NT4XAI: a Framework Exploiting Network Theory to Support XAI on Classifiers

(Discussion Paper)

Gianluca Bonifazi[1,†], Francesco Cauteruccio[1,†], Enrico Corradini[1,†], Michele Marchetti[1,†], Giorgio Terracina[2,†], Domenico Ursino[1,*,†] and Luca Virgili[1,†]

[1]*DII, Polytechnic University of Marche*

[2]*DEMACS, University of Calabria*

Abstract

Explainable AI (XAI, for short) aims to explain the behavior of closed AI systems that act as black-boxes (like many Machine Learning and Deep Learning systems). In this paper, we propose NT4XAI, a model-agnostic framework carrying out explainable AI on classifiers. NT4XAI is based on network theory and, consequently, is able to take advantage of the enormous amount of results found over the years by researchers in this area. Here, we describe both the data model and the approach used by NT4XAI to achieve its goals. Furthermore, we contextualize our framework within the existing XAI research scenarios. Finally, we illustrate some tests we carried out to assess its adequacy in performing the tasks for which it was designed.

Keywords

Explainable Artificial Intelligence, Model-Agnostic XAI Systems, Graph Theory, Feature Relevance, Feature Dyscrasia, Sensitivity Analysis

1. Introduction

Explainable AI (XAI, for short) aims to identify transparent and interpretable explanations to the decisions and actions of black-box AI systems [1, 2, 3, 4, 5]. It aims to know, at least partially, how a black-box AI model acts and to use that information for improving its performance, increasing confidence in it, as well as the level of acceptance of the knowledge it returns [6, 7]. With the pervasive diffusion of Deep Learning (DL, for short), the number of black-box models has grown tremendously and, in hand, interest in XAI has increased. One of the most challenging issues in XAI concerns the study and development of "model-agnostic" XAI approaches. These are capable of interpreting and explaining the decisions of any black-box system, regardless of

SEBD 2023: 31st Symposium on Advanced Database Systems, July 02–05, 2023, Galzignano Terme, Padua, Italy

[*] Corresponding author.

[†] These authors contributed equally.

g.bonifazi@univpm.it (G. Bonifazi); f.cauteruccio@univpm.it (F. Cauteruccio); e.corradini@pm.univpm.it (E. Corradini); m.marchetti@pm.univpm.it (M. Marchetti); terracina@mat.unical.it (G. Terracina); d.ursino@univpm.it (D. Ursino); luca.virgili@univpm.it (L. Virgili)

0000-0002-1947-8667 (G. Bonifazi); 0000-0001-8400-1083 (F. Cauteruccio); 0000-0002-1140-4209 (E. Corradini); 0000-0003-3692-3600 (M. Marchetti); 0000-0002-3090-7223 (G. Terracina); 0000-0003-1360-8499 (D. Ursino); 0000-0003-1509-783X (L. Virgili)

CEUR Workshop Proceedings (CEUR-WS.org)

the model on which it is based. Therefore, they are extremely general, and investing in them provides a considerable return since they can be applied to explain very varied AI models. The downside is that these systems are very difficult to design because they must feature a high abstraction level with respect to the black-box models they want to explain.

In this paper, we aim to make a contribution in this setting by proposing NT4XAI (Network Theory for Explainable AI), a model-agnostic framework for explainability of classifiers. NT4XAI operates on a classifier model whose behavior is unknown. The classifier receives as input a set of instances, all characterized by the same set of features, and assigns a class to each of them. As its name indicates, NT4XAI is based on network theory [8]; in fact, it builds and maintains a fully connected network. In it, nodes represent instances, while the direction of the arc between two nodes is an indicator of the confidence level with which the classifier has classified the corresponding instances. Once the network is constructed, NT4XAI computes the "dyscrasia" of each feature for all instances. This measure indicates the effectiveness of a feature in discriminating instances. Starting from the values of dyscrasia thus obtained and the properties of the constructed network, NT4XAI computes the relevance of each feature during the classification process [9, 10, 11, 12, 13]. For this purpose, it uses a version of PageRank [14] specifically defined to address this issue. The knowledge of the most relevant features provides valuable information about the behavior of the black-box classifier, as it has already been shown in the scientific literature on XAI [1, 15, 9, 16, 17]. The choice to use network theory in NT4XAI is motivated by the extreme generality and flexibility characterizing network-based representations. Furthermore, network theory has been intensively studied in the past, in terms of both its theoretical aspects and its possible applications [18, 19, 20]. Therefore, NT4XAI can benefit from the wide range of past results in this research field adapting them to address the issue for which it was thought.

