An Interpretable Adaptive Multiscale Attention Deep Neural Network for Tabular Data

Vincenzo Dentamaro, *Member, IEEE*, Paolo Giglio, Donato Impedovo, *Senior Member, IEEE*,
Giuseppe Pirlo, *Senior Member, IEEE*, and Marco Di Ciano

***Abstract*— Deep learning (DL) has been demonstrated to be a valuable tool for analyzing signals such as sounds and images, thanks to its capabilities of automatically extracting relevant patterns as well as its end-to-end training properties. When applied to tabular structured data, DL has exhibited some performance limitations compared to shallow learning techniques. This work presents a novel technique for tabular data called adaptive multiscale attention deep neural network architecture (also named excited attention). By exploiting parallel multilevel feature weighting, the adaptive multiscale attention can successfully learn the feature attention and thus achieve high levels of F1-score on seven different classification tasks (on small, medium, large, and very large datasets) and low mean absolute errors on four regression tasks of different size. In addition, adaptive multiscale attention provides four levels of explainability (i.e., comprehension of its learning process and therefore of its outcomes): 1) calculates attention weights to determine which layers are most important for given classes; 2) shows each feature's attention across all instances; 3) understands learned feature attention for each class to explore feature attention and behavior for specific classes; and 4) finds nonlinear correlations between co-behaving features to reduce dataset dimensionality and improve interpretability. These interpretability levels, in turn, allow for employing adaptive multiscale attention as a useful tool for feature ranking and feature selection.**

***Index Terms*— Adaptive multiscale attention, attention mechanism, deep learning (DL), excited attention, explainable artificial intelligence (AI), machine learning, squeeze and excitation (SE), tabular data.**

I. Introduction

IN THE last decade, deep neural networks have revolutionized the way machines learn, delivering high-accuracy results on several different tasks such as object recognition, image classification, natural language processing, audio classification, and much more [1]. In practice, deep learning (DL) has achieved outstanding results on unstructured data [2]. Also, there is an increasing need, in complex tasks, of merging various types of signals, such as video, image, sound, and tabular data, all together. An example of this multimodal learning can be referred to the medical field, where, for instance, computer-aided tomography (CT) is merged with medical records (i.e., tabular data) to improve the quality of acute stroke prediction [3]. A classic approach would consider the fusion at the decision level of different approaches (e.g., DL solutions for images and decision trees for tabular data); on the other hand, DL solutions also in the tabular data field would open the possibility to inspect multimodal DL integrated solution: each part of the multimodal architecture would be a neural network trained with gradient descent in an end-to-end fashion. In this direction, it must be considered that, in the last years, some attempts have been made to apply DL techniques to tabular data, i.e., data composed by samples (instances): Google's TabNet is an example [4]. Apart from this, the most popular techniques are based on ensembles of decision trees [5]: these techniques have been shown to provide good levels of accuracy even on datasets of limited size [6], [7]. Additionally, they usually have the intrinsic capability of approximating hyperplane boundaries [4]. Finally, these techniques come with a certain degree of interpretability: it is possible to compute post hoc feature ranking and accordingly to understand which features contributed the most to a particular task [8]. In general, tree-based techniques such as Random Forests [6], Catboost [9], Adaboost [10], and Gradient Boost [7], [11] do not allow an end-to-end training out of the box. So that, the challenge of this work has been to propose a DL architecture to process tabular data and able to exhibit the state-of-the-art performance while providing an explainability of the observed results. Moreover, the end-to-end training process would allow that all the modules of the learning algorithm would be optimized in terms of a specific outcome [12], [13], [14]. This would open new and unexplored integration of different types of data [12], [15]. It is essential to underline that gradient-based optimization usually requires a high amount of data (not always available). A partial solution to this problem could be the use of an attention mechanism acting as a weighting function on the inputs. In this way, only features containing most of the information for the needed task would be used [16]. To date, it is believed that the unsuccess of DL models on tabular data is due to various factors affecting tabular data such as: data quality; the presence of missing values and strong outliers; the lack of spatial dependencies among features, invalidating the use of inductive bias and thus of the algorithms designed to exploit it (e.g., convolutional neural networks); the choice of preprocessing technique to handle categorical variables; and deep neural network models sensitivity to small data perturbations [17]. On the other hand,

Manuscript received 20 January 2023; revised 15 April 2023, 9 October 2023, and 24 January 2024; accepted 18 April 2024. This work was supported in part by InnovaPuglia S.p.A., with the purpose of setting up research and scientific investigation on artificial intelligence algorithms targeting public administration data process (June 17, 2019), under Grant prot.n.47380111/6; and in part by the Project FAIR—Future AI Research, Spoke 6—Symbiotic AI, through the National Recovery and Resilience Plan (NRRP) Ministry for Universities and Research (MUR) Program funded by the NextGenerationEU, under Grant PE00000013. *(Corresponding authors: Vincenzo Dentamaro; Donato Impedovo.)*

Vincenzo Dentamaro, Paolo Giglio, Donato Impedovo, and Giuseppe Pirlo are with the Department of Computer Science, University of Bari Aldo Moro, 70125 Bari, Italy (e-mail: Vincenzo.dentamaro@uniba.it; donato.impedovo@uniba.it).

Marco Di Ciano is with the Department of Computer Science, University of Bari Aldo Moro, 70125 Bari, Italy, and also with InnovaPuglia S.p.A., Valenzano, 70010 Bari, Italy.

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2024.3392355, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3392355

it has been proved that deep neural networks for unstructured data are robust to noisy data and, in practice, they generalize well on a variety of problems [18]. A way to assess the ability of a machine learning algorithms to generalize in an unsupervised learning context is by calculating the maximum mean discrepancy (MMD) [19]. MMD serves as a distance metric that evaluates the difference between two probability distributions. It helps quantify the change in distributions between training and testing sets. This information can then be used to identify the approach that best generalizes to new, unseen distributions, typically resulting in higher accuracy as the distribution shift increases. An alternative approach involves visually examining the distribution of training and testing data by reducing the dimensionality and subsequently plotting the data distributions for both sets [20], [21], [22]. If the density estimations of the two sets exhibit considerable differences, the top-performing algorithm will likely be the one that generalizes more effectively. Assessing the overlap between training and testing data distributions can help gauge their similarity. A high degree of overlap suggests good generalization since the model is learning features relevant to both sets. On the other hand, low overlap may indicate overfitting or the learning of overly specific features from the training set. The visual examination of data distributions serves as a valuable supplement to evaluation metrics, providing an intuitive understanding of the model's ability to generalize. This insight can help pinpoint areas or features where the model faces challenges in generalization, thus informing potential improvements to the model's architecture or training process.

In this regard, the regularization techniques and variants of gradient descent algorithms lead to generalization [18], [23] because of the way the loss surface is shaped during training can affect the generalization capabilities of the model. The mentioned standard ensemble learning techniques (which are the state-of-the-art on tabular data) do not exhibit this behavior. In practice, for multimodal problems, where some input is tabular, some other parts can be made of various types of signals such as image, video, or audio, DL is the appropriate technique to use because of the gradient descent optimization across the various modalities. Of course, the tabular part should be processed by a proper neural network such as the architecture here proposed. In this light, this work introduces the adaptive multiscale attention mechanism (also named excited attention) as a way of increasing the generalizability of deep neural network for tabular data. Adaptive multiscale attention is inspired by the intuition of the excitation module in the squeeze and excitation (SE) networks. Indeed, this module can learn a multilevel latent-space representation of the data at different resolutions. This technique had great success when employed for computer vision applications where it is used for automatically selecting relevant feature-maps channel-wise [24]. In other words, adaptive multiscale attention learns at the same time various levels of compression and latent-space data representation used to weight features. In this work, these levels are merged in four possible configurable ways and a novel attention mechanism, called trainable attention, is then added to optimize the selection of relevant features. The so-called trainable attention mechanism learns the attention weights of the final features and filters the input values accordingly. It is designed to be robust on strong outliers and to reduce sensitivity to small data perturbations while ensuring high generalization capabilities. The main contribution of the adaptive multiscale attention network is that it provides a method for learning feature attention weights and correlations from both local and global perspectives in a scalable way. Specifically, the adaptive multiscale attention network uses a parallel execution of excitation layers with increasing squeeze ratios, which allows for the learning of compressed representations of feature weights. These compressed representations are then used to weigh the attention of features from a local perspective.

