvement of actual methods, in this report a new approch in neuro-fuzzy systems is presented. The new method uses Fuzzy C-Means algorithm to replace *membership functions* in ANFIS model. The main idea is not classifying the data with functions, but using clusters to do so. That new approch improve ANFIS models in the way that there is no need to initializate *membership functions*, since Fuzzy C-Means is an unsupervised *Machine Learning* method, what means that is trained itself based on the data. That suppose avoiding errors in a sensitive step of ANFIS designing.

The structure of this report is as follows. First, the Case Study is presented with an explanation of the dataset used to test the models. In the section 3, the studied models are shown. In the section 4, there are exposed the methodology followed in the research and how we have trained and tested the models. In the section 5, there are exposed the results of the research. Finally, in the section 6, there are some conclusions of the work done. Moreover, some interesting plots are shown in appendixs.

2 Case Study

This research is based on real historical data from the water distribution system of the city of Barcelona. The first step in the research was filtering raw data to get a valid datset. There were three considerations. First, in raw data each row meant an operation over a pipe, however we just got the rows that meant a failure in the pipe, not in its enviornment due to foundations for example. Second, we delated wrong data, for example, some rows had 'NaN' values in pressure data. Third, after all the filters were done, we just took the needed information. For exemple, there were some columns related to the operation company that were not relevant for the problem.

Once the data was filtered, we compute the expected *failure rate* for each pipe. This is a very important operation, because all the models designed try to learn how to get this rate using pipe's parameters. There are different ways of defining *failure rate* [5], in our case study, we have calculated it as follows,

$$\lambda = \frac{number\ of\ failures}{pipe's\ age} \tag{1}$$

In other literature, *failure rate* is computed also using the kilometer where the leak appears. In our case study, we have not considered this option as we do not have that information. Another important aspect to explain is how the *failure rate* has been calculated with our dataset. The same pipe may have had more than one failure. However, the age has been considered as the difference between the *Installation Year* of the pipe and the *Break Year* when the break was detected and the pipe was repaired.

Finally, we have worked with a dataset of dimensions 1617x9, where the information considered is exposed in the table 1. In table 1, the used inputs are compared with other common inputs considered in the checked literature, [5] and [2].

Case Study	[5]	[2]
Age	Age	NOPB
Material	Diameter	Material
Diameter	Lenght	Diameter
Lenght	Pressure	Lenght
Usage	Height	Traffic
Pressure		
Temperature		
CodiPis		

Table 1: Comparision between considered data in our case study and in checked literature.

With regards to the table 1, *CodiPis* is a codification of the zone where the pipe is installed. It is very common in WDS to divide the system in segments based on the depth of the installation. Not all the city is at the same altitude. This factor can also be called as *Height* in other papers. In this report, we have finally rejected using this information. Another factor that must be explained is *NOPB* or *Number Of Previous Breaks*. This input shows a different way of working with the data. In our case, we have agrouped all the breaks in a pipe together. However, it could also be interesting to not do so and work with *NOPB* because once the pipe failure is repaired, the pipe has a different resistence than before. It is important to explain that *Temperature* has been exctracted from the historical meteorological open database of Barcelona local government.

So, even this report tries to give a basic patterns to define a predictive model over WDS, the truth is that it might have some modifications depending on the initial considerations over the problem.

3 Models

In this section more information about the tested models is exposed. As it is explained in the introduction, some models has been tested but only the ones that have achieved good performances are explained. The order of exposure is the order of tested models. First, we talk about state-of-the art DDM, ANN and ANFIS. And, finally, the Fuzzy C-Means approach is shown.

3.1 Artificial Neural Networks

Artifcial Neural Networks are the basic feed-forward configuration with a linear combination of the information in *neurons* with *activation functions* that add non-linearity to the model. As shown in Figure 1.

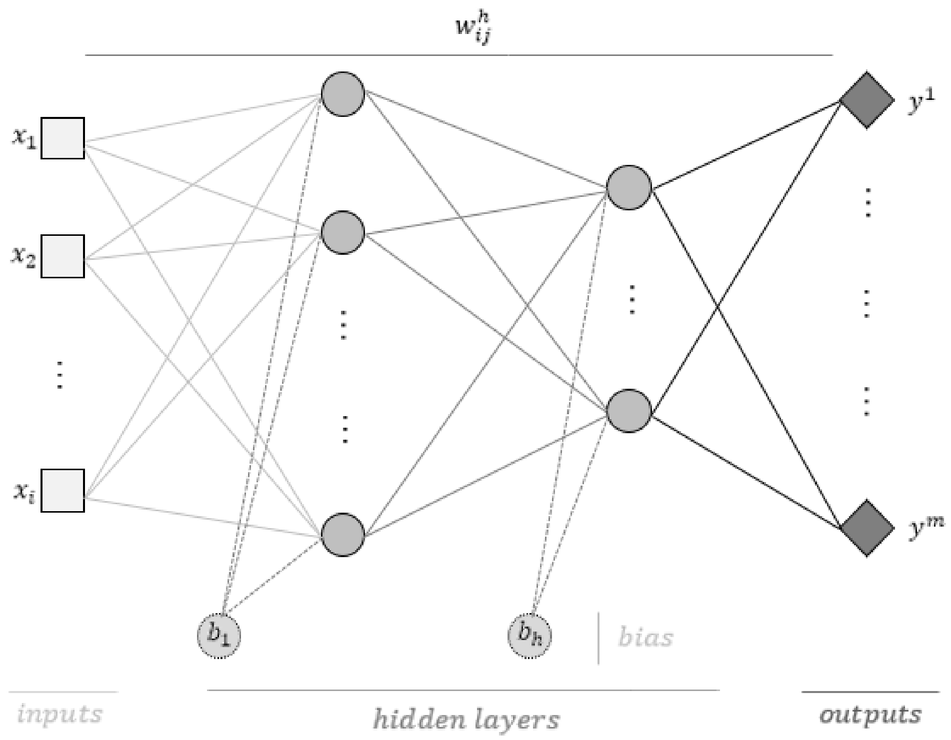


Figure 1: Artificial Neural Network generic representation. Where x_i are inputs, y^m are outputs, b_i are biases and w_{ij}^h is the weight of the connexion between neuron i and neuron j in the layer h.

Inspired by [5], we have tested different ANN models. As it has been mentioned before, there are huge number of possible models using this technic, for this reason, working with just one configuration could lead to omit better approaches. A common error in ANN is to design oversized networks. That is why we have dimensioned our networks in function of the number of imputs used. Specifically, we have designed 9 proportional ANN. More details of the number of neurons and layers is shown in the table 2. As *activation functions* we have used the tangent sigmoid function (2) after each layer which is a common activation function because its domain characteristics as it is explained in [3].

$$tan\ sig(x) = \frac{2}{1 - e^{-2x}} - 1 \tag{2}$$

Apart from the 9 proportional ANN, we have also disigned a 10th ANN that pretends to emulate fuzzification layer in ANFIS models, see table 2. The fact is that the first layer of the ANN has as neurons as *membership fucntions* it would has if it was an ANFIS model. The main idea behind this model is to see if this ANN is able to learn by their own the performance of a fuzzification layer without previous *membership functions* initialization.

Model	**num neurons 1^{st} layer**	**num neurons 2^{nd} layer**
ANN1	n	-
ANN2	2n	-
ANN3	3n	-
ANN4	n	n
ANN5	n	2n
ANN6	n	3n
ANN7	2n	n
ANN8	2n	2n
ANN9	2n	3n
ANN10	MF	max(n+1,3)

Table 2: Resum of implemented ANN models where n represents *num inputs* and MF is equal to the number of a theoretical *fuzzification layer*.

3.2 Adaptative Neuro-Fuzzy Inference System

Adaptative Neuro-Fuzzy Inference Systems are a more closed environtment. As shown in Figure 2, this kind of models are defined in 5 layers.

In the first layer, *fuzzification layer*, *membership functions* are defined. The idea is, for each input, define the number of fuzzy labels using distribution functions. That is a good way of considering non-categorical data. Then, in the *rule layer*, the different labels are combined in a logical combination IF-THEN. In the third layer, a *normalization* is done. That three layers are known as Fuzzy Inference Systems (FIS) because they appliy the fuzzy logic to the model. Then, the last two layers are just a Neural Network. The fourth layer is known as *defuzzification* because model inputs are weighted using values obtained in FIS. Finally, after a linear combination, all is sum in the fifth layer. More information can be found in [1].

In ANFIS models, the most sensitive part in the designing are *membership functions* initialization. As it has been explained before, they are just distribution functions, so a huge number of them can be proposed, from Triangular to Gaussian ones. There is no best type of function, it all depends on the data. So, a previous exploration of the data is needed. If they are not correctly defined, it would lead to an error and a not covnergence of the model. In our particular case, we have tested ANFIS models with three different *membership functions*. Rectangular functions (3) for non-categorical data and then, Triangular functions (4) or Gaussian funcitons

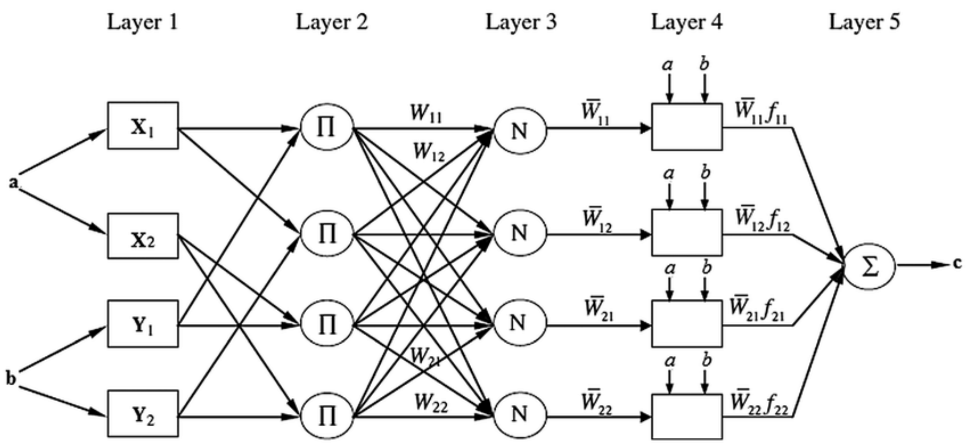


Figure 2: Adaptative Neuro-Fuzzy Inference System generic representation.