The outline of this paper is as follows: In Section 2, we describe NT4XAI in detail. In Section 3, we present some experiments we performed to evaluate it. Finally, in Section 4, we draw some conclusions and define some possible future developments of this research.

2. Description of NT4XAI

In this section, we illustrate the model underlying NT4XAI and the behavior of the latter. Let $\mathcal{I} = \{I_1, I_2, \cdots, I_l\}$ be a set of instances to be classified and let $\mathcal{C} = \{C_1, C_2, \cdots, C_m\}$ be the set of possible classes. Let $\mathcal{F} = \{F_1, F_2, \cdots, F_n\}$ be the set of features characterizing the instances of $\mathcal{I}$. Accordingly, an instance $I_i \in \mathcal{I}$ can be represented by the set $\mathcal{F}_i = \{F_{1_i}, F_{2_i}, \cdots, F_{n_i}\}$ of the values of its features. Here, $F_{k_i} \in \mathcal{F}_i$ indicates the value of the feature F_k in I_i. Each feature F_k can be numeric, categorical or textual.

Suppose we have a classifier model $\mathcal{M}$ that was already trained. For each instance $I_i \in \mathcal{I}$, $\mathcal{M}$ assigns a class of $\mathcal{C}$ to it with a confidence level c_i[1] belonging to the real interval $[0, 1]$; the higher c_i, the more confident $\mathcal{M}$ in classifying I_i. The behavior of $\mathcal{M}$ can be represented by a network $\mathcal{N} = \langle N, A \rangle$. The nodes of $\mathcal{N}$ represent the instances of $\mathcal{I}$, while its arcs indicate the confidence level of $\mathcal{M}$ in classifying the instances associated with the corresponding nodes. Formally speaking, there is a node $n_i \in N$ for each instance $I_i \in \mathcal{I}$. Since a biunivocal correspondence

[1] Our classifier model assumes that each instance can be assigned to exactly one class.

exists between a node n_i and an instance I_i, in the following we will use the terms "node" and "instance", as well as the symbols n_i and I_i, interchangeably. There is an arc of A for each pair of nodes (n_i, n_h) of $\mathcal{N}$. It is directed from n_i to n_h if $c_i < c_h$; otherwise, if $c_h < c_i$, it is directed from n_h to n_i. Finally, if $c_i = c_h$, its direction is set randomly.

Having defined the model underlying NT4XAI, let us now see how our framework defines the dyscrasia $\delta(F_{k_i}, F_{k_h})$ between the values F_{k_i} and F_{k_h} of the feature F_k for the instances I_i and I_h. The concept of dyscrasia is intended to capture the "disharmony" in the role that two occurrences F_{k_i} and F_{k_h} of the same feature F_k played in the classification of two instances I_i and I_h made by $\mathcal{M}$. As we shall see below, the dyscrasia between two occurrences of the same feature will play a key role in calculating the relevance of the latter. The reasoning behind the definition of $\delta(F_{k_i}, F_{k_h})$ is as follows: If $\mathcal{M}$ assigned I_i and I_h to the same class, the value of $\delta(F_{k_i}, F_{k_h})$ is the greater the more: *(i)* F_{k_i} and F_{k_h} have dissimilar values, and *(ii)* the confidences c_i and c_h with which $\mathcal{M}$ classified I_i and I_h are low (meaning that there is no significant confidence about the correctness of the actions of $\mathcal{M}$). In contrast, if $\mathcal{M}$ assigned I_i and I_h to different classes, the value of δ is the greater the more: *(i)* F_{k_i} and F_{k_h} have similar values, *(ii)* the value of c_h is high and the one of c_i is low (meaning that the possibility that $\mathcal{M}$ classified I_h correctly and I_i incorrectly is significant).