The adaptive multiscale attention network also uses a trainable attention layer (TrA Layer) that learns to weigh features and correlations from a global perspective. This is achieved by training an $f \times f$ weighting matrix in an end-to-end fashion using a gradient descent algorithm. The resulting weights are then used to compute the dot product of the normalized input representation and the sigmoid of the weighting matrix.

Overall, the adaptive multiscale attention network provides a method for learning interpretable feature attention weights and correlations in a scalable way that is suitable for tabular datasets. This is a novel contribution that is not addressed by many existing methods and has the potential to improve the performance of machine learning models on tabular datasets.

Of particular interests related to the work here proposed, are the works in [25] and [26], where the authors propose a multilocal channel excitation (MCE) block to investigate channel context by discovering the semantic relationships between feature maps of multiple local channels. The concept of multiscale learning with attention has been already proposed in various articles, but mostly related to image recognition. Specifically, in [27], the multiscale visual-attribute co-attention (mVACA) improves the categorization of unseen classes by linking multiscale visual and semantic variables utilizing hybrid visual attention in zero-shot image recognition. Always for image recognition, Li et al. [28] have developed a novel dual-channel spatial, spectral and multiscale attention convolutional long short-term memory neural network (A3 CLNN) model to effectively integrate hyperspectral images and LiDAR data for enhanced remote sensing data analysis using multiscale attention mechanisms as well as various attention mechanism for feature integration. To tackle the problem of semantic segmentation in computer vision, Shi et al. [29] Fare clic o toccare qui per immettere il testo. proposed a lightweight multiscale-feature-fusion network (LMFFNet) for real-time semantic segmentation that balances accuracy with parameter efficiency.

The main contributions of the architecture proposed in this work can be summarized as follows.

1) It achieves the state-of-the-art accuracies on several different small-, medium-, and large-size datasets for both classification and regression problems.
2) It shows good generalization capabilities while using multidimensional scaling and kernel density estimation plot among train and test sets for any problem. The resulting plots are shown in Appendixes A and B. Particularly, adaptive multiscale attention performs better than other techniques with problems that show a noticeable difference in density distributions between the train and the test sets. Therefore, it appears to be a powerful technique in terms of generalization power.
3) Given a fixed architectural design, the only hyperparameter to setup is the type of merging layer. In the current implementation, the merging layer can be one of the following: concatenate, add, average, or multiply.

4) No preprocessing and dimensionality reduction pipeline is needed.
5) It adds four levels of interpretability: a) excitation layer interpretability: calculates attention weights to determine which layers are most important for given classes; b) global magnitude feature attention: shows each feature's attention across all instances; c) global feature attention per class and behavior analysis: understands learned feature attention for each class to explore feature attention and behavior for specific classes; and d) nonlinear Spearman correlation among learned features: finds nonlinear correlations between co-behaving features to reduce dataset dimensionality and improve interpretability. It is important to highlight that feature weighting is not computed post hoc, thus concretely representing the feature attention given by the network during the training phase. Thus, the network learns how to pay attention to some features with respect to others. This learned correlation allows adaptive multiscale attention to be used as a feature selection technique.

The work is organized as follows. Section II reviews the state-of-the-art techniques for tabular data. Section III-A reviews the background in attention mechanisms, while Section III-B presents the adaptive multiscale attention network. Section IV describes the interpretability of adaptive multiscale attention. Section V sketches the experimental setup and datasets used. Results and discussion are provided in Section VI. The discussion on the interpretability of the results of the adaptive multiscale attention is provided in Section VII. Section VIII shows conclusions and future research directions.

II. Related Work

The state-of-the-art machine learning techniques on tabular data can be based on an ensemble of decision trees or on standalone solutions. Among the ensemble of tree-based models, one of the top-performing algorithms is extreme gradient boosting (XGBoost): it is an optimized distributed gradient boosting [30]. In the gradient boosting framework, trees are built sequentially so that each subsequent tree aims to minimize the residual error of the previous ones. In other words, each tree learns from its predecessors and updates the residual errors. XGBoost enhances the gradient-boosting framework through system optimizations and algorithmic improvements. In particular, XGBoost applies the following optimizations.

1) *Parallelization:* The algorithm addresses the process of sequential construction of trees using a parallel implementation. There are two loops working in parallel: the first one enumerates the leaf nodes of a tree and the second one calculates its characteristics.
2) *Hardware Optimization:* The algorithm is designed for efficient hardware usage. The cache allocates internal buffers in each thread to store the gradient statistics. Additional enhancements, such as out-of-core processing, optimize the available disk space by handling large data frames that do not fit in memory.
3) *Tree Trimming:* The algorithm trims unnecessary trees "backward," thus reducing computational overhead without worsening the performance of the model.

Algorithmic improvements have been also proposed.

1) *Regularization:* A penalization of more complex models by the employment of least absolute shrinkage and selection operator (LASSO) and ridge regularization to avoid overfitting [31].
2) *Missing Value Handling:* The model can learn about the lack of missing values and shape them efficiently by automatically learning the best direction to take for each split based on the available values. XGBoost uses a technique called "sparse-aware" split finding to handle missing values efficiently.
3) *Cross-Validation:* The algorithm performs a validation of the results at each iteration to verify their acceptability.

XGBoost works well on small data, data with subgroups, big data, and complex data. However, it does not perform at its best on sparse data; very distributed data can also create problems. Anyway, it outperforms many supervised learning algorithms. Similar to Random Forest, it lacks end-to-end training along with all its properties.

Decision forests, like XGBoost, similar to an end-to-end fashion, can process raw data and predict labels without preprocessing or feature engineering. However, their sequential training process, involving adding trees and optimizing hyperparameters, makes them harder to fine-tune. The adaptive multiscale attention model, a deep neural network, simultaneously learns feature representations and predictions, optimizing all parameters jointly for better performance. Despite this, decision forests remain effective for many tasks, and the proposed model is not meant to replace them entirely.

Another state-of-the-art technique belonging to the ensemble of decision trees category is CatBoost [9]. It is an algorithm designed by Yandex with the purpose of effectively managing categorical features without having to rely on procedures like encoding results in a high-dimensional space. One of the possible approaches to managing categorical features without encoding is computing label-dependent statistics, such as replacing a value with the average of the labels having that value for each categorical feature.

In order to overcome this problem, a part of the dataset should be used to compute these statistics, but this would result in a reduction of the data available for training and validation. CatBoost addresses this issue by performing a permutation of the entire dataset and, for each row, by calculating the mean with respect to the instances preceding it. The algorithm is a special case of the gradient boosting decision tree (GBDT) [7] algorithm, as it natively supports ordered and categorical variables. Some disadvantages are as follows.

1) It needs to build deep decision trees in order to seek dependence when features have high cardinality.
2) It does not work for unknown category values, which in turn means that these cannot be inferred.

Among techniques making use of an ensemble of decision trees but also based on deep neural networks, it is possible to find neural oblivious decision tree ensembles (NODE) [15]. NODE is built on top of equal-depth oblivious differentiable decision trees. The differentiable property of the decision trees integrated into the NODE algorithm allows the model to be trained end-to-end, and thus, errors can be backpropagated through the differentiable trees using gradient descent. It uses the same splitting function for all nodes at the same level making computation high parallelizable. It uses the EntMax transformation [32] being based on decision trees, and it is not necessary to employ any kind of data preprocessing to handle categorical variables.