(5) for categorical data.

$$Rectangular(x) = \begin{cases} 0 & x \leq a \\ 1 & a \leq x \leq b \\ 1 & b \leq x \leq c \\ 1 & c \leq x \leq d \\ 0 & d \leq x \end{cases} \tag{3}$$

$$Triangular(x) = \begin{cases} 0 & x \leq a \\ \frac{1}{b-a}x + \frac{a}{a-b} & a \leq x \leq b \\ -\frac{1}{c-b}x + \frac{c}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \tag{4}$$

$$Gaussian(x) = exp\left[-\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{5}$$

More detailed information of the models can be found in the table 3 and in annex 1.

3.3 Fuzzy Clustering Systems

In fuzzy clustering approach the goal is to improve ANFIS models by changing *fuzzification layer* for a fuzzy clustering technic. Specifically, Fuzzy C-Means algorithm (1). This is a well-known *Unsupervised Machine Learning* method that has achieved good results and has low execution and training cost.

Input	Number of labels	Label	Rectangular (a,b,c,d)	Triangular (a,b,c)	Gaussian (μ,σ)
Age	3	Little	-	0,15,30	15,6.5
		Young	-	25,50,75	50,11.5
		Old	-	60,90,120	90,14
Material	12	Steel	0.4,0.6,1.4,1.6	-	-
		Galvanized iron	1.4,1.6,2.4,2.6	-	-
		Asbesto	2.4,2.6,3.4,3.6	-	-
		Reinforcement concrete weld junction	3.4,3.6,4.4,4.6	-	-
		Condensed reinforcement concrete	4.4,4.6,5.4,5.6	-	-
		Soft smelting	5.4,5.6,6.4,6.6	-	-
		Grey smelting	6.4,6.6,7.4,7.6	-	-
		Palosca	7.4,7.6,8.4,8.6	-	-
		Polyvinyl chloride	8.4,8.6,9.4,9.6	-	-
		Fiberglass polyester	9.4,9.6,10.4,10.6	-	-
		High density polythene	10.4,10.6,11.4,11.6	-	-
		Low density polythene	11.4,11.6,12.4,12.6	-	-
Diameter	3	Small	-	0,80,160	80,35
		Medium	-	150,200,250	200,20
		High	-	240,300,360	300,25
Lenght	3	Small	-	0,20,40	20,5
		Medium	-	30,70,110	70,15
		Large	-	100,300,500	300,100
Usage	3	Distribution	0.4,0.6,1.4,1.6	-	-
		Transport	1.4,1.6,2.4,2.6	-	-
		Production	2.4,2.6,3.4,3.6	-	-
Pressure	3	Low	-	0,25,50	25,7.5
		Medium	-	40,60,80	60,7
		High	-	70,90,110	90,8
Temperature	3	Low	-	0,10,15	10,1.5
		Medium	-	13,17,21	17,1
		High	-	20,25,30	25,1.5

Table 3: Resum of *membership functions* defined for each input. Parameters for each *membership functions* type can be seen at equations (3), (4) and (5).

Algorithm 1 Fuzzy C-Means algorithm

Data: Set of points X (vector of lenght N), allowed error $\epsilon \approx 10^{-12}$ and parameter $m \in \{1, 2\}$
Results: Vector of clusters centers C, where lenght of C $c \in \{2, N\}$ is the number of centers
  initializate membership matrix $U^0_{c,N}$;
  (where position c, N is the fuzzy logic value of point N with respect to cluster c)
  while Not convergence **do**
    for $i = 1...c$ **do**
      for $j = 1...N$ **do**
        $U_{ij} \leftarrow \sum_{k=1}^{c}[(\frac{dist(x_{ij}, C_k)}{dist(\hat{C_k})})^{\frac{2}{m-1}}]^{-1}$
      end for
    end for
    if $||U^{(y+1)} - U^{(y)}|| \leq \epsilon$ **then**
      convergence;
    else
      Not convergence;
    end if
  end while

Therefore, the number of free parameters in designing process is reduced. With this model, the designer only have to define the number of clusterings needed for each input. It depends on the range of the data, but as a thumb rule, normally three labels are defined for each input with exception of non-categorical data where the number of labels is equal to the number of categories.

To go further, in this report a pre-training of the C-Means algorithm is also done to define the number of clusters for each input in an autonomous way looking at the *fitting coefficient* (6) of clusters using different number of centers, see algorithm 2 and annex 2. The number of clusters autonomously chosen by the algorithm can be found in table 4.

Fitting coefficient is computed by normalized Dunn's partition coefficient [4].

$$FC(U_{c,N}) = \frac{(\frac{1}{N}\sum_{k=1}^{k}\sum_{i=1}^{N} m_{ik}^2) - \frac{1}{K}}{1 - \frac{1}{K}} \tag{6}$$

Where $U_{C,N}$ is the membership matrix where each column represents a point of the dataset with lenght N and each row represents a cluster of the chosen number of clusters C. Each position of the matrix is the value m_{ic} that represents the unknown membership of the point i in the cluster c.

Algorithm 2 Autonomous Fuzzy C-Means algorithm initialization

Data: Set of points X (vector of lenght N)
Results: Vector F of *fitting coeficients* for each tried iteration with different number of clusters.
for $c = 1...N$ **do**
  Do Algorithm 1
  Compute FC with equation (6)
end for
Chose the number of centers with the higher FC in F

Inputs	Clusters
Age	2
Material	10
Diameter	12
Lenght	2
Usage	2
Pressure	2
Temperature	2

Table 4: Number of clusters autonomously chosen for C-Means implementation.

That means that for each input, two Fuzzy Clustering Systems are tested. One using predefined number of centers and another using an autonomous definition of the number of centers for each input.

4 Methodology

In this section, the proposed methodology in the experiments is exposed. First, we explain how we have done the input analysis. Then, we explain some particular aspects of the models and, finally, we expose the training algortihm.

All the models have been implemented using Pytorch in Google Colaboratory over a GPU. The data has been split as follows: 85% for Training set, 10% for Test set and 5% for Validation set to check if the models where overfitting over Training set every 100 epochs.

As it has been explained in table 1, seven input has been considered as intersting: Age, Material, Diameter, Lenght, Usage, Pressure and Temperature. The input study consisted in starting testing the models for the most important input and, then, adding inputs according to their importance one by one and testing again the models over more dimensionality. To do that, the first step is to define the order of importance between the imputs. We have decided to compute the correlation between *failure rate* and each proposed input and order them using this criterion, see table 5 and Figure 3.

Inputs	Correlation
Age	0.373018
Material	0.299201
Diameter	0.091248
Lenght	0.012711
Usage	0.009001
Pressure	0.006681
Temperature	0.006353

Table 5: Correlation of inputs with respect to *failure rate*.

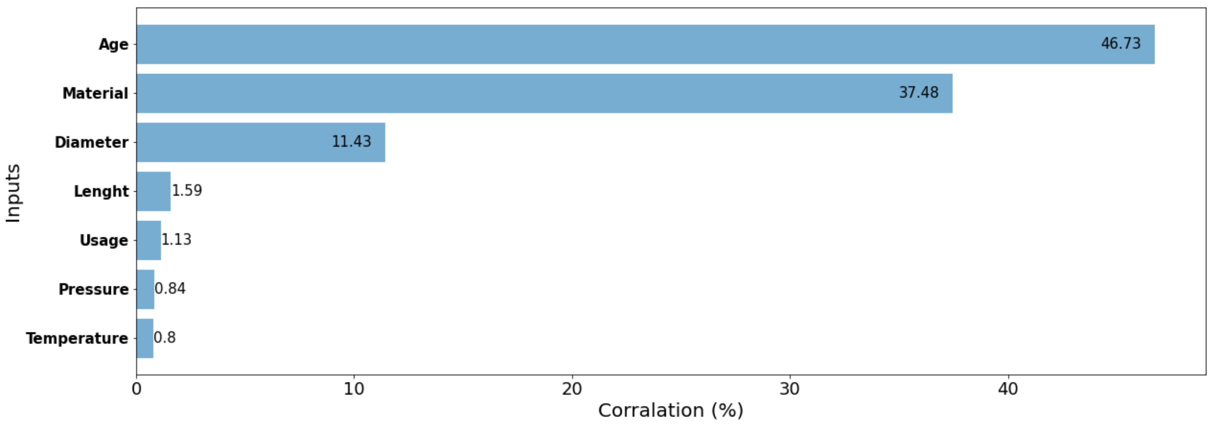
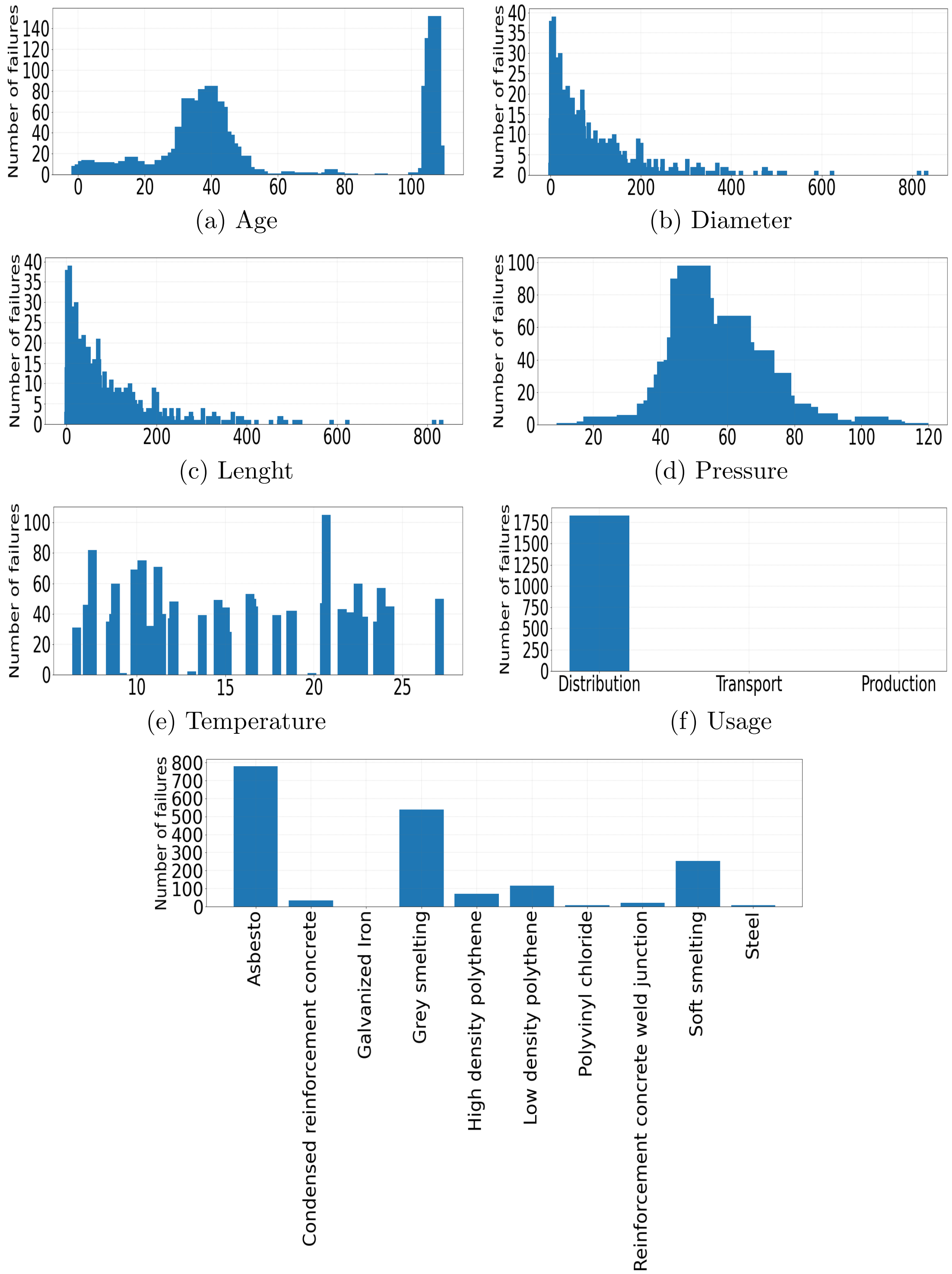