The dyscrasia $\delta(F_{k_i}, F_{k_h})$ can be formalized as follows:

$$\delta(F_{k_i}, F_{k_h}) = \begin{cases} \varepsilon(n_i) \cdot \varepsilon(n_h) \cdot \lambda(F_{k_i}, F_{k_h}) & \text{if } \mathcal{M} \text{ assigned } I_i \text{ and } I_h \text{ to the same class} \\ \varepsilon(n_i) \cdot \gamma(n_h) \cdot [1 - \lambda(F_{k_i}, F_{k_h})] & \text{otherwise} \end{cases}$$

Here, $\lambda(\cdot, \cdot)$ is a function that receives two values F_{k_i} and F_{k_h} and returns a value in the real interval $[0, 1]$ indicating the dissimilarity degree between F_{k_i} and F_{k_h}. Clearly, the definition of $\lambda(\cdot, \cdot)$ depends on the type of F_k. For example, if F_k is numeric, $\lambda(\cdot, \cdot)$ might return the absolute value of the dissimilarity between F_{k_i} and F_{k_h}, suitably normalized. $\gamma(\cdot)$ is a function that receives a node n_i and returns the confidence c_i of $\mathcal{M}$ in classifying the instance I_i corresponding to n_i. Finally, $\varepsilon(\cdot)$ receives a node n_i and returns the error of $\mathcal{M}$ in classifying I_i. It is defined as $\varepsilon(n_i) = 1 - \gamma(n_i)$.

Having defined the dyscrasia between two occurrences of a feature, we are now able to describe how NT4XAI defines the relevance of a feature during a classification process performed by a (possibly) black-box classifier. Recall that, based on the definition of the model underlying NT4XAI, given a node $n_i \in N$, its incoming (resp., outgoing) arcs start from nodes whose associated instances have been classified with lower (resp., higher) or equal confidence. The two sets can be defined as follows: $N_i^{out} = \{n_h | n_h \in N, n_h \neq n_i, (n_i, n_h) \in A\}$ and $N_i^{in} = \{n_h | n_h \in N, n_h \neq n_i, (n_h, n_i) \in A\}$. Let F_k be the feature whose relevance NT4XAI must determine. In order to carry out this task, NT4XAI must preliminarily determine the relevance of F_{k_i} for each instance $I_i \in \mathcal{I}$. Let n_i be the node corresponding to I_i in N. Based on what we said above, in determining the role of F_k in the classification task, n_i can act as a "guide" for the nodes of N_i^{in}, while it should be "guided" by the nodes of N_i^{out}. One way to formalize this reasoning is to adapt PageRank centrality [14] to this scenario. Proceeding in this way, we have that the relevance $\rho(F_{k_i})$ of F_{k_i} can be defined as:

$$\rho(F_{k_i}) = \frac{1 - d_{k_i}}{|N|} + d_{k_i} \cdot \left(\sum_{n_h \in N_i^{in}} \frac{\rho(F_{k_h})}{|N_h^{out}|} \right)$$

As can be seen from this formula, the relevance of F_{k_i} includes a fixed and a variable component. The former depends on the number of nodes in $\mathcal{N}$. The latter depends on the relevance of the feature occurrences related to the starting nodes of the arcs incoming into n_i. The relevance $\rho(F_{k_h})$ of each of these nodes n_h is weighted by the number of arcs outgoing from n_h. In fact, the greater the number of these arcs, the lower the weight of $\rho(F_{k_h})$. This is justified considering that the number of arcs outgoing from n_h indicates the number of nodes having a higher confidence than n_h.

Unlike the original PageRank formula [14], the damping factor d_{k_i} in the definition of $\rho(F_{k_i})$ has not a constant value, but varies for each node $n_i \in N$ and depends on the characteristics of the latter. In particular, it depends on the number of arcs outgoing from n_i and the dyscrasia between the feature occurrence of each of these nodes and the feature occurrence F_{k_i} of F_k in n_i. More specifically, d_{k_i} can be defined as follows: $d_{k_i} = \sigma \left(\frac{\sum_{n_h \in N_i^{out}} \delta(F_{k_i}, F_{k_h})}{|N_i^{out}|} \right)$.