TabNet [4] is a deep neural network for tabular data; it was designed to learn in a similar way to decision tree-based models with the goal of earning the same benefits: interpretability and sparse feature selection. It uses multiheaded attention in a sequential way to select features to rely on at each decision step. This multistep learning enables high interpretability and lower error costs. The feature selection is performed depending on the instances, i.e., it can vary because of the specific input instance. It has several advantages, such as integrating multiple data types like images or tabular data, and it supports end-to-end training along with all its desirable properties. One of the main drawbacks is the huge amount of data required for the learning process, which is, by the way, one of the biggest limitations of the multiheaded attention model. Thus, TabNet is not designed to work with small or medium-sized datasets.

Another deep neural network architecture, which is built to imitate the GBDT-like models (XGBoost and CatBoost), is Net-DNF [33]. The key insight about Net-DNF is that every decision tree is a Boolean formula in disjunctive normal form. In Net-DNF, hard Boolean formulas are replaced with differentiable and fuzzy versions of them. The final architecture is built on the innovative disjunctive normal neural form (DNNF) block containing a multilayer perceptron (MLP) and one DNNF layer composed of a soft version of binary conjunctions over a differential version of AND and OR gates. The model is an ensemble of DNNF blocks.

Among classical methods for tabular data not involving an ensemble of trees, there are feed forward neural networks, also known as MLP networks, and support vector machines (SVM).

In this work, the following approaches have been considered and tested for comparison aims:

1) tree-based:
    a) decision tree;
    b) random forests;
    c) XGBoost;
    d) catboost.
2) non-tree-based:
    a) TabNet;
    b) MLP;
    c) SVM.

The choice of these techniques has been driven by the following reasons.

1) The high performance generally observed (i.e., these are, in general, the best-performing ones).
2) Their large use in many relevant tasks (i.e., these are the most used).
3) The presence of a stable and tested implementation or package (for replication aims).

III. Adaptive Multiscale Attention

A. Attention Mechanisms—Background

A relevant part of the method proposed here is based on the attention mechanism. An attention mechanism consists of weighting the inputs (of a neural network) in such a way as to highlight relevant information during the time or among the available samples. Attention layers are generally used between input and output elements (general attention) [34] or only by focusing on the layer's inputs (self-attention) [35].

General attention emphasizes the importance of certain features or channels in a model's output. The adaptive multiscale attention enhances this concept by learning attention weights using various excitation blocks with differing compression ratios. SE [24] attention recalibrates feature responses in models by capturing interdependencies between channels, differing from adaptive multiscale attention which focuses on featurewise importance using a trainable matrix. Self-attention [36], predominantly used in natural language processing, determines attention based on input similarities, contrasting with adaptive multiscale attention that centers on featurewise importance. Spatial attention [37] models relationships between spatial locations in images or feature maps, diverging from the feature-centric focus of adaptive multiscale attention. Temporal attention [38], suitable for sequence-to-sequence models, emphasizes the relationships between sequence time steps, while adaptive multiscale attention hinges on feature significance. Multihead attention, linked to the Transformer architecture [36], calculates several attention scores concurrently, as opposed to adaptive multiscale attention that computes a single set of featurewise attention scores. Graph attention network (GAT) [14], tailored for graph data, concentrates on connections between graph nodes, in contrast to the feature-focused of the adaptive multiscale attention. In essence, while other attention mechanisms, including self-attention, spatial attention, temporal attention, and GAT, model relationships among input elements, spatial points, time steps, or graph nodes, respectively; adaptive multiscale attention distinctively emphasizes feature wise importance within one layer.

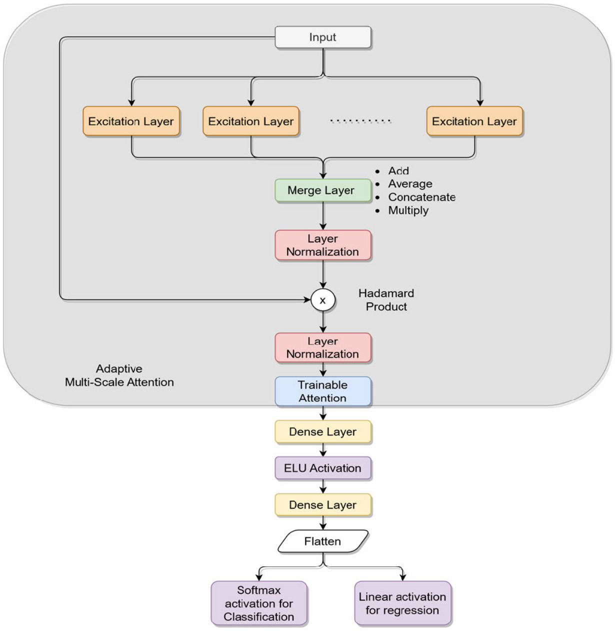

Fig. 1. Adaptive multiscale attention network.

B. Adaptive Multiscale Attention

Adaptive multiscale attention is a deep neural network composed of several parallel excitation layers and a TrA Layer. The adaptive multiscale attention network is depicted in Fig. 1: it is based on the intuition of the parallel execution of excitation layers with an increasing squeeze ratio. This parallel execution can allow learning n compressed representation of features weights having

$$n = \max\left(2, \sqrt{f}\right) \tag{1}$$

being f the number of features (columns) of the tabular dataset. This ensures a sublinear network growth.

The excitation layer is depicted in Fig. 2. As it is possible to observe, it is composed of a dense layer with a number of neurons equal to $f*r$, where r is the squeeze ratio having $r \in M$

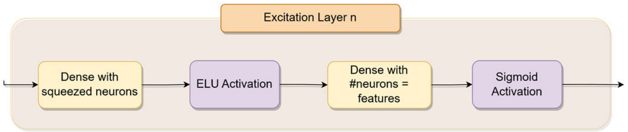

Fig. 2. Excitation layer.

and M is the set of n evenly spaced real numbers containing values in the range $[(1/n), 1]$, one for each parallel excitation layer.

The squeeze ratio r is used to create a nonlinear compressed embedding space of the original feature space. The size of this embedding space is varying, allowing it to capture information at varying expressive power.

Computing parallel attention scores at varied compression ratios enhances a model's adaptability and expressive capacity. Specifically, this approach facilitates the following.

1) *Multiscale Feature Learning:* Different compression ratios enable the model to discern features at multiple granularities. Lower ratios emphasize detailed local features, while higher ratios capture broader, abstract characteristics, leading to a holistic data comprehension.
2) *Enhanced Adaptability:* It exhibits improved adaptability to diverse datasets and tasks by capturing both intricate details and abstract notions.
3) *Augmented Expressive Capacity:* It amplifies the model's capability to comprehend intricate relationships, thereby boosting performance and representation learning.
4) *Robustness:* Attention scores at multiple scales render the model resilient to data noise and variations, as it leverages various abstraction levels, minimizing sensitivity to minor data perturbations.

This dense layer is followed by an exponential linear unit (ELU) [39], [40], activation function, and another dense layer having the number of neurons equal to f and a sigmoid activation function.

The SE paper uses a sigmoid nonlinearity for computing attention scores due to its ability to produce outputs in the [1, 0] range, smoothness, and differentiability. Attention scores are used as weights for rescaling feature maps, requiring values in this range. Sigmoid's properties, such as bounded output, smoothness, interpretability, and saturation, make it suitable for attention mechanisms. In contrast, activation functions like rectified linear unit (ReLU) and its variants are not ideal for this purpose because their outputs are not constrained between [1, 0], which is necessary for generating attention weights.

The choice of ELU activation function instead of standard ReLU activation is about its properties of avoiding the dead ReLU problem and its capability of producing negative weights and thus activation when calculating the gradient instead of sharply cutting them. Specifically, in the context of the adaptive multiscale attention model, the ELU activation function offers benefits like smoothness, stability, and faster convergence. Smoothness helps with gradient-based optimization and attention distribution. Stability results from a balance between positive and negative output values, making the attention mechanism less sensitive to small input perturbations. Faster convergence compared to ReLU can improve training efficiency. However, it is crucial to experiment with different activation functions to find the best fit for a specific problem or dataset.