Figure 3: Normalized corralation of each input with respect to *failure rate*.

Also as a part of input analysis, the behaviour of the leaks with respect to the different inputs has been ploted. As it can be found in Figure 4, the older the pipe, the more number of leaks appears. Pressure and temperature have also a quite proportional relation with the number of leaks. Whereas, diameter and lenght are inversely proportional to the leaks. The shorter the diameter, the more errors appears, and the same happens with lenght. Finally, looking at non-categorical data, it is observable that the majority of leaks happens in the pipes used for distribution. Talking about the materials, pipes of asbesto presents more breaks than the others, meanwhile galvanized iron seems to be the safer material.

So, for each exposed model we did seven different analysis with different inputs. That means 35 experiments. The models have been tested in the order exposed in section 3: ANN, ANFIS with Triangular, ANFIS with Gaussian, ANFIS with fuzzy clustring and ANFIS with fuzzy clustering with autonomous number of centers definition. This is the same order we use to



(a) Age (b) Diameter

(c) Lenght (d) Pressure

(e) Temperature (f) Usage

(g) Material

Figure 4: Number of failures in function of analyzed inputs.

expose the results. It is important to notice that, ANFIS implementation is based on ANFIS Pytorch library of James Power from Maynooth University [6]. And for C-Means algorithm Scikit Fuzzy library has been used.

With regard to the training algorithm, it is important to explain that for ANFIS models and hybrid learning had been proposed. That means that the gradient graph starts from *membership functions* to the last layer. So, the *fuzzification layer* and *rule layer* are also modified during training. In some literature, some researchers prefer not to modify this layers during training as they believe they have good previous knowledge of their problem and they have defined that parameters correctly. In fuzzy clustring approach that is not done. The main reason is because it would be quite exepensive in time and computation cost, however, for future studies, it could be interesting to modify a little the clusters during training to get a better adaptation to the data.

All the models have been trained with the same number of epochs, batch size and the same optimizer. More details can be found in the table 6.

Epochs	5000
Batch size	1374
Optimizer	Rprop
Learning rate	0.0001

Table 6: Training hyperparameters.

To define this training parameters it is not trivial. Indeed, it is crucial to get a convergence of the networks. Some advices from our research is to use a big batch size to get smoother behaviour of the training and avoid oscillations. And, also, avoid using Stochastic Gradient Descent (SGD). In sevaral tests done, SGD was a problem in the optimization because it gets stuck or originate oscillant behaviour of loss. The best optimiztion techincs are adaptative ones such as Rprop, Adam or RMSprop with low learning rate. An adaptative techinc is that one that considers gradients from previous backpropagations to optimize parameters. An it is a good way of not getting stuck in local minima and getting a smoother behaviour.

Finally, as loss function and test function, two function have been defined. On the one hand, Root Mean Square Error (RMSE) which is a metric of the absolut distance between the expected value and the obtained value. The lower the RMSE, the lower is the distance between predicted *failure rate* (y_{pred}) and real *failure rate* (y), so the better is the model.

$$RMSE(y, y_{pred}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - y_{pred})^2} \tag{7}$$

On the other hand, Index of Accuracy (IOA) which is a relative metric between the expected values and the obtained ones. To do that, this metric uses the mean of expected *failure rate* ($\hat{y}$). As its name indicate, it is a metric of the accuracy of the model. So, we pursue the higher IOA as possible.

$$IOA(y, y_{pred}) = 1 - \frac{\sum_{i=1}^{n}|y_{pred} - y|^2}{\sum_{i=1}^{n}(|y_{pred} - \hat{y}| + |y - \hat{y}|)^2} \tag{8}$$

Both metrics have been used to test the algorithms. However, in training the use of both have been alternate. That means that, for example, ANN have been trained using a loss function of,

$$Loss(y, y_{pred}) = RMSE(y, y_{pred}) + (IOA(y, y_{pred} - 1) \tag{9}$$

Note that we use $IOA(y, y_{pred} - 1)$. That is made because we want to minimize $Loss(y, y_{pred})$.

Meanwhile, ANFIS models use a loss function of,

$$Loss(y, y_{pred}) = MSE(y, y_{pred}) \tag{10}$$

This information can be found in detail in the tables of Section 3. The decision of using this combinations for loss functions is based on the performance. We have tested all the model using only RMSE, MSE or IOA and using RMSE and IOA at the same time and we have observed than for some models have obtained better results with a single loss funcion and other using a combination of both. However, this choice does not change too much the results, so the most important advice is to use one of this possible loss functions to train similar models.

5 Results

There are several ways of exposing the results. For a better understanding, we have divided the results depending on the number of inputs. The order of the models in each section follows the order in which we have exposed the models.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.022270	25.3666	0.045507	0.007268
ANN2	0.007119	84.2583	0.045041	0.005076
ANN3	0.000556	98.9609	0.044846	0.004959
ANN4	0.022271	25.3668	0.045719	0.005090
ANN5	0.002029	96.1375	0.048043	0.004929
ANN6	0.006445	87.1508	0.046107	0.004966
ANN7	0.001254	97.6196	0.055075	0.004995
ANN8	0.001491	97.1486	0.046152	0.005100
ANN9	0.000361	99.3431	0.046022	0.005064
ANN10	0.000334	99.3975	0.045729	0.004928
ANFIS Triangular	0.026962	97.4473	20.0848	0.318664
ANFIS Gaussian	0.032781	96.1649	17.0584	0.303119
ANFIS C-Means	0.046422	95.0639	2656.90	0.353463
ANFIS Automatic C-Means	0.050933	90.6639	2363.18	0.357628

Table 7: Results with the Age as input, where IOA is a % and time is in seconds.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.000948	97.6985	0.056777	0.007525
ANN2	0.000489	98.7941	0.057557	0.006808
ANN3	0.000227	99.4466	0.068725	0.006844
ANN4	0.005077	88.2982	0.067912	0.007234
ANN5	0.000879	97.8422	0.058295	0.007051
ANN6	0.000881	97.8351	0.096128	0.006742
ANN7	0.002648	94.3531	0.059993	0.006796
ANN8	0.000484	98.8061	0.065860	0.006761
ANN9	0.000868	97.8568	0.058622	0.006801
ANN10	0.000231	99.4381	0.05965	0.006818
ANFIS Triangular	0.025215	97.8171	100.079	0.313227
ANFIS Gaussian	0.029257	97.2315	94.8045	0.325650
ANFIS C-Means	0.043240	96.0684	5162.69	0.484177
ANFIS Automatic C-Means	0.040733	97.0162	5083.26	0.471273

Table 8: Results with the Age and Material as inputs, where IOA is a % and time is in seconds.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.000879	96.7430	0.056169	0.006140
ANN2	0.000503	98.3707	0.056124	0.004138
ANN3	0.000492	98.4259	0.062693	0.003811
ANN4	0.013809	53.4339	0.066603	0.003864
ANN5	0.000846	96.8935	0.063138	0.004034
ANN6	0.000685	97.5648	0.055889	0.004336
ANN7	0.010979	61.5582	0.056902	0.004323
ANN8	0.000794	97.1894	0.058657	0.004078
ANN9	0.005725	83.1279	0.060453	0.004178
ANN10	0.000536	98.2564	0.063682	0.004573
ANFIS Triangular	0.020216	98.6128	174.807	0.284971
ANFIS Gaussian	0.037704	96.9997	182.483	0.303509
ANFIS C-Means	0.031788	97.7709	7156.54	0.517126
ANFIS Automatic C-Means	0.041262	96.9412	7391.76	0.473339

Table 9: Results with the Age, Material and Diameter as inputs, where IOA is a % and time is in seconds.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.007728	57.8287	0.061624	0.006191
ANN2	0.007765	64.5952	0.055594	0.004828
ANN3	0.000754	90.4248	0.055664	0.004837
ANN4	0.011087	45.0625	0.062095	0.004886
ANN5	0.008018	63.7053	0.055764	0.004695
ANN6	0.000238	97.0912	0.055457	0.005275
ANN7	0.000192	97.8387	0.060256	0.005053
ANN8	0.000140	98.4119	0.064390	0.004991
ANN9	0.000343	95.5579	0.063574	0.005011
ANN10	0.000352	96.1107	0.074729	0.006244
ANFIS Triangular	0.016471	99.3117	431.691	0.318273
ANFIS Gaussian	0.017533	99.1559	375.323	0.361595
ANFIS C-Means	0.020821	98.8912	9676.16	0.535128
ANFIS Automatic C-Means	0.026131	98.3918	8326.25	0.544123