The rationale for this definition is the following: the value of d_{k_i} depends on the magnitude of the dyscrasia between the occurrence of F_k for n_i and the occurrence of F_k for all the ending nodes of the arcs outgoing from n_i, thus characterized by a higher confidence than the one of n_i. Therefore, there is a positive correlation between the values of the damping factor and those of dyscrasia. Let us now consider the definition of $\rho(F_{k_i})$; in it, if the value of d_{k_i} is high, the weight of the first term in the formula tends to be very low. The second term depends strongly on the number of arcs incoming into n_i. If that number is low (implying that the confidence of $\mathcal{M}$ in the classification of I_i is low) then the relevance of F_{k_i} will be low. This is correct since $\mathcal{M}$ did not show a high confidence in classifying I_i, and F_{k_i} showed a high dyscrasia with the feature occurrences of the nodes whose instances were classified by $\mathcal{M}$ with a higher confidence than I_i. The function $\sigma(\cdot)$ present in the formula of d_{k_i} is the sigmoid function. It varies between 0 and 1 when its argument varies from $-\infty$ to $+\infty$. In particular, if the argument can only be non-negative, as in our case, $\sigma(\cdot)$ varies between 0.5 and 1 and acts as an amplifier of the differences in the values taken on by the argument as it goes along.

Having defined the relevance of a single feature occurrence F_{k_i}, we can define the relevance of a feature F_k as the mean of the relevances of all its occurrences: $\rho(F_k) = \frac{\sum_{n_i \in N} \rho(F_{k_i})}{|N|}$.

3. Experimental campaign

We implemented NT4XAI in Python 3.9 and performed our tests on a 2019 MacBook Pro equipped with 16GB of RAM and 2.6 GHz Intel Core i7 6 core. In addition, we chose multiple classifier models among those most widely used in the literature [11, 21, 22]. Specifically, the classifiers we chose are: *(i)* Naive Bayes (hereafter, NB); *(ii)* SVM with polynomial kernel (hereafter, SVMP); *(iii)* SVM with radial basis function kernel (hereafter, SVMR); *(iv)* Multi-Layer Perceptron (hereafter, MLP); *(v)* Random Forest (hereafter, RF). Naive Bayes is a probabilistic classifier, unlike SVM. Regarding the latter, we considered two kernels. The first, polynomial, considers features and their combinations. The second, radial, separates data using a nonlinear decision-boundary. Multi-Layer Perceptron is a special case of neural network and therefore is a totally black-box model. Finally, Random Forest is an ensemble learning model. In these experiments, we chose classifiers of different types, which exhibit very different behaviors, because we wanted to test the real ability of NT4XAI to be model-agnostic.

During the test campaign, we used the Iris dataset [23] published on the UCI Machine Learning Repository [24]. It consists of 150 instances, 4 features and 3 classes. Specifically, the features are: *(i)* `sepal_length`, representing the sepal length in centimeters; its values range in the real interval [4.3, 7.9]; *(ii)* `sepal_width`, denoting the sepal width in centimeters; its values range in the real interval [2.0, 4.4]; *(iii)* `petal_length`, indicating the petal length in centimeters; its values range in the real interval [1.0, 6.9]; *(iv)* `petal_width`, representing the petal width in centimeters; its values range in the real interval [0.1, 2.5]. Although all features are numerical, their values are very heterogeneous. To homogenize them, we performed a normalization task by using a min-max scaler [25]. It operates as follows: given the value F'_{k_i} of a feature, whose maximum and minimum values are $F'_{k_{max}}$ and $F'{}_{k_{min}}$, the scaler obtains the normalized value F_{k_i} of F'_{k_i} as: $F_{k_i} = \frac{F'_{k_i} - F'_{k_{min}}}{F'_{k_{max}} - F'_{k_{min}}}$. F_{k_i} belongs to the real interval $[0, 1]$. Now, since all feature occurrences are normalized between 0 and 1, we chose as the dissimilarity function $\lambda(F_{k_i}, F_{k_h})$ between two feature occurrences F_{k_i} and F_{k_h} the absolute value of their difference: $\lambda(F_{k_i}, F_{k_h}) = |F_{k_i} - F_{k_h}|$.

The first test we carried out was the computation of the accuracy of classifiers. In Table 1, we report the results obtained. As can be seen from this table, the values are very high. This allows us to conclude that all classifiers considered can guarantee high confidence values and, therefore, can be employed in the next tests.