Each excitation layer is used to weigh the compressed representation of original features with respect to the goal task. Formally

$$s = \varphi(W_2\delta(W_1 z)) \tag{2}$$

where z is the input, δ is the ELU function [39], [40], $W_1 \in \mathbb{R}^{(f*r)xf}$, $W_2 \in \mathbb{R}^{fx(f*r)}$, and φ is the sigmoid activation function.

The interesting concept behind this architecture is that the parallel execution of excitation layers with different compression ratios r would allow the network to learn multilevel (multiscale) feature embeddings with a varying embedding space. This varying embedding space allows capturing nonlinear feature attention weights at different scales with varying expressive power. The expressive power can be defined as the capability of the network to learn meaningful feature correlations. After the computation of the parallel excitation blocks, a tensor S is obtained as follows:

$$S = [s_1 \ldots, s_n],\ S\in\mathbb{R}^{fxn} \tag{3}$$

containing feature weighting at varying compression ratios. This allows to weigh features' attention from a local perspective.

Successively, all the s_i are merged according to four different and alternative ways: concatenation, sum, average, and Hadamard product as in the following equation, where $\odot$ is the concatenation operator. In the current implementation, there are four ways of merging information, these are the hyperparameters of adaptive multiscale attention model. New ways of merging layer can also be proposed in future works:

$$W_l = \begin{cases} \left[s_1 \odot s_2, \ldots, s_{n-1} \odot s_n\right] & \text{Concatenate} \\ \displaystyle\sum_{i=1}^{n} s_i & \text{Sum} \\ \dfrac{1}{n}\sum\nolimits_{i=1}^{n} s_i & \text{Average} \\ \prod\nolimits_{i=2}^{n} s_{i-1} * s_i & \text{Hadamard Product.} \end{cases} \tag{4}$$

The idea of using different attention heads, like in transformers, will also be explored in a future work. The main reason for not using multiple attention heads in the excitation block is to keep the design simple and computationally efficient. The primary goal of the excitation block is to provide a lightweight attention mechanism that can be easily integrated into existing architectures, such as the one here provided, with minimal computational and memory overhead.

The merged representation W_l is then normalized using layer normalization [41] where merged activations from the previous layer are normalized by subtracting the mean activation across the layer and dividing by the standard deviation. The normalization procedure is repeated after multiplying the merged representation with the input using the Hadamard product.

By applying the normalization layer after multiplying the inputs and attention weights, the excitation block can focus on learning the relative attention weights of the features without being affected by the scale of that features. The subsequent normalization layer then ensures that the output feature maps are on the same scale as the input feature maps, allowing for better learning in the subsequent layers. Thus, the normalization layer is applied after the multiplication of inputs and attention weights in the excitation blocks to maintain the scale of the output feature maps and to allow the attention mechanism to focus on learning featurewise dependencies without being influenced by the initial scale of the feature maps.

Let $k^{W_{lz}}$ be the output of the last layer normalization in Fig. 1; this initial feature weighting mechanism is called "normalized attention."

In normalized attention, it is possible to observe the weighting of each feature with respect to the various compression ratios.

The TrA Layer is composed of an $f * f$ trainable matrix w_t (weighting matrix) that is initialized as shown in the following equation:

$$w_t = I * \tau \tag{5}$$

being $I \in R^{fxf}$ the identity matrix and τ a constant. In this work, τ is 0.1 (empirically determined and adopted as a default value). Successively, the matrix w_t is trained: a gradient descent algorithm trained in an end-to-end fashion can change w_t values in order to reduce the error. This allows to automatically weigh features and correlations from a global perspective.

Thus, the learning process is performed as in the following equation:

$$\text{TrA} = k^{W_{lz}} \times \varphi(w_t).. \tag{6}$$

TrA is the dot product of $k^{W_{lz}}$ with the sigmoid of w_t. The sigmoid function is used to control the values of w_t matrix squashing them into [1, 0] range, thus avoiding gradient exploding and vanishing problems in subsequent layers.

The specific TrA design offers several advantages, including controlled attention weights, trainability, feature selection and emphasis, flexibility, and interpretability. The sigmoid function helps regulate attention weights, preventing gradient issues. The TrA mechanism is trainable, allowing the model to learn important features and adjust weights. It also emphasizes important features while suppressing less relevant ones. TrA can be easily integrated into various DL architectures, and its attention weights offer some interpretability. While not optimal for all situations, TrA's simplicity and adaptability make it an attractive choice for incorporating attention into DL models.

This concludes the adaptive multiscale attention mechanism. The output of the TrA Layer is passed to two dense layers with the number of neurons equal to the number of features f and the ELU activation function to perform the nonlinear learning as shown in (7). This last output, called u, is then passed to flatten function F_f that collapses the u tensor into a 1-D vector. Sequentially, its output is passed to a dense layer with softmax activation function γ for classification purposes or to a linear activation function ϕ for regression tasks, as shown in the following equations:

$$u = \delta\big(W_j \delta(W_k \text{TrA})\big) \tag{7}$$

$$v = \begin{cases} \gamma\big(W_t F_f(u)\big), & \text{if classification} \\ \phi\big(W_t F_f(u)\big), & \text{if regression.} \end{cases} \tag{8}$$

IV. Interpretability of Adaptive Multiscale Attention

Understanding the processes behind the decisions made by an artificial intelligence system is an increasingly essential issue.

It is important to state that, in general, a deep neural network' internal mechanisms cannot be easily interpreted because the behavior of the network with respect to an input is determined by nonlinear interactions among neurons and firing schemes learned during training. The adaptive multiscale attention mechanism proposed here is engineered to be a semitransparent box model which does not require any external optimization or approximation module to provide an explanation of its predictions: the output of each excitation layer is simply the nonlinear weighting of the original input features.

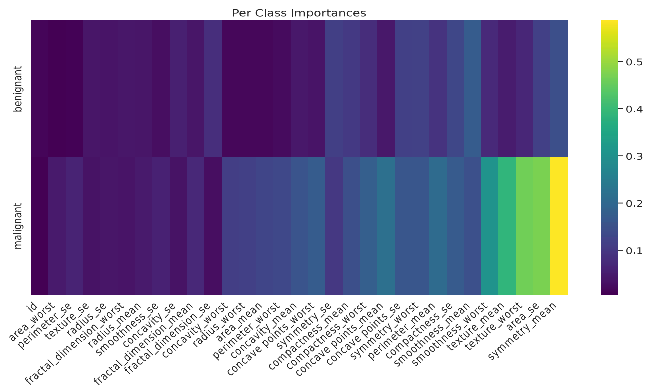

Fig. 3. Plot of learned feature attention with respect to each excitation layer and class (0, 1).

This is different from saliency maps [42] and activation maximization techniques [43], because, opposed to activation maximization, there is no gradient descent optimization nor activation maximization computation. Differently from saliency maps, no filter and no spatial relationship (as in the case of pixels) is present. It is just the plot of the output of the network cut at a certain point solicited by a given input instance.

The adaptive multiscale attention mechanism aims to improve the interpretability of deep neural networks by providing insights into the attention mechanism and relationships of input features. It does so through the following levels.

1) *Interpretability of Excitation Layers:* By calculating the attention weight of each excitation layer concerning different classes, the mechanism offers insights into which layers are more important for specific classes.
2) *Global Magnitude Feature Attention:* This metric reveals the overall attention of each feature across all instances. It is computed as the mean of the absolute values obtained by soliciting the adaptive multiscale attention network.
3) *Global Feature Attention Concerning the Class and Behavior Analysis:* This level of interpretability shows the learned feature attention with respect to each class, allowing you to analyze feature attention and behavior for specific classes.
4) *Nonlinear Spearman Correlation Among Learned Features:* This step helps identify nonlinear correlations among features that tend to co-behave, providing valuable information for reducing dataset dimensionality and increasing interpretability.