Table 10: Results with the Age, Material, Diameter and Lenght as inputs, where IOA is a % and time is in seconds.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.009437	50.7803	0.060035	0.003979
ANN2	0.010669	54.1496	0.061132	0.003809
ANN3	0.000629	96.5163	0.062686	0.003839
ANN4	0.000563	96.6219	0.064580	0.003673
ANN5	0.000389	97.7043	0.059311	0.004097
ANN6	0.000280	98.3792	0.059158	0.003996
ANN7	0.000377	97.9151	0.065138	0.003873
ANN8	0.000443	97.4038	0.057411	0.003791
ANN9	0.001324	94.1536	0.058871	0.003832
ANN10	0.000912	95.2116	0.110003	0.007252
ANFIS Triangular	0.013181	99.5023	1447.50	0.367417
ANFIS Gaussian	0.012869	99.5338	1486.49	0.393353
ANFIS C-Means	0.020847	98.8881	11467.3	0.555637
ANFIS Automatic C-Means	0.025961	98.4106	12111.5	0.616218

Table 11: Results with the Age, Material, Diameter, Lenght and Usage as inputs, where IOA is a % and time is in seconds.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.027173	37.9382	0.071886	0.007305
ANN2	0.000311	99.6716	0.069758	0.005272
ANN3	0.001106	98.7357	0.061646	0.004722
ANN4	0.020997	60.3282	0.061561	0.004962
ANN5	0.025561	26.4290	0.061524	0.004867
ANN6	0.027449	22.1365	0.063651	0.004771
ANN7	0.023863	66.3046	0.063607	0.004640
ANN8	0.000328	99.6485	0.063289	0.004779
ANN9	0.015454	67.4829	0.061862	0.004851
ANN10	0.000766	99.1651	0.230214	0.015128
ANFIS Triangular	0.065396	81.8607	5138.60	0.434976
ANFIS Gaussian	0.009508	99.7296	4977.63	0.342269
ANFIS C-Means	0.018126	99.1363	15179.1	0.670773
ANFIS Automatic C-Means	0.026339	98.3488	14624.5	0.647034

Table 12: Results with the Age, Material, Diameter, Lenght, Usage and Pressure as inputs, where IOA is a % and time is in seconds.

Model	RMSE	IOA	Training time	Execution time
ANN1	0.020186	38.0299	0.139229	0.008379
ANN2	0.001188	98.3106	0.016790	0.004723
ANN3	0.000798	98.7815	0.023516	0.004447
ANN4	0.012775	65.3788	0.094261	0.004711
ANN5	0.001801	97.5065	0.018295	0.004605
ANN6	0.001729	96.9018	0.020327	0.004589
ANN7	0.002477	96.0997	0.018114	0.004725
ANN8	0.000141	99.7748	0.012913	0.004784
ANN9	0.000327	99.5114	0.012854	0.004892
ANN10	0.009376	86.9213	0.011847	0.033613
ANFIS Triangular	0.010959	99.6534	15533.5	0.493754
ANFIS Gaussian	0.006335	99.8805	16810.8	0.632894
ANFIS C-Means	0.017737	99.1623	20287.7	0.792700
ANFIS Automatic C-Means	0.040899	96.3783	21003.6	0.806995

Table 13: Results with the Age, Material, Diameter, Lenght, Usage, Pressure and Temperature as inputs, where IOA is a % and time is in seconds.

5.1 C-Means algorithm

Deeper discussion of the results is done in the conclusions. However, in this section the best C-Means model is presented. There are several criterions to decide which is the best model. In our case, we have decided to use the model with highest IOA. Therefore, the best model is when we work with 7 inputs without using the autonomous cluster initialization.

Model	RMSE	IOA	Training time	Execution time
ANFIS C-Means 7 inputs	0.017737	99.1623	20287.7	0.792700

Table 14: C-Means model with 7 inputs without using autonomous cluster initialization, where IOA is a % and time is in seconds.

This model achieves a very good accuracy. It is not the model with lower RMSE, neither the fastest one. However, it is able to predict te *failure rate* with less than a second and, considering our problem, this is a good time. An other interesting fact is that this model achieves an accuracy greater than 95% even if we work with one input (Age), see table 7. That means that if we have a system were we can only measure the age, we can already get trustworthy predictions. We do not have to forget the problem we are tackling. Of course, we want the maximum accuracy with the lowest execution time. However, in a real implementation we do not need the highest accuracy to get proper results. Moreover, one of the most important factors is the robustness of the method and ANFIS C-Means model is the most robust one. One the one side, because this model improves if we add more inputs. So, it is able to accept new information. On the other side, because it is not subject to human factor at designing. It has no sensible free parameters and that protects the model against possible human errors. The self-implementation of *membership functions* throught clusters reduce the dimensionality of the problem and delates the need of a previous study of the data. In addition, in ANFIS method, and as far as we are concerned, the implementation of *membership functions* is quite an heuristic procedure because there is not an exact methodology to define the range, the number and the type of the functions. That uncertainty takes more relevance ones we notice that to define this parameters is a sensible part of the method and that it is essential for a convergence in the results.

In the Figure 5, an interesting plot is shown. There we can see a comparison between the real evolution of *failure rate* and the predicted evolution.

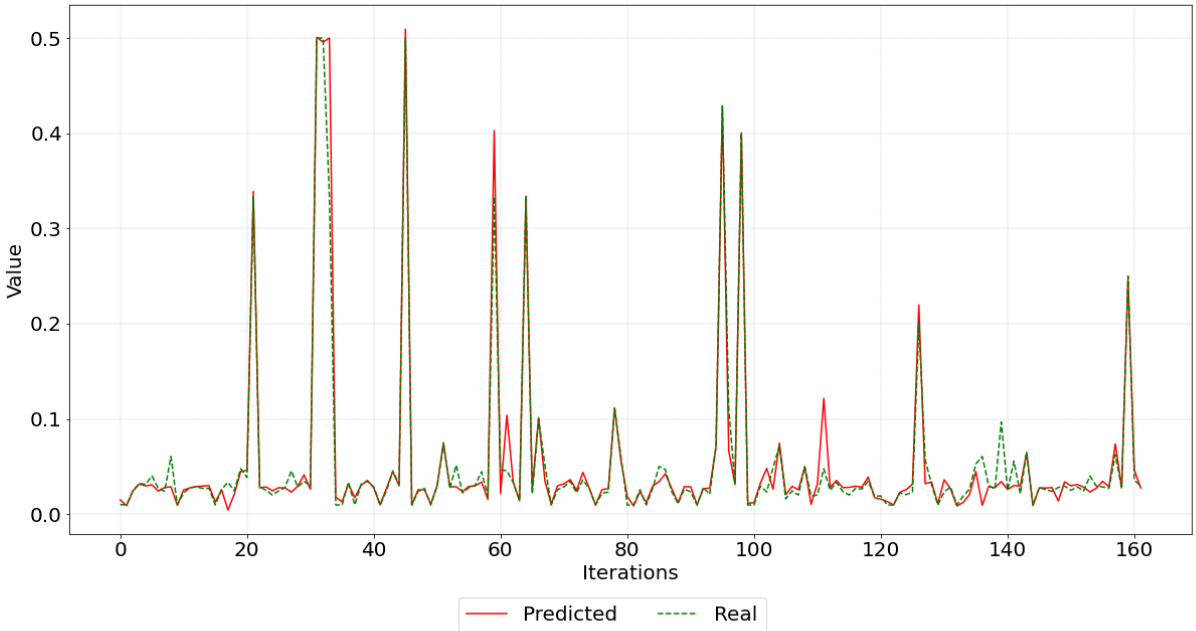


Figure 5: Comparison between the real evolution of *failure rate* and the predicted evolution using C-Means algorithm with 7 inputs.

6 Conclusions

As a conclusions, in this report we have tackled two problems. Talking about the input analysis, we have demonstrated that there is no need of using to much parameters to achieve good results. If we look at table 7, using only the Age as input we can see an excellent performance of Artificial Neural Network 9 and 10, and also very good results of other ANNs and ANFIS models. Going throught the other results, we can see that the results gets better generally. However, the imporvement is not very high. That phenomena supports our initial theory of the most important parameters, see section 4.

It can be logical that the more inputs we use, the better the predictions are. However, as we explain in the introduction, not every network has the same possibilities. So, this report shows which methods are better in function of the measureable parameters.

Talking about the methods, we can see that Artificial Neural Networks are the fastest ones. In practice, it is not a bigger advantage because we don't need such velocity and, moreover, the ANFIS models have quite low training and execution cost. In general, ANN are better than ANFIS with little inputs. But that changes onces the numer of parameters increase. In addition, not all of the ANN tested models are as good, and some shows good performances with a certain quantity of inputs, but then shows bad performances with different inputs, for example ANN5. That phenomena does not appear in ANFIS models. So, we can conclude that ANFIS models are more robust and, therefore, are more interesting as you have more control over the model and they tend to improve if more information is added to the network.

Focusing on the C-Means approch, we can conclude it is an interesting method for distribution systems management. It is true that this method never bits ANFIS original algorithm, however it is not too far from its results. Apart from the fact that C-Means apporch has other benefits over ANFIS, see section 3. It is important to know that to achieve the correct *membership functions* used, we spent some time testing different options, meanwhile in C-Means it was automatic. In addition, *membership functions* are modified during the training, while clusters are not. That means that C-Means model has not achieved all its potential.

About the autonomous way of fixing the number of clusters, althought achieving good results, it has always worse performance than the C-Means first approach. We have to think that, as we have just explained, clusters are not modified during the training and that can be an interesting imporvement. Moreover, to decide the number of clusters, C-Means algorithm is trained apart. This procedure has its own hyperparameters and that means that it can be tuned and maybe get better results. All this explained can be part of a future work.