Model	*Accuracy*
Naive Bayes	0.93
SVM with polynomial kernel	0.98
SVM with radial basis function kernel	0.96
Multi-Layer Perceptron	0.93
Random Forest algorithm	0.96

Table 1
Classifier accuracy with the Iris dataset

Before proceeding further, a premise is necessary. The main objective of our analysis is to check whether there are any features that have a higher relevance value than others. Therefore, if all classifiers showed no significant differences between the relevance values of the various features, we could reasonably conclude that the latter all have the same relevance. In contrast, if some or all of the classifiers show significantly different relevance values for the various features and agree in indicating which of them are the most relevant, we could reasonably conclude that the relevance values of the features are significantly different and could determine which features are most relevant. In this case, the best classifiers would be those that can best show the differences in the relevances among the various features. Having this in mind, we can proceed with the next tests. The first of them aims to compute the value of the damping factor for the various features and classifiers. Figure 1 reports the corresponding distributions represented by means of boxplots.

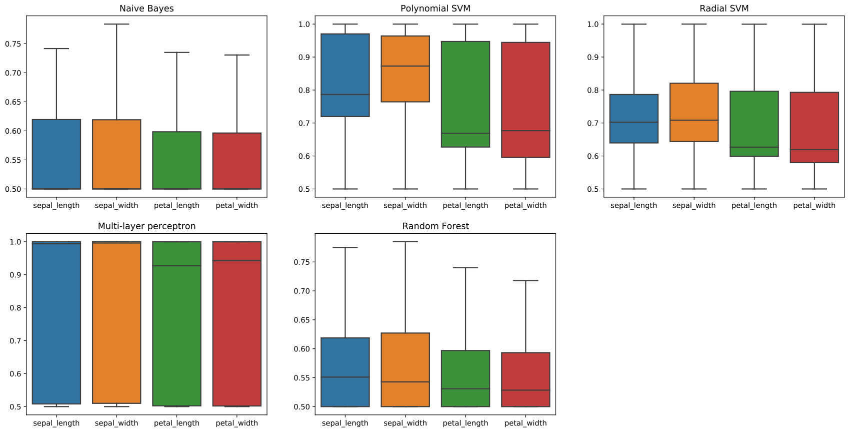

Figure 1: Distribution of the values of the damping factor

From the analysis of this figure we can see that the classifiers show completely different behaviors. In fact:

- Naive Bayes tends to assign similar and very low values to the damping factor for all features.
- Polynomial SVM assigns very different values to the damping factor for different features. Therefore, it shows a very good ability to discriminate features.
- Radial SVM shows differences in the values of the damping factor, although these are smaller than the ones shown by Polynomial SVM.
- Multi-Layer Perceptron returns very different values of the damping factor for the occurrences of the same feature. In contrast, median values are all very high. This classifier proved less capable of discriminating features than the two SVM classifiers, although it seems better than Naive Bayes.
- Random Forest returns results similar, albeit less extreme, to the ones returned by Naive Bayes. It does not reveal much ability to discriminate features.

The results on the damping factor shown above are indicative of potential trends but are still preliminary. In fact, they need to be confirmed or corrected by the analysis of the relevance values, which represent the final outcome of our XAI process. These results are shown in Figure 2. From the analysis of this figure we can conclude that:

- Naive Bayes and Random Forest are unable to discriminate feature relevances.
- The two SVM classifiers and Multi-Layer Perceptron are capable of discriminating feature relevances, although to different degrees.

- The differences identified by the various classifiers are concordant. In fact, the two SVM classifiers and, to some extent, also Multi-Layer Perceptron, show that `petal_length` and `petal_width` are more relevant than `sepal_length` and `sepal_width`.
- Polynomial SVM and Radial SVM prove to be the most capable of discerning differences in feature relevances.