Overall, the adaptive multiscale attention mechanism strives to provide an interpretable understanding of the neural network's internal processes, focusing on feature attention, correlations, and behavior analysis. While it may not provide complete transparency, it offers semitransparent model than traditional deep neural networks. Sections IV-A–IV-D report the description of the interpretability of the levels cited above.

A. Interpretability of the Excitation Layers

The attention weight of each excitation layer is computed with respect to the classes: it simply consists of the sum of the feature weights per class, computed for each excitation layer when providing as input the whole test set and their ground trough labels as in the following equation:

$$W_{\mathrm{exc},j} = \begin{cases} \sum_{k=1}^{n} \sum_{i=1}^{a_0} \phi(S_k(J_i)), & \text{if label}_{J_i} = 0 \\ \sum_{k=1}^{n} \sum_{i=1}^{a_1} \phi(S_k(J_i)), & \text{if label}_{J_i} = 1 \\ \vdots & \\ \sum_{k=1}^{n} \sum_{i=1}^{a_T} \phi(S_k(J_i)), & \text{if label}_{J_i} = T \end{cases} \tag{9}$$

where W_{exc} is the output weight matrix for the n excitation layers, $A = [a_0, a_1, \ldots, a_t]$ are the total number of instances belonging to each class from 0 to T. J_i is the ith instance input, ϕ is the trained adaptive multiscale attention network output cut at the specific nth excitation layer $S = [S_1 \ldots S_n]$ shown in (3).

Equation (9) computes the attention weights of each excitation layer with respect to the different classes. Specifically, for each class (denoted by T in the equation), the feature weights for each instance in the test set are summed up across all excitation layers, resulting in a single weight value for that class. The output of this computation is a weight matrix W_{exc}, where each row represents an excitation layer and each column represents a class. The resulting values in each cell of the matrix represent the attention weights of that excitation layer for the corresponding class.

To give an example, suppose there are three excitation layers and four classes. W_{exc} would be a 3×4 matrix, with each row representing an excitation layer and each column representing a class. The values in each cell of the matrix would represent the attention weight of that excitation layer for the corresponding class.

For instance, if the value in the cell at row 1 (representing the first excitation layer) and column 2 (representing the second class) is 0.8, this would indicate that the first excitation layer is particularly important for class 2, while the values in other cells may indicate different levels of attention for different layers and classes. The pictorial result is represented in Fig. 3.

B. Global Magnitude Feature Attention

The global feature attention is given in the following equation:

$$\varrho = \frac{1}{k} \sum_{i=1}^{k} |\phi(W_{lz}(J_i))| \tag{10}$$

where J_i is ith input instance, k is the total number of instances, ϕ is the trained neural network, and let W_{lz} be the output of the Hadamard product between the output of the first layer normalization and the input feature vector.

In other words, it is computed as the mean of the absolute values obtained by the attention weights of the adaptive multiscale attention network, which in turn is cut at the Hadamard product in Fig. 1. The absolute value is relevant because the attention weights are calculated globally and not with respect to a specific class; thus, if an attention weight provides strong negative values for a specific class on specific features, this implies that these features are very sensitive (even if associated with a negative elicitation) for the global task.

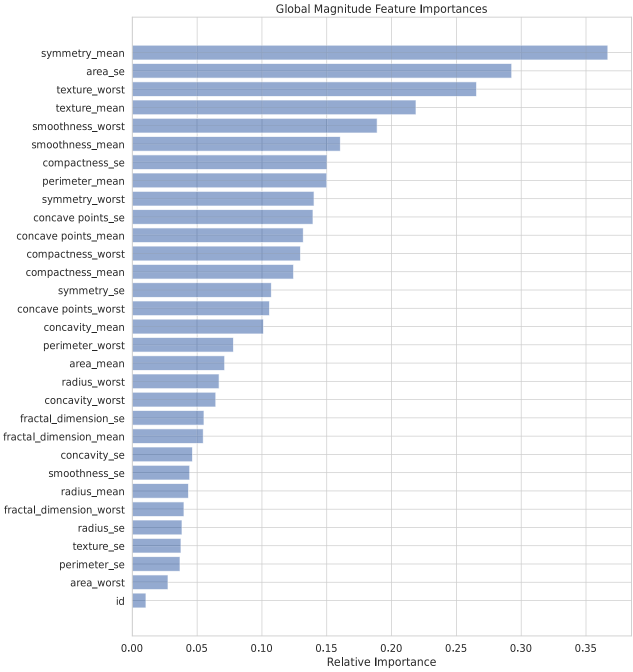

Fig. 4. Ranking of learned global feature attention.

In the context of this article, an attention weight refers to the process of passing the input instances through the adaptive multiscale attention network and extracting the corresponding weights at a specific layer. The weights represent the learned attention of the input features for the given task. In the case of the adaptive multiscale attention network, the attention weights extracted at each layer can be used to analyze the learned feature representations and guide the selection of optimal compression ratios for each layer.

The result of this computation is presented in Fig. 4.

C. Global Feature Attention With Respect to the Class and Behavior Analysis

The learned feature attention with respect to each class W_{imp} is evaluated according to the following equation:

$$W_{\mathrm{imp}} = \begin{cases} \frac{1}{a_0} \sum_{i=1}^{a_0} |\phi(W_{lz}(J_i))|, & \text{if label}(J_i) \text{ is } 0 \\ \vdots & \\ \frac{1}{a_T} \sum_{i=1}^{a_T} |\phi(W_{lz}(J_i))|, & \text{if label}(J_i) \text{ is} T \end{cases} \tag{11}$$

where $A = [a_0, a_1, \ldots, a_T]$ are the total number of instances belonging to each class from 0 to T.

In other words, W_{imp} is the average of absolute values of the attention weights.

This equation represents behavior in direction and magnitude of attention with respect to the input and the respective class. A pictorial representation is depicted in Figs. 5 and 6 for, respectively, class 0 and class 1. Thanks to (11), it is possible

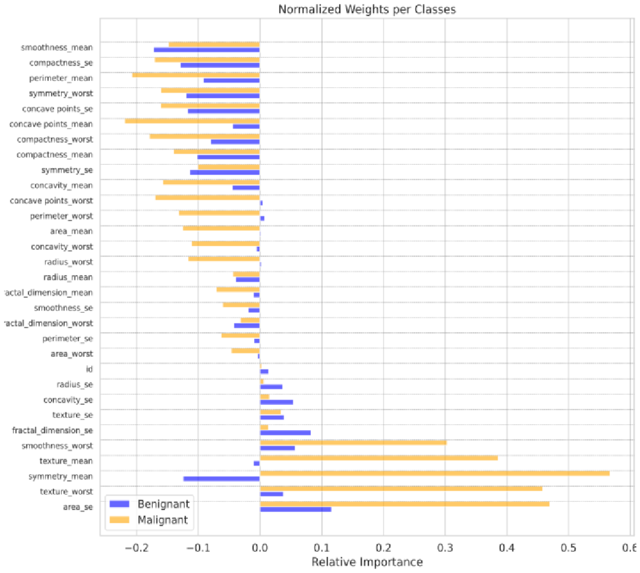

Fig. 5. Is the average of attention weights from the output of the Hadamard product layer with respect to class 0 (benignant).

to visualize features that tend to co-behave. This leads to the fourth analysis: nonlinear feature correlations.

D. Nonlinear Spearman Correlation Among Learned Features

The motivation for analyzing nonlinear Spearman correlation among learned features lies in addressing the "curse of dimensionality" problem [51] and increasing interpretability. By identifying pairs of features that tend to co-behave and removing one of them, while keeping the most important one, it is possible to reduce the dimensionality of the dataset, which in turn simplifies the problem and improves the interpretability of the results. This analysis also serves as an important tool for post hoc examination of correlations between variables, which is useful in a scenario of causal learning of effects directly from the data. Nonetheless, this is an important tool for an "a posteriori" analysis of correlations between variables, which is useful in a scenario of causal learning of effects directly from the data.