Finally, we conclude that C-Means approach is the most interesting method. It strenght does not come from the results, althought it has always one of the best performances and it is always close to the benchmark methods. The best characteristic of this method is its easy implementation and its robustness against human factor. In ANN and, above all, in ANFIS, the human factor is very rellevant to define correct parameters to obtain good results. In our research, we spend a lot of time fitting parameters. Meanwhile, C-Means approach has worked properly since its first implementation. That makes this model able to be implemented for anyone because there is no need of having previous knowledge of the data and having previous knowledge in parameters such as *membership functions*.

A Membership Functions

In this appendix, the *membership fucntions* are exposed in a graphical way.

In this section only triangular *membership fucntions* are exposed. En each plot we can see the initial *membership fucntions* over the training dataset histogram. The maximum value of a *membership fucntions* is 1, however, here we have oversized the functions to improve the visibility. And also, the *membership fucntions* after training is shown.

Gaussian *membership fucntions* are not exposed. The initial ones are similar than triangular initialization. The range of the distribution is the same, only the shape changes. For this reason, they are not shown. The main goal of this appendix is to show a thumb rule to define the *membership fucntions*. The main factor is the range of each membership.

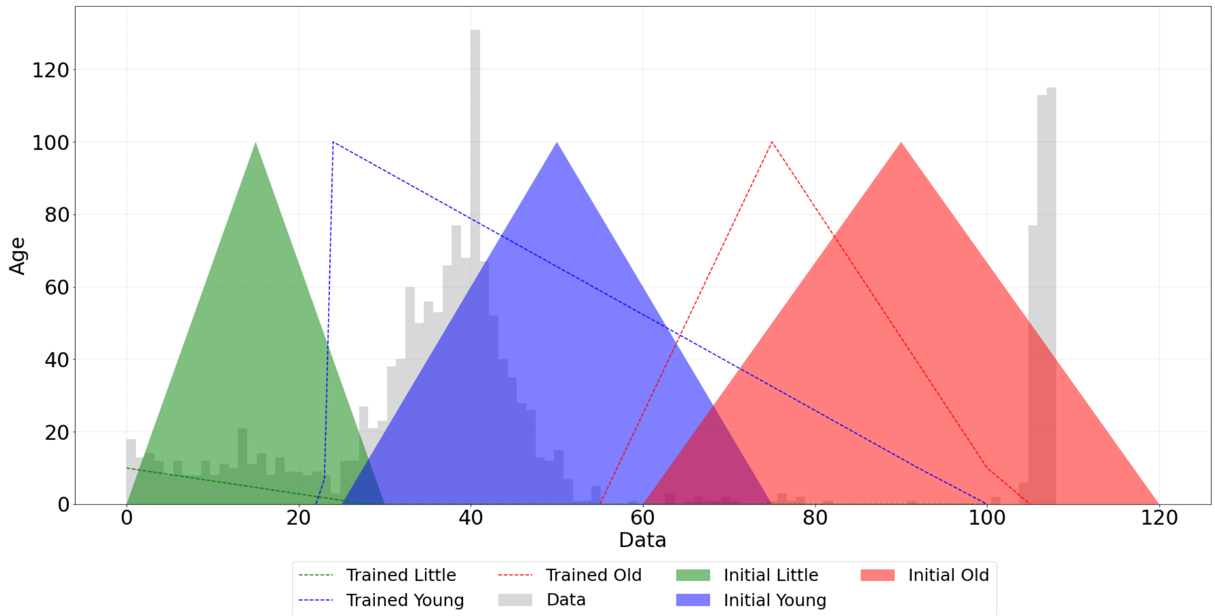


Figure 6: Age *membership functions.*

Figure 7: Material *membership functions.*

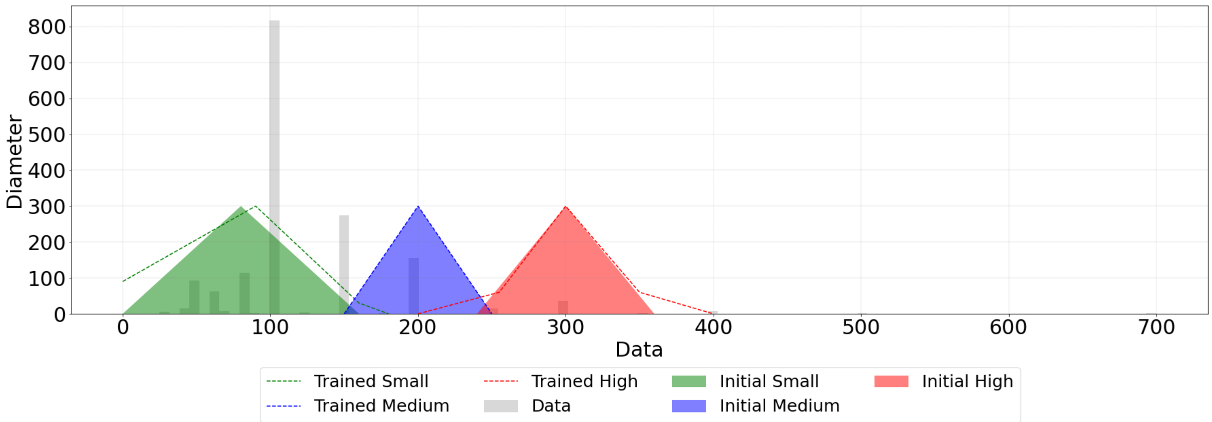


Figure 8: Diameter *membership functions.*

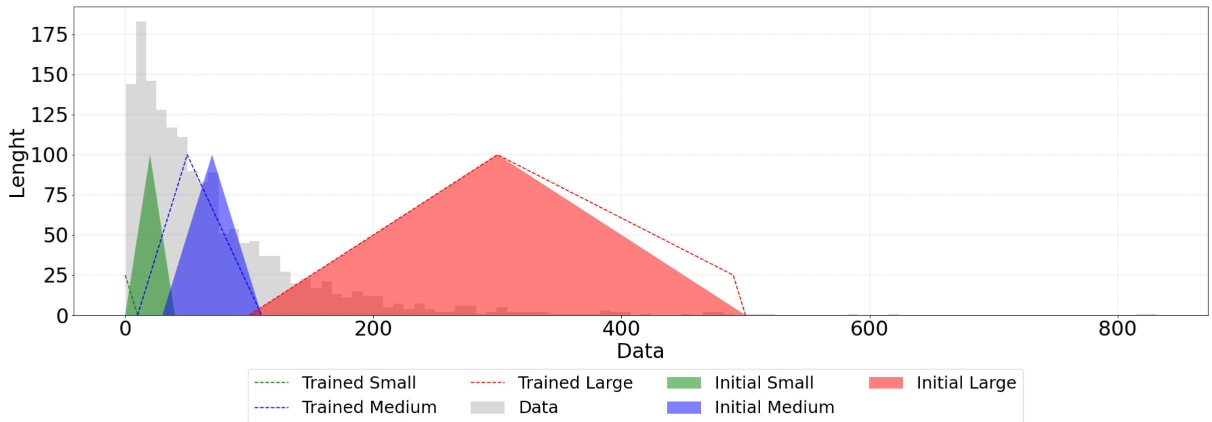


Figure 9: Lenght *membership functions.*

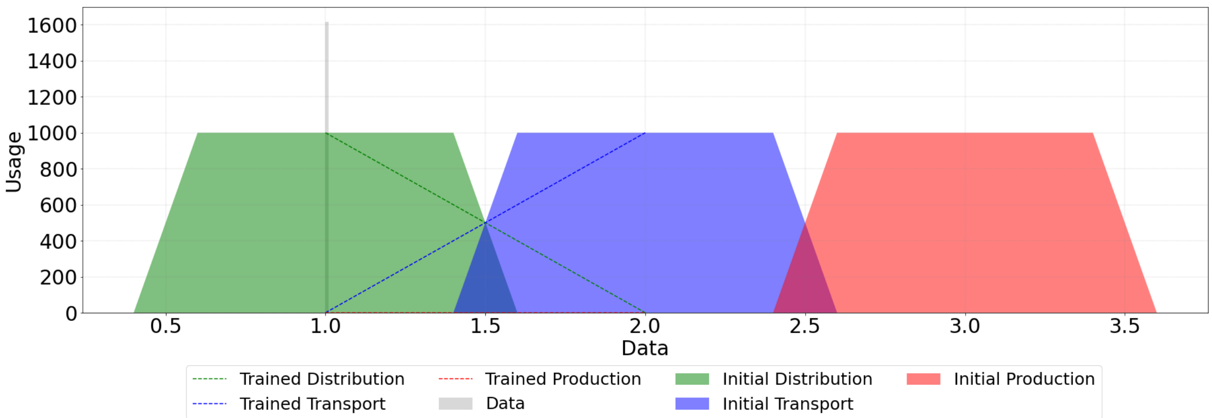


Figure 10: Usage *membership functions.*

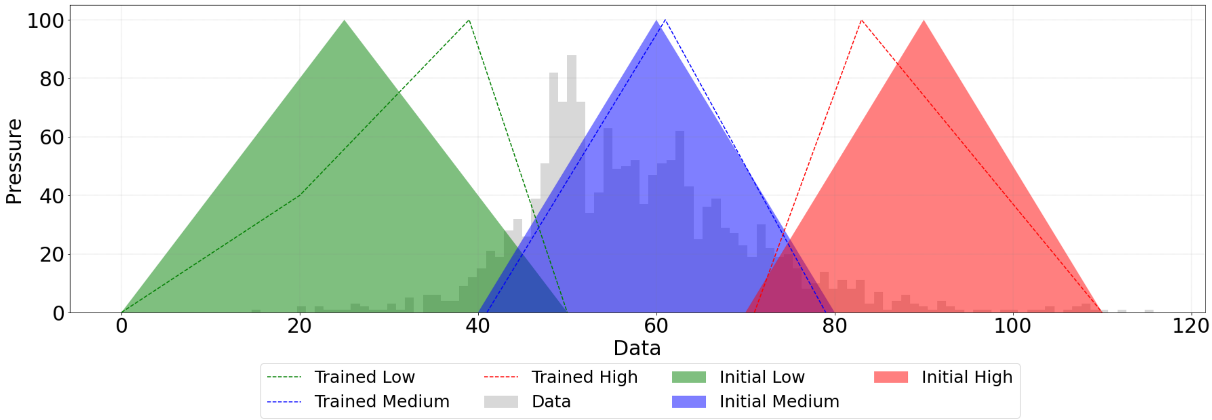


Figure 11: Pressure *membership functions.*

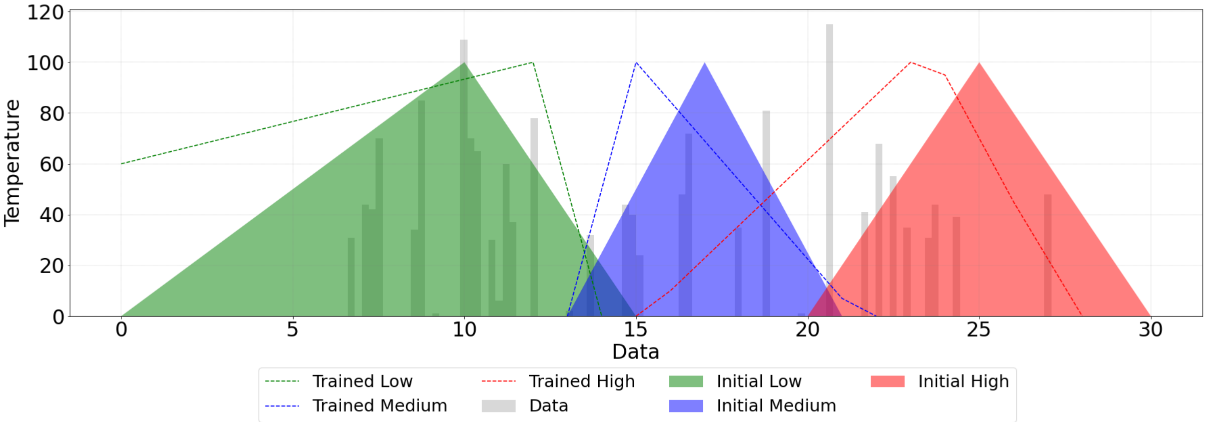


Figure 12: Temperature *membership functions.*

B Clusters

In this appendix, the clusters tested for each input are shown. We can see different number of clusters for each input and the *fitting coeficient* (FPC) for each test.

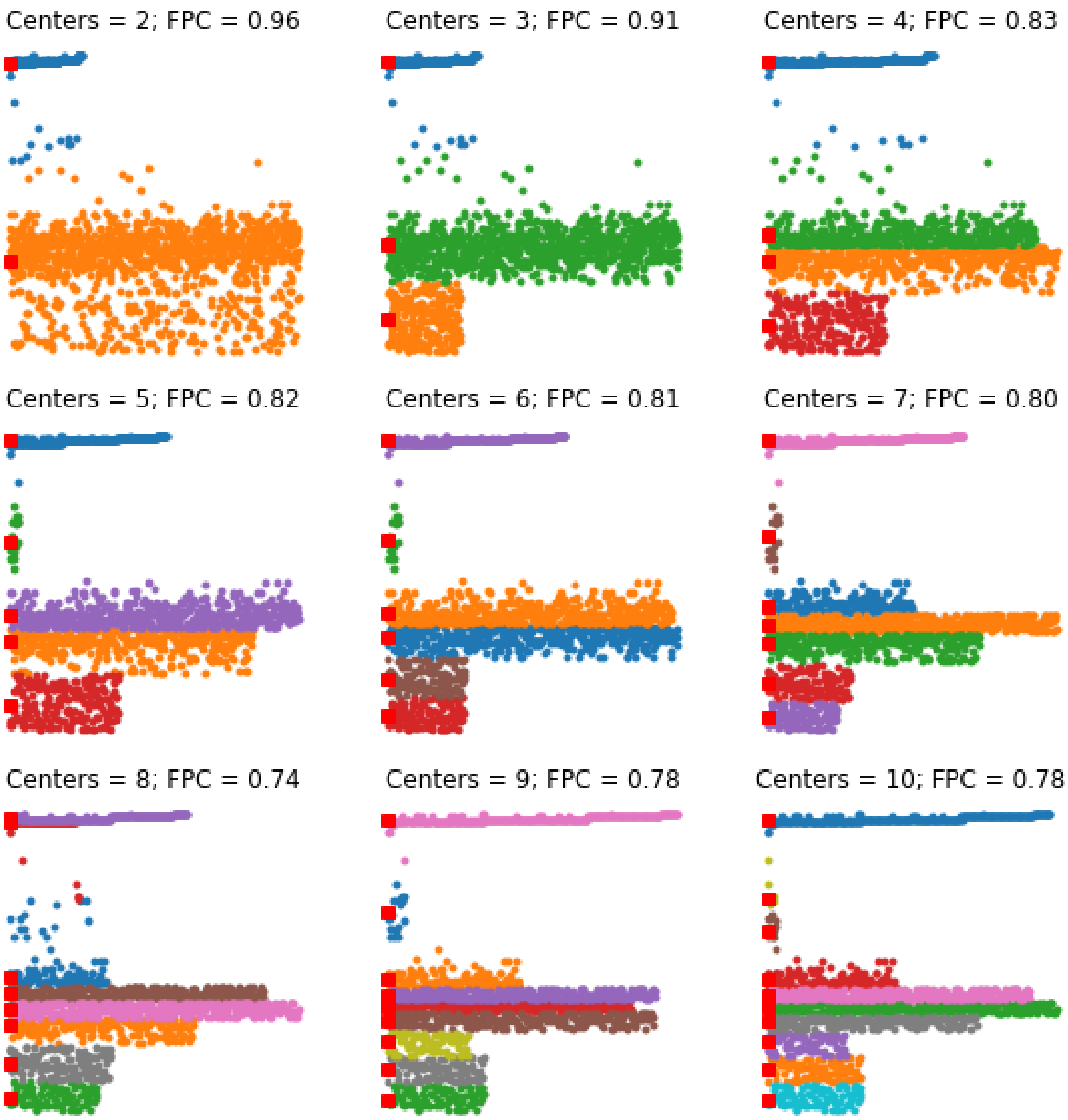


Figure 13: Age clusters.

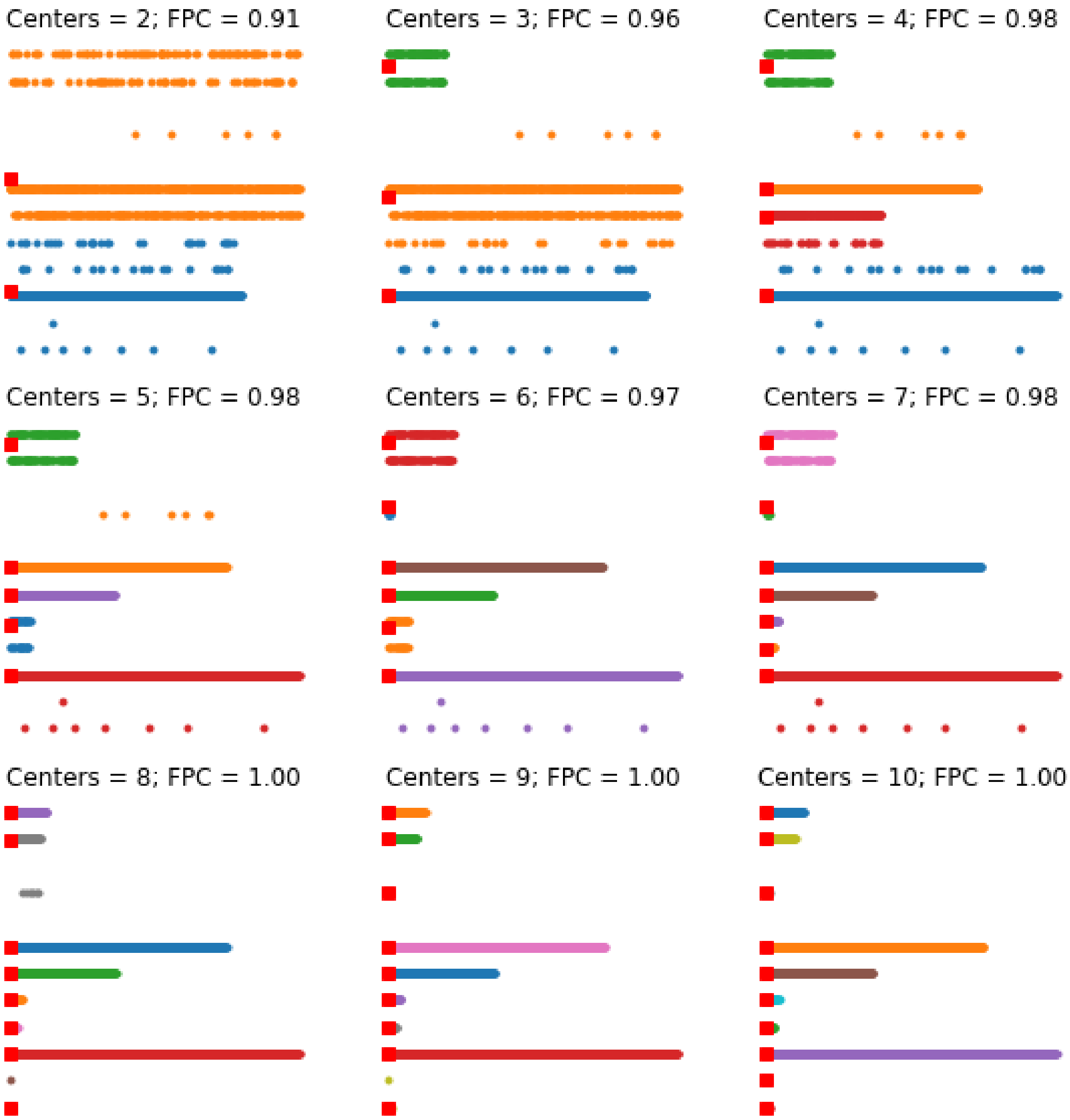


Figure 14: Material clusters.