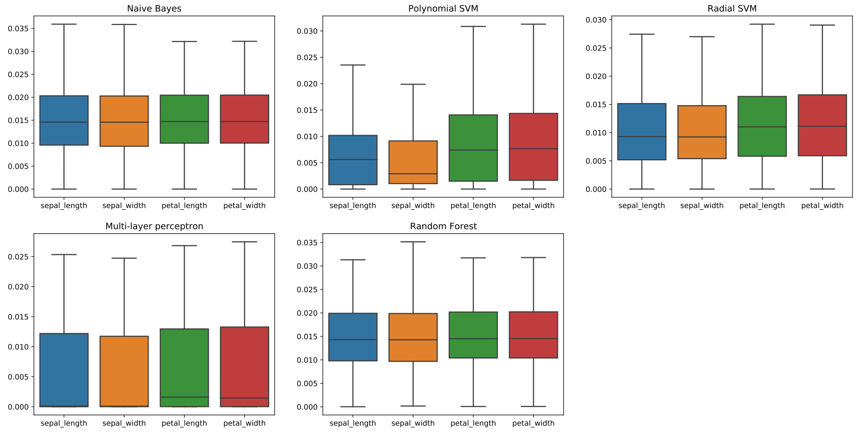

Figure 2: Distribution of the relevance values

The conclusions drawn from the examination of Figure 2 are qualitative and only partially quantitative. Actually, it would be important to find a way to quantify the different abilities of the classifiers to discern feature relevance. A first way to achieve this goal is to compare the median values of the occurrence relevances for each feature and for each classifier. These values are reported in Table 2. The analysis of this table shows that, even at the quantitative level, `petal_length` and `petal_width` are more relevant than `sepal_length` and `sepal_width`.

Model	*Feature*	*Relevance*	*Model*	*Feature*	*Relevance*
NB	sepal_length	0.014598	SVMP	sepal_length	0.005608
	sepal_width	0.014572		sepal_width	0.002904
	petal_length	0.014696		petal_length	0.007408
	petal_width	0.014714		petal_width	0.007660
SVMR	sepal_length	0.009293	MLP	sepal_length	0.000108
	sepal_width	0.009238		sepal_width	0.000082
	petal_length	0.011012		petal_length	0.001598
	petal_width	0.011139		petal_width	0.001454
RF	sepal_length	0.014313			
	sepal_width	0.014280			
	petal_length	0.014504			
	petal_width	0.014534			

Table 2
Median relevance of each feature returned by the five classifiers

A second, more accurate way to achieve the goal above is to introduce a new function $\alpha(\cdot)$. It receives a classifier $\mathcal{M}$ and returns a real number in the interval $[0, 100]$ that measures the ability of $\mathcal{M}$ to differentiate feature relevances. $\alpha(\cdot)$ can be defined as follows:

$$\alpha(\mathcal{M}) = \frac{max_{\mathcal{M}} - min_{\mathcal{M}}}{MaxCPI_{\mathcal{M}}} \cdot 100$$

Here, $max_{\mathcal{M}}$ (resp., $min_{\mathcal{M}}$) is the maximum (resp., minimum) value taken by the median relevance of a feature when $\mathcal{M}$ is adopted. $MaxCPI_{\mathcal{M}}$ (Maximum Central Percentile Interval) is obtained in the following way: first we compute the widths of the intervals between the values corresponding to the 25th and 75th percentiles of the distributions of the feature relevances returned by $\mathcal{M}$. Then, we calculate the maximum of these widths. In the formula of $\alpha(\cdot)$, we decided to take the values corresponding to the 25th and 75th percentiles, instead of all values, to avoid $\alpha(\cdot)$ being sensitive to outliers.

In Table 3, we report the values returned by $\alpha(\cdot)$ for the classifiers of our interest. This table gives us an accurate quantitative result of what we had guessed qualitatively from examining Figures 1 and 2 and Table 2. In particular, it allows us to conclude that the best classifier in differentiating feature relevances is Polynomial SVM, with a value of $\alpha(\cdot)$ equal to 37.47%, while the second best classifier is Radial SVM, with a value of $\alpha(\cdot)$ equal to 17.62%. Multi-Layer Perceptron is still a good classifier, while Naive Bayes and Random Forest are incapable of discriminating which features are most relevant.

	Naive Bayes	*Polynomial SVM*	*Radial SVM*	*Multi-Layer Perceptron*	*Random Forest*
Value of $\alpha(\cdot)$	1.29%	37.47%	17.62%	11.43%	2.50%

Table 3
Values of the function $\alpha(\cdot)$ for the classifiers into consideration

4. Conclusion

In this paper, we have proposed NT4XAI, a model-agnostic, network-based XAI framework to explain the behavior of any classifier. As its name indicates, NT4XAI is based on network theory and the vast amount of results obtained in this research area in the past. NT4XAI achieves its goal by evaluating the relevance of features in the behavior of a classifier. We also described some tests that allowed us to evaluate the effectiveness of NT4XAI both quantitatively and qualitatively. The main contributions of this paper are: *(i)* the definition of NT4XAI, a new model-agnostic network-based XAI framework; *(ii)* the definition of the concept of dyscrasia, by which the consistency of the occurrences of a feature during the classification process can be qualitatively evaluated; *(iii)* the definition of an approach for calculating the relevance of a feature in classifying the corresponding instances.