The Spearman correlation is a statistical measure of the magnitude of a monotonic relationship between paired data. This correlation metric has been considered here because it tends to model monotonic relationships of variables that change together but are not necessary at the same rate. In this case, it is not possible to guarantee the same rate of change between features, thus the choice of Spearman's correlation [52].

Let us consider

$$Ws^T = \left[\phi\left(W_{lz}\left(J_0^T\right)\right), \ldots, \phi\left(W_{lz}\left(J_i^T\right)\right)\right] \tag{12}$$

where Ws^T is the vector of weights extracted by the network ϕ at the layer W_{lz} with respect to the ith input instance J_i belonging to class T and ρ is the Spearman correlation. The correlation matrix W_{spear} is computed as in the following equation on all the weights presented in (12):

$$W_{\text{spear}} = \frac{1}{T}\sum_{k=0}^{T}\left|\rho\left(Ws_k^T\right)\right| \tag{13}$$

where T is the total number of classes.

Weights are extracted from the trained adaptive multiscale attention network model cut at the Hadamard product layer by passing test instances with respect to each class. Its nonlinear nature is due to the fact that attention weights are learned in a nonlinear way; thus, they represent the nonlinear co-behavior of features. The absolute value of the Spearman correlation is then kept into consideration; therefore, the final value ranges between 0 (no correlation or co-movement) and 1 (maximum correlation or perfect co-movement, even if opposite). The mean of the absolute correlation among all classes is finally computed: it represents the nonlinear correlation among features that tend to co-behave in both the same and the opposite direction.

V. Experimental Setup

Experiments have been performed on 11 different datasets, seven used for classification tasks and four for regression tasks with various sizes ranging from small, medium, large, and very large, as shown in Table I.

The following datasets have been used for the classification task.

1) *UCI Arrhythmia [44]:*
The dataset includes 452 patients with 279 features, including width, height, age, and gender, plus feature-engineered data from ECG. There are 16 classes, where all classes different from *class 1* (normal) are reported to have an abnormal ECG pattern.
2) *UCI Winsconsin Breast Cancer Diagnostic [45]*:
The dataset contains 569 instances with 32 features computed from digitized images of a fine needle aspirate of a breast mass. Features describe various characteristics of the cell nuclei that are present in the extracted image. It is a binary classification problem (malignant and benignant).
3) *UCI Cervical Cancer [46]:*
The dataset is composed of 72 instances and 19 attributes regarding cervical cancer behavior risk and two classes: 1 if the patient has cancer, 0 if not.
4) *UCI Diabetic Retinopathy Debrecen [47]:*
The dataset contains 1151 instances and 20 features. Features are extracted from the Messidor image that is set to predict whether an image contains signs of diabetic retinopathy or not, as specified in [47].
5) *UCI Heart Disease [48]:*
The dataset contains 303 instances and 14 features carefully selected by relevant works from a set of 75 features as in [57], [58], [59], and [60]. There are five classes, class 0 means normal, while all other numbers from 1–4 mean the presence of different heart diseases.
6) Click-through rate prediction of advertising when a query is given. It was retrieved from KDD Cup 2012 (Track 2) [49].
A fundamental technology behind search advertising is the click-through rate prediction of advertising banners since ads are ranked and priced with respect to the amount of clicks they receive. The data are derived from real session logs from Tencent's search engine known as soso.com. The problem is to predict if an advertisement will be clicked or not. This dataset is a subset of the original dataset and contains 1 000 000 instances balanced 500 000 positive class and 500 000 negative

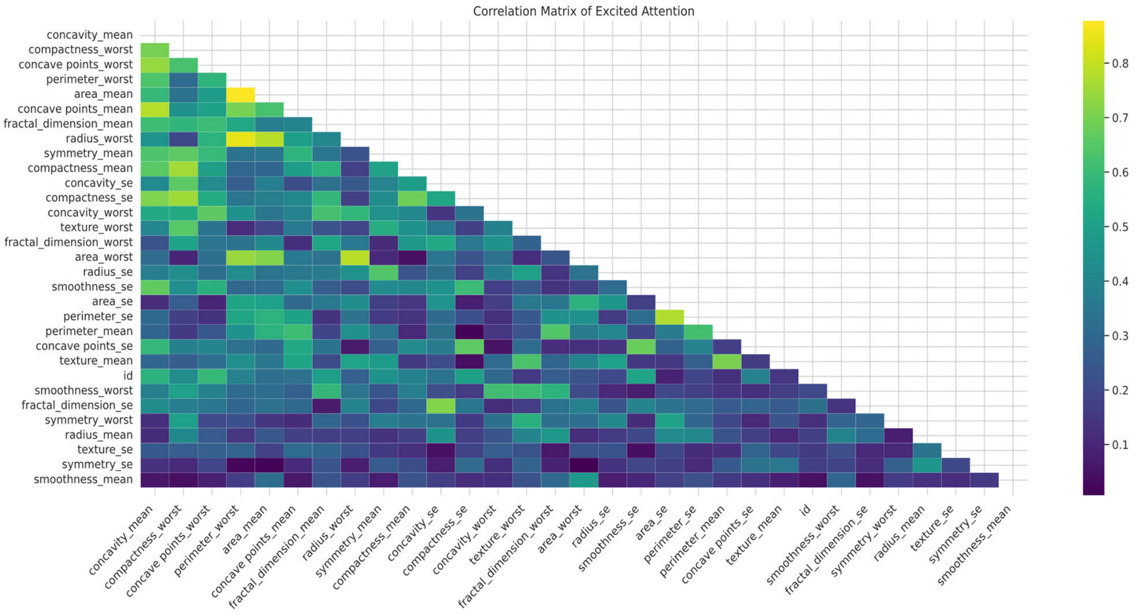

Fig. 6. Nonlinear features correlation matrix.

TABLE I

DATASETS USED FOR THE EXPERIMENTS

	Instances	Features	Problem type	Metric	Size
Arrhythmia [45]	452	279	Classification	F1-Weighted	Small
Winsconsin Breast Cancer [46]	569	32	Classification	F1-Weighted	Small
UCI Cervical Cancer [47]	72	19	Classification	F1-Weighted	Small
Diabetic Retinopathy Deb. [48]	1151	20	Classification	F1-Weighted	Small
UCI Heart Disease [49]	303	14	Classification	F1-Weighted	Small
Click-Through Rate [50]	1000000	12	Classification	F1-Weighted	Large
Higgs Boson [51]	11000000	28	Classification	F1-Weighted	Very Large
Echocardiogram [52]	132	13	Regression	Mean Absolute Error	Small
Bike sharing hourly [53]	17289	16	Regression	Mean Absolute Error	Medium
Year Prediction Million Song [54]	515345	90	Regression	Mean Absolute Error	Large
Yahoo Learning to Rank [55]	9436184496	519	Regression	Mean Absolute Error	Very Large

class with 12 features. The majority of instances are identifiers that are one-hot encoded. The target variable (click) is binary; thus, only two classes are identified.

7) *Higgs Boson [50]:*
The goal of this dataset is to distinguish between a signal process that produces Higgs bosons and a background process that does not. The dataset is composed of 11 000 000 instances and 28 numerical features. This is a binary classification problem.

Concerning the regression task, the following datasets have been considered.

1) *UCI Echocardiogram [53]:*
The goal of this dataset is to predict the number of months a patient survived since all patients had heart attacks at different times. The dataset consists of only 132 instances and 13 numerical features.

2) *Bike Sharing Hourly [54]:*
The dataset contains the hourly count of rental bikes retrieved between 2011 and 2012 in the Capital bike share system with the corresponding seasonal and weather information. The dataset is composed of 17 289 instances and 16 features. The goal is to predict the count of total rental bikes for each day.