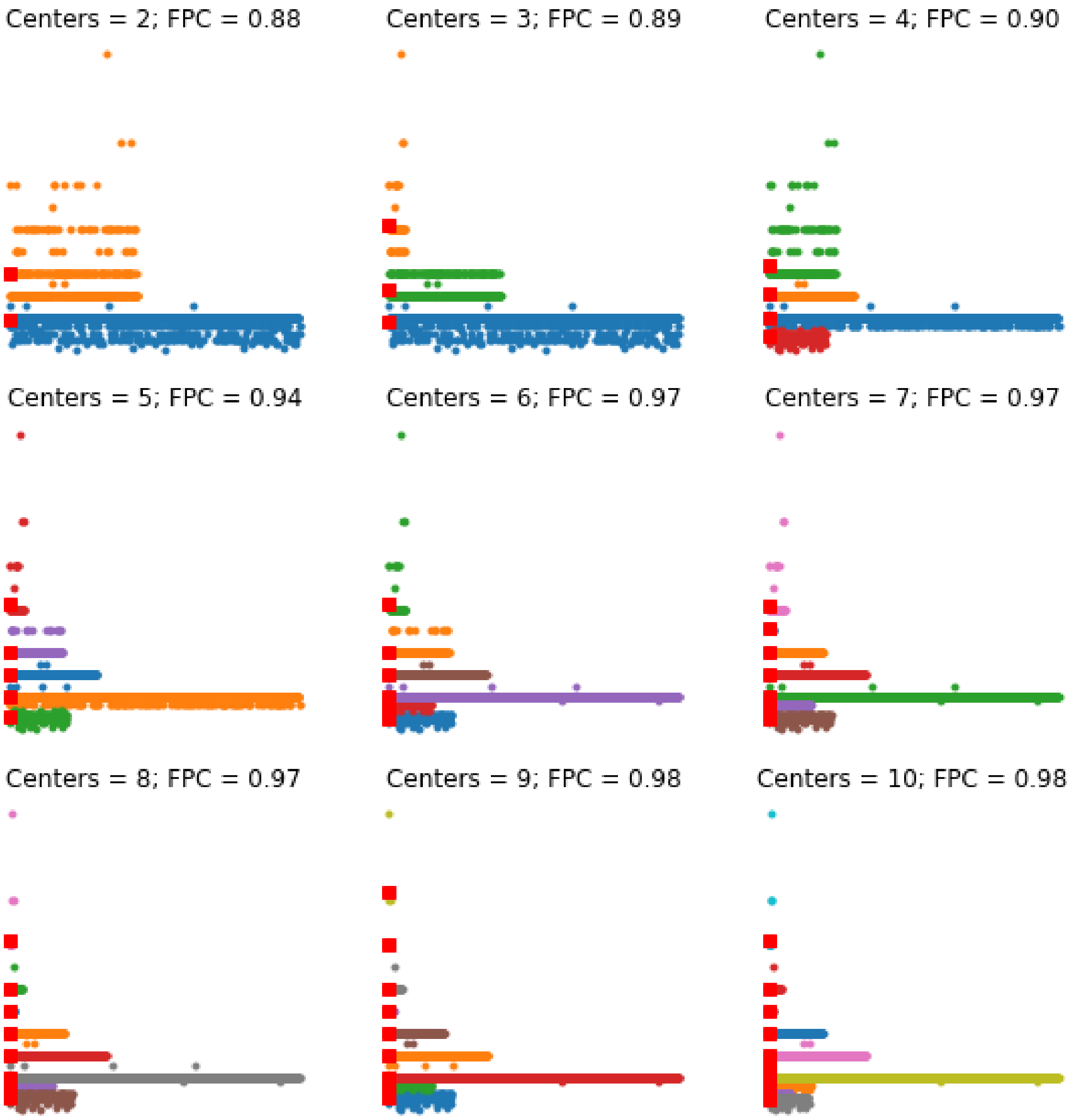


Figure 15: Diameter clusters.

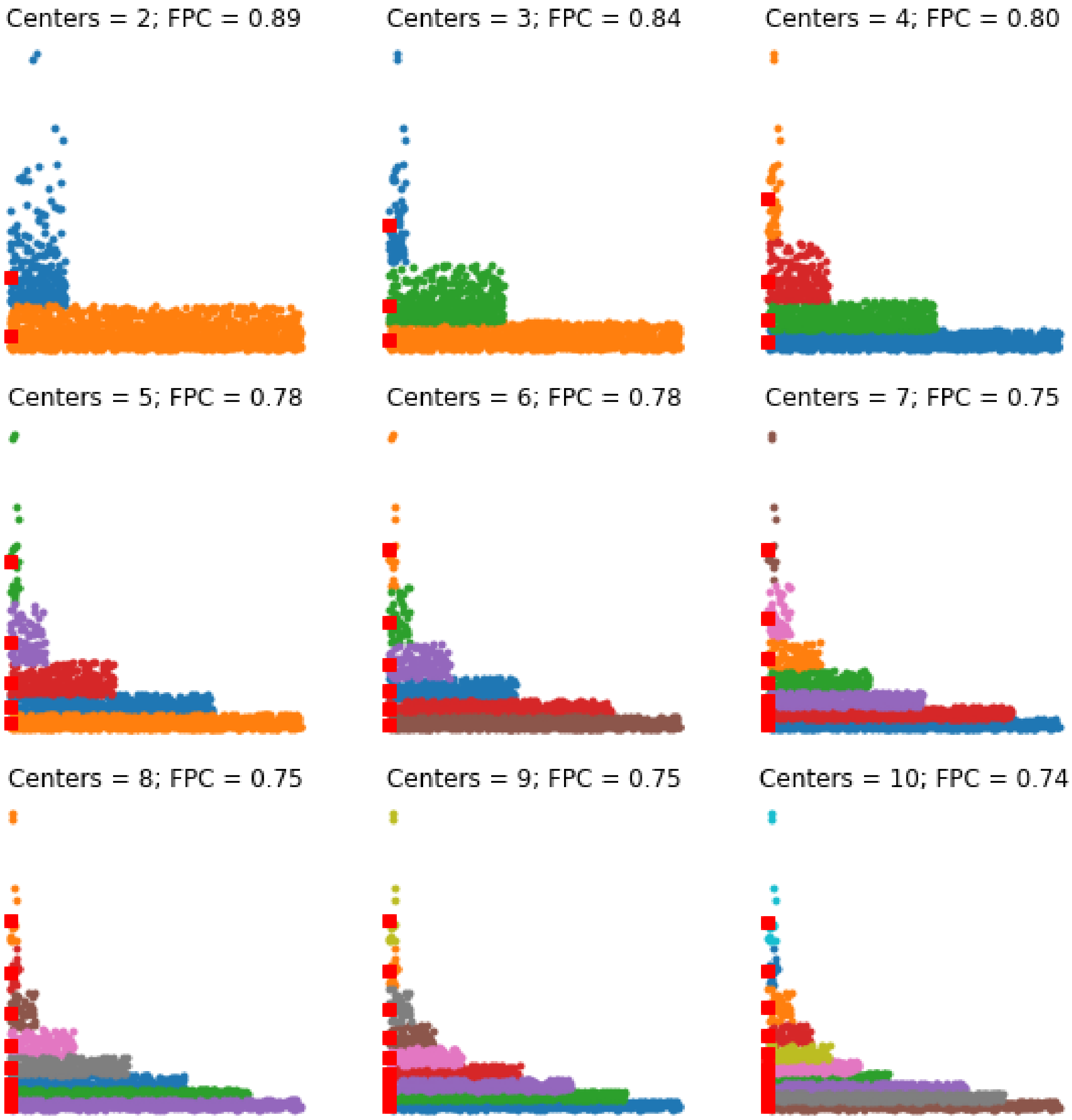


Figure 16: Lenght clusters.

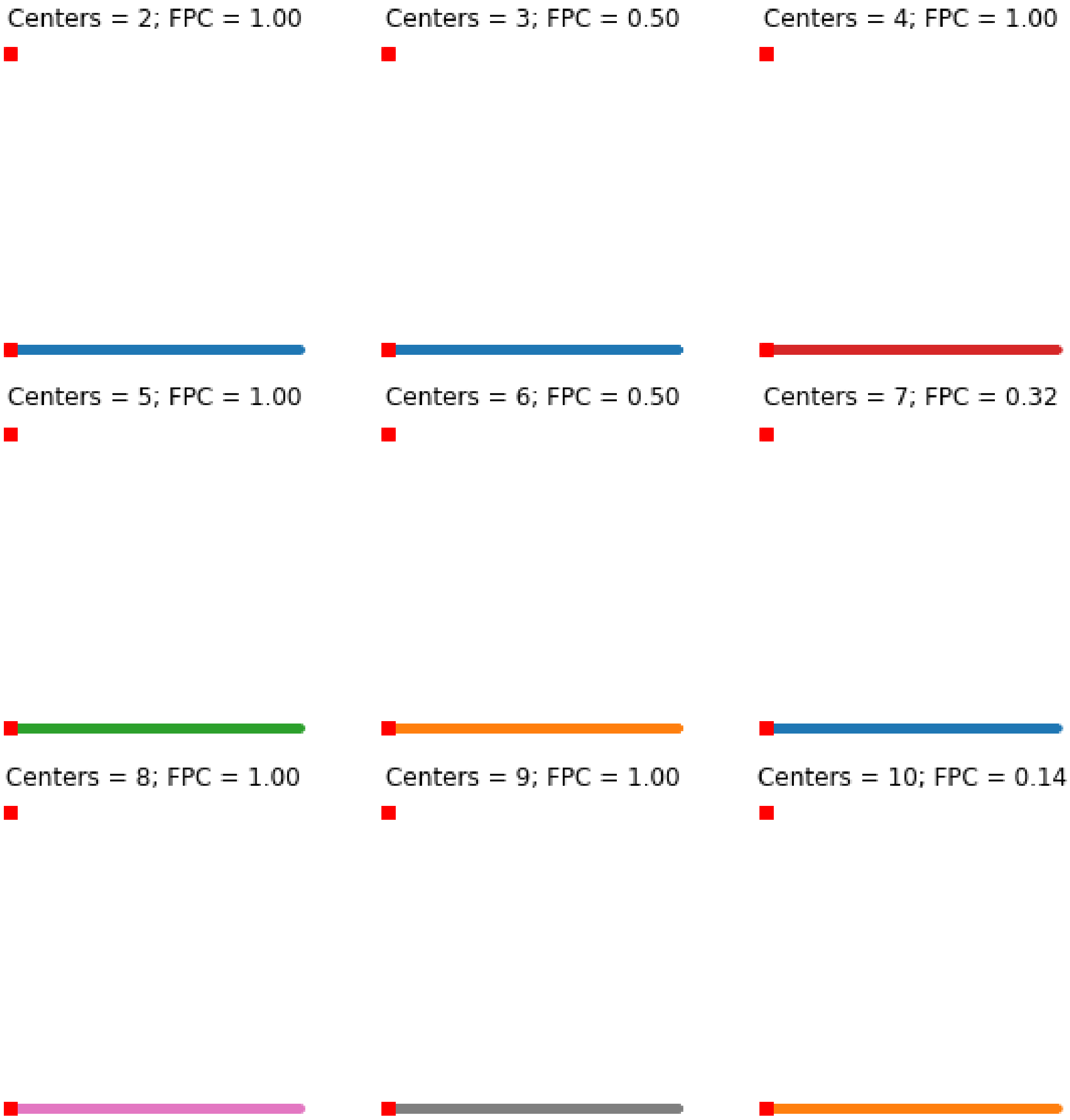


Figure 17: Usage clusters.