As for possible future developments of this research, we can first think of extending NT4XAI by considering latent structural properties in our network-based model. Also, we could use a totally different network model, such as a multilayer network [8, 26], to support NT4XAI. This would allow us to have a new point of view and capture different properties [1] using local model knowledge.

References

[1] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion 58 (2020) 82–115. Elsevier.

[2] D. Gunning, D. Aha, DARPA's Explainable Artificial Intelligence (XAI) Program, AI Magazine 40 (2019) 44–58. AAAI.

[3] J. Zini, M. Awad, On the Explainability of Natural Language Processing Deep Models, ACM Computing Surveys 55 (2022). ACM.

[4] A. Adadi, M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access 6 (2018) 52138–52160. IEEE.

[5] S. Yoo, N. Kang, Explainable Artificial Intelligence for manufacturing cost estimation and machining feature visualization, Expert Systems with Applications 183 (2021) 115430. Elsevier.

[6] D. Kaur, S. Uslu, K. J. Rittichier, A. Durresi, Trustworthy Artificial Intelligence: A Review, ACM Computing Surveys 55 (2022). ACM.

[7] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy AI: From Principles to Practices, ACM Computing Surveys (2022). ACM.

[8] M. Newman, Networks, 2018. Oxford University Press.

[9] S. M. Lundberg, S. I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[10] A. Razmjoo, P. Xanthopoulos, Q. Zheng, Online feature importance ranking based on sensitivity analysis, Expert Systems with Applications 85 (2017) 397–406. Elsevier.

[11] E. Strumbelj, I. Kononenko, An efficient explanation of individual classifications using game theory, The Journal of Machine Learning Research 11 (2010) 1–18. JMLR.org.

[12] P. Dabkowski, Y. Gal, Real Time Image Saliency for Black Box Classifiers, in: Proc. of the International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 2017, pp. 6970–6979. Curran Associates Inc.

[13] R. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proc. of the International IEEE Conference on Computer Vision (ICCV'17), Venice, Italy, 2017, pp. 3449–3457. IEEE.

[14] S. Brin, L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, Computer Networks 30 (1998) 107–117.

[15] N. Burkart, M. F.Huber, A survey on the explainability of supervised machine learning, Journal of Artificial Intelligence Research 70 (2021) 245–317.

[16] E. Štrumbelj, I. Kononenko, M. R. Šikonja, Explaining instance classifications with interactions of subsets of feature values, Data & Knowledge Engineering 68 (2009) 886–904. Elsevier.

[17] M. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD'16), San Francisco, CA, USA, 2016, pp. 1135–1144.

[18] M. Gosak, R. Markovič, J. Dolenšek, M. Rupnik, M. Marhl, A. Stožer, M. Perc, Network science of biological systems at different scales: A review, Physics of life reviews 24 (2018) 118–135. Elsevier.

[19] O. Sporns, Graph theory methods: applications in brain networks, Dialogues in clinical neuroscience (2022). Taylor & Francis.

[20] D. Camacho, A. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, E. Cambria, The four dimensions of social network analysis: An overview of research methods, applications, and software tools, Information Fusion 63 (2020) 88–120. Elsevier.

[21] A. Datta, S. Sen, Y. Zick, Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems, in: Proc. of the International Symposium on Security and Privacy (SP'16), IEEE, Fairmont, San Jose, CA, USA, 2016, pp. 598–617.

[22] A. Henelius, K. Puolamäki, A. Ukkonen, Interpreting classifiers through attribute interactions in datasets, arXiv preprint arXiv:1707.07576 (2017).

[23] R. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188. Wiley Online Library.

[24] A. Asuncion, D. Newman, UCI machine learning repository, 2007. Available online at: https://archive.ics.uci.edu/ml/index.php.

[25] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, Z. Siddique, Effect of data scaling methods on machine learning algorithms and model performance, Technologies 9 (2021) 52. MDPI.

[26] G. Bonifazi, B. Breve, S. Cirillo, E. Corradini, L. Virgili, Investigating the COVID-19 vaccine discussions on Twitter through a multilayer network-based approach, Information Processing & Management 59 (2022) 103095. Elsevier.