3) *Year Prediction Million Song Dataset [55]:*
The dataset contains 515 345 instances with 90 numerical features. The goal is to predict the year of a song, given its audio features. Songs are mostly western commercial tracks ranging from 1922 to 2011 with a peak in the 2000s. In order to avoid the producer effect, which means that different songs from the same producer fall in both train and test, the training and testing split originally suggested were respected: 463 715 instances were used in the training set and 51 630 in the test set.

4) *Yahoo Learning to Rank [56]:*
This dataset is large. It is composed of 19 944 queries and 473 134 documents. Each query-document pair consists of 519 features. The label denotes the relevancy of each query-document pair. The goal is to predict its relevance ranging from 0 (irrelevant) to 4 (very relevant).

Datasets have been selected with the idea of reporting different algorithms' accuracies and errors with respect to the number of instances and the number of features, thus the intrinsic complexity of the dataset.

Table I summarizes the datasets details. Some of them, such as the click-through rate prediction, have mainly identifiers that are one-hot encoded, creating a very large number of dimensions and thus a very sparse dataset. The idea is to stress the algorithms from various points of view: capabilities of managing large sparse datasets, the capability of approximating hyperplane boundaries with highly nonlinear data, continuous or discontinuous function approximation (such as in the case of learning to rank regression problems), and generalization power on limited size datasets.

For each dataset, all categorical variables were one-hot encoded; numerical variables were used without any preprocessing technique. Datasets were standardized with z-score normalization prior to the training process in order to use standard algorithms such as MLP and SVM. Training, validation, and test sets were used for all algorithms in exactly the same conditions. Regarding different approaches here considered for comparison aims, in the following implementation/parameters, details are reported. SVM adopts radial basis function (RBF) kernel with γ equal to $(1/f)$, where f is the number of features. Random Forest is configured to use 100 trees and "gini" index. XGBoost is configured with a max depth prepruning parameter equal to 6 and a learning rate equal to 0.3, as suggested by the authors of the algorithm [30]. CatBoost is set up without any hyperparameter tuning or selection and with default parameters. TabNet's attention embedding width and width of decision precision are both equal to 8 [4].

The feed-forward neural network (called MLP) is configured with one single hidden layer having the number of neurons equal to $f * 0.8$, being f the number of features; the max number of training epochs is equal to 150 and ReLU as the activation function, with Adam optimizer [61].

Regarding the adaptive multiscale attention, the merge function is one of the hyperparameters of the model, but it is not the only one. There are several other hyperparameters and design choices that can impact the model's performance, such as the choice of activation functions, the depth of the network, the number of neurons in each layer, the weight initialization, the regularization techniques, as well as the learning rate and optimization algorithms.

Anyway, while designing and training a DL model, it is possible to keep all the hyperparameters and design choices fixed, instead of tuning them. In this scenario, after a lot of trial and error, we chose to adopt a fixed architecture. The selected architecture achieved the lowest generalization error on different tasks. Thus, in the selected architecture, we fixed the number of layers, number of neurons in each layer, and activation functions. Also, weight initialization methods, learning rate, optimization algorithm, and batch size are kept fixed. These trial and error were performed on the large click through rate problem using 30% of the dataset as a statistically significant portion of the dataset. Additionally, it allowed us to change the hyperparameters of the network in a timely manner. Once the hyperparameters are found, they are kept fixed for all tests and all datasets. The hyperparameters are the following.

The used weight initializations are He initialization [62] for excitation layers and a matrix of zeros for the TrA Layer. The learning rate starts at 0.01 and decays exponentially by a factor of 0.9 every 50 steps. The batch size is 128 and the number of epochs is 150.

VI. Results and Discussion

All tests were run on exactly the same conditions and data using tenfold cross-validation. The F1-weighted score has been used to summarize the results because the majority of considered datasets are strongly unbalanced. F1-score is the harmonic mean between precision and recall; moreover, the weighted F1-score gives balanced attention (weight) to classes represented by a lower number of instances.

The F1-weighted score applied to classification problems is represented in Table II; the results are computed by averaging F1-weighted scores over ten different runs. It can be observed that the adaptive multiscale attention with the "Add" operator as well as with the "Hadamard Multiplicator" operator are among the top-performing algorithms. In particular, the adaptive multiscale attention with the "Add" operator outperforms all the other algorithms at four small classification problems: the Diabetic Retinopathy, Arrhythmia dataset, the Winsconsin Breast Cancer dataset, and Cervical Cancer dataset. Instead, the adaptive multiscale attention with the "Hadamard Multiplicator" operator outperforms all the other algorithms on three datasets, one small, the University of California, Irvine (UCI) heart disease, and two large datasets, namely, the click-through rate and the Higgs boson datasets. On average, the next most performing algorithms are the CatBoost and the XGBoost ones, with a top performance at two tasks out of 7.

Random forests exhibit an average performance, while SVM RBF, decision tree, MLP, and TabNet perform below the average in all datasets. The standard deviation of F1-scores in Table II implies that TabNet has the highest standard deviation of scores, whereas adaptive multiscale attention mechanisms generally deliver very stable results with respect to different runs of the tenfold cross-validation. This is an important result achieved by the proposed architecture because neural networks are known to be very sensitive to randomness both with data and with initialization conditions, thus delivering different results for each run. The standard deviation in Table II shows that the adaptive multiscale attention (especially the one with "Hadamard Multiplicator" operator) seems to converge to a stable solution with a sufficient number of runs, despite the randomness in the data and the layer's initializations. In conclusion, adaptive multiscale attention with "Hadamard Multiplicator" operator seems to be a good solution for both small, medium, and large datasets on classification problems. This is achieved with a significant degree of generalization with respect to the size of the datasets, dispelling the false myth of DL requiring a large amount of data. Indeed, the goal of DL is to find correlations among features, with those correlations being usually found when large amounts of data are provided. In this case, however, correlations are evaluated by the mechanisms of parallel attention, which intuitively encodes the feature selection phase. This phase is absent in almost all the other algorithms, apart from TabNet. The automatic-weighted feature selection phase (included in the proposed architecture thanks to the end-to-end training) increases the accuracies and ensures a generalization power with respect to different dataset sizes.

Concerning the regression problems with results presented in Table III, it is possible to observe that the adaptive multiscale attention with "Concatenation" operator and with "Average" operator as merging layer are among the best-performing algorithms. Even in this case, the algorithms were robust with respect to different dataset sizes, thus confirming the intuition of the feature weighting encoding.

TABLE II

F1-Weighted Average Scores (± Standard Deviation)

	SVM RBF	**DT**	**MLP**	**Random Forest**	**XGBoost**	**TabNet**	**CatBoost**	**Adaptive Multiscale Attention - Add**	**Adaptive Multiscale Attention - Avg**	**Adaptive Multiscale Attention - Mul**	**Adaptive Multiscale Attention - Conc**
Arrhythmia [45]	0.48599 ± 0.09	0.64570 ± 0.06	0.63356 ± 0.06	0.66499± 0.03	0.67535± 0.03	0.39195± 0.09	0.66357± 0.05	0.67905± 0.04	0.61894± 0.03	0.64802± 0.004	0.62969± 0.04
Winsconsin Breast Cancer [46]	0.97018 ± 0.01	0.90178 ± 0.03	0.97071± 0.01	0.94755± 0.02	0.97011± 0.008	0.93484± 0.05	0.97016± 0.02	0.97023 ± 0.004	0.97023± 0.01	0.96854± 0.01	0.96682± 0.01
UCI Cervical Cancer [47]	0.99069 ± 0.01	0.99381± 0.004	0.98899± 0.008	0.98683± 0.01	0.99106± 0.01	0.99010± 0.01	0.98708± 0.009	0.99647± 0.004	0.99518± 0.005	0.99518± 0.005	0.99518± 0.005
Diabetic Retinopathy Deb. [48]	0.70449 ± 0.02	0.61068± 0.02	0.70550± 0.02	0.68483± 0.03	0.69405± 0.001	0.61713± 0.05	0.68213± 0.02	0.74006± 0.02	0.74392± 0.01	0.72367± 0.01	0.72326± 0.02
UCI Heart Disease [49]	0.52106 ± 0.06	0.49019± 0.05	0.50300± 0.07	0.52480± 0.08	0.50715± 0.05	0.46409± 0.05	0.53946± 0.05	0.53295± 0.05	0.53463± 0.05	0.58480± 0.05	0.49724± 0.05
Click-Through Rate [50]	0.52106 ± 0.06	0.49951± 0.03	0.48673± 0.05	0.53644± 0.06	0.50715± 0.05	0.46409± 0.05	0.53946± 0.05	0.55016± 0.05	0.55128± 0.05	0.60882± 0.04	0.55763± 0.05
Higgs Boson [51]	0.76422 ± 0.001	0.71612± 0.001	0.76117± 0.001	0.74141± 0.001	0.77361± 0.001	0.76280± 0.001	0.76933± 0.001	0.76945± 0.001	0.76525± 0.001	0.77520± 0.0005	0.76569± 0.001