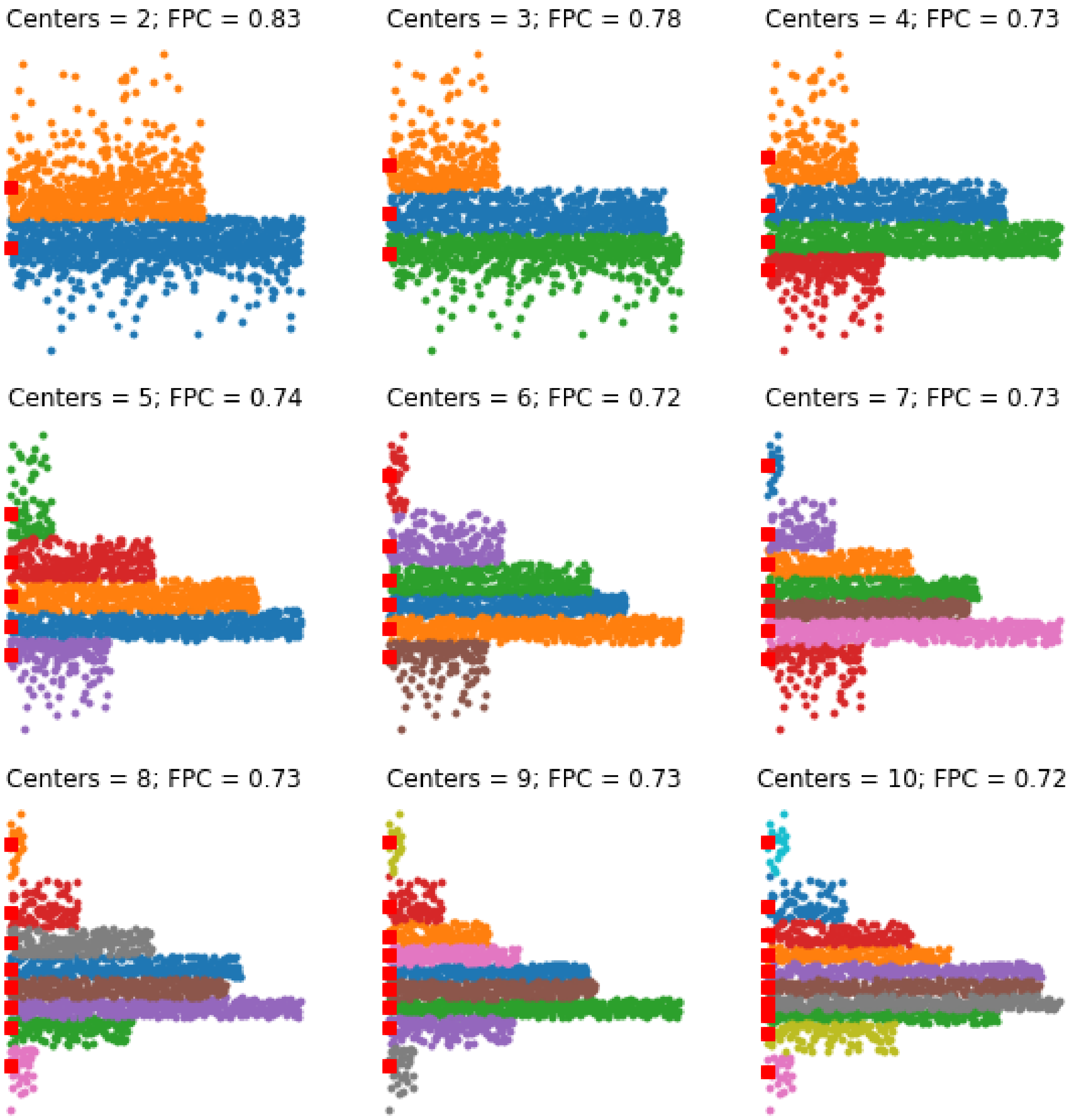


Figure 18: Pressure clusters.

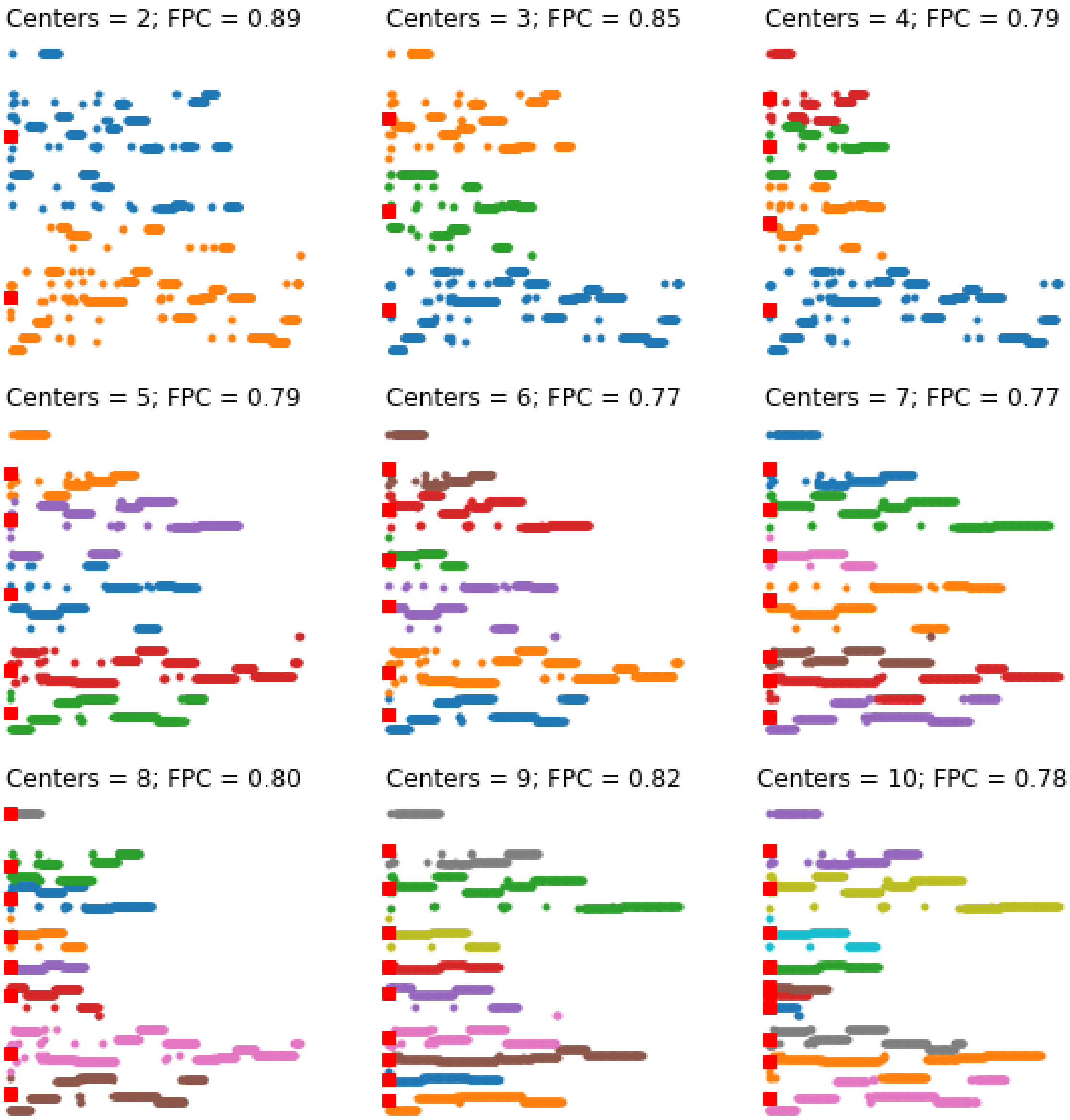


Figure 19: Temperature clusters.

References

[1] R. Al-Hmouz Ahmed Al-Hmouz, Jun Shen and Jun Yan. Modeling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning. In *Third quarter*, pages 226–237. IEEE Transactions on Learning Technologies, 2012.

[2] Symeon Christodoulou and Alexandra Deligianni. A neurofuzzy decision framework for the management of water distribution networks. In Springer Science, editor, *Water Resour Manage*, pages 139–156, 2009.

[3] H. Demuth and M. Beale. Neural Network Toolbox User's Guide for Use with MATLAB. 2002.

[4] Dr. Jerry L. Hintze. Chapter 448: Fuzzy Clustering. In *NCSS User's Guide*, pages 1–9. NCSS Statistical Data, 2007.

[5] R. Farmani M. Tabesh, J. Soltani and D. Savic. Assessing pipe failure rate and machanical reliability of water distribution networks using data driven modelling. Journal of Hydroinformatics, 2009.

[6] James F. Power. An ANFIS framework for PyTorch. IEEE International Conference on Fuzzy Systems, 2019.

Acknowledgements

The authors thank to Spanish State Research Agency through Maria de Maeztu Seal of Excellence to IRI (MDM-2016-0656), and Spanish Ministry of Science and Technology through Project ECOCIS (Ref. DPI2013-48243-C2-1-R).

IRI reports

This report is in the series of IRI technical reports.
All IRI technical reports are available for download at the IRI website
`http://www.iri.upc.edu`.