TABLE III

MAE Average Scores (± Standard Deviation)

	SVM RBF	**DT**	**MLP**	**Random Forest**	**XGBoost**	**TabNet**	**CatBoost**	**Adaptive Multiscale Attention - Add**	**Adaptive Multiscale Attention - Avg**	**Adaptive Multiscale Attention - Mul**	**Adaptive Multiscale Attention - Conc**
Echocardiogram [52]	12.03620± 1.51	12.74058 ± 2.28	21.01464 ± 3.01	9.81510 ± 1.31	10.33023 ± 0.42	12.06987 ± 1.81	9.53763 ± 1.68	10.89667 ± 1.75	9.57553 ± 1.34	10.17631 ± 1.31	9.34173 ± 1.55
Bike sharing hourly [53]	37.86235± 13.52	2.77146 ± 1.26	10.12176 ± 6.05	1.31650 ± 0.71	2.14329 ± 0.76	3.69334 ± 0.88	2.97641 ± 1.04	1.83980 ± 1.87	0.98250 ± 1.68	0.37032 ± 1.76	1.57568 ± 1.37
Year Prediction Million Song [54]	OOM	9.18889 ± 0.05	6.76758 ± 0.14	6.60425 ± 0.06	6.34614 ± 0.05	6.48950 ± 0.11	6.36503 ± 0.06	6.42590 ± 0.13	6.37862 ± 0.12	6.32562 ± 0.09	6.27047 ± 0.15
Yahoo Learning to Rank [55]	OOM	9.17654 ± 0.05	6.76244 ± 0.08	6.60131 ± 0.06	6.34614 ± 0.05	6.48950 ± 0.11	6.36503 ± 0.06	6.31205 ± 0.07	6.36500 ± 0.06	6.45027 ± 0.07	6.25441 ± 0.08

Considering the related standard deviations, as reported in Table III, it is possible to observe that adaptive multiscale attention algorithms have higher standard deviations for regression problems. Thus, they end up with fewer stable results if compared to those from the classification problems. Anyway, the standard deviation is in line with the other algorithms.

In order to stress the generalization capabilities of the proposed approach, two different solutions have been identified. The first solution makes use of the MMD [19]. MMD is a distance metric that measures the discrepancy between two probability distributions. In this case, MMD has been used to compare the distributions of the training and testing sets of each dataset.

In practice, MMD is calculated by computing the distance between the means of the feature representations of the training and testing sets in a reproducing kernel Hilbert space (RKHS). The feature representations are obtained by passing the data through a kernel function that maps the data into a higher dimensional space, where linear separation is possible.

MMD can be used to detect distribution shift between the training and testing sets. By measuring the discrepancy between the distributions of the training and testing sets, MMD can provide insights into the nature of the shift and help improve the generalization performance of the model. Figs. A and B in Appendix A shows the MMD value and the ranking (based on the F1-score) from bottom (first ranked high F1-score) to lowest ranked (top) for all classification problems (Fig. A) and all regression problems (Fig. B).

These values are averaged because tenfold cross-validation was performed.

As it is possible to observe from Fig. A in Appendix A, the adaptive multiscale attention is the set of techniques achieving the largest F1-score with higher MMD value for classification problems and lower mean absolute error (MAE) with high MMD for regression problems as in Fig. B. As a confirm, in a second solution aimed at demonstrating the generalization capabilities of the proposed approach, it has been used 2-D multidimensional scaling [63], to reduce the dimensionality of training and test sets of each problem. Then, kernel density estimation [64] has been applied independently on each problem train and test sets. This procedure has been repeated several times and its results analyzed. The plotted results are shown in Appendix B. Analyzing the number of distributions, their geometry, and orientation in the space in problems such as UCI heart dataset, click-through rate, and bike sharing, it was found a different number of modes between train and test data distribution between train and test sets; very different distribution' geometry; and for diabetic retinopathy and heart disease even different orientation in space. For these problems, the adaptive multiscale attention outperformed other techniques, such as XGBoost, while producing comparable results on others that do not specifically exhibit visible differences in data distributions. These findings are of paramount attention because they show that, for difficult problems with quantitatively different data distributions between train and test sets, the adaptive multiscale attention generalizes better on unknown distributions. Table IV reports the training time of the different approaches. Infinity means that after ten days of training, no result was provided, and thus, the algorithm training was canceled. The adaptive multiscale attention approaches always have a larger training time with respect to all other algorithms on small problems, but when it comes to large and very large datasets, SVM with RBF, as well as TabNet have a much larger training time. In fact, SVM was unable to converge on Higgs Boson dataset, and on the same dataset, TabNet took over the double of computation time with respect to adaptive multiscale attention. Thus, adaptive multiscale attention can be useful for creating very accurate interpretable models on very large datasets. Additionally, adaptive multiscale attention can leverage the execution time expense with generalization capabilities and higher accuracies on average.

For ablation studies, Tables V and VI show, respectively, the F1 average score and MAE of the architecture of adaptive multiscale attention with and without excitation layers and without TrA Layer. Tables V and VI show that for the classification problems, there is a sensible decrease in accuracies when the trainable attention is removed as well as excitation layers. When both are removed, the accuracies drop significantly. This means that the synergic integration of excitation layers with the trainable attention led to the jump in accuracies. This result is much more visible in regression problems, where on certain very large problems such as Yahoo learning to rank, removing both trainable attention and excitation layer resulted in generating a much larger error in magnitude than the proposed architecture. Even just removing the excitation layers resulted in a decrease in accuracy for classification problems and an increase in error for regression problems. One of the limitations of this study is the number of variations of the proposed architecture, especially for what concerns the merging layers. This is a limitation but it allows the scientific community to investigate and find novel merging solutions that may even perform better than actual solutions.

As represented in Tables II and III, adaptive multiscale attention is not the best algorithm for all the datasets. In facts, different algorithms happen to perform best on different datasets. Adaptive multiscale attention, in general, shows interesting generalizations and explainable properties that, in our opinion, should be looked for when choosing which algorithm to apply.

VII. Interpretability of Results

A. *Interpretability*

The discussion reported in this section is only referred to experiments performed on the Winsconsin Breast Cancer dataset for the sake of readability. According to the interpretability of elements that the excitation layer considers important as the input changes, Fig. 3 shows the sum over all excitation layers (with varying compression ratio per excitation layers) of the learned weights associated with each feature with respect to the predicted class (refer to excitation layer in Fig. 1) as in (9). Here, it is possible to observe that the feature symmetry_mean is the most discriminant for classifying breast cancer as malignant. While compactness_se has a relative high importance in classifying the instance as benignant. The global feature attention, evaluated as in (10), is reported in pictorial terms in Fig. 4. It can be observed that as previously stated "symmetry_mean" is the most important feature while the *identifier* of each instance is the least relevant feature as it could be easily hypothesized.

Concerning the feature attention behavior with respect to the class, refer (11), its pictorial representation is reported in Fig. 5 which shows how the network behaves in direction and magnitude of attention weights with respect to the input and the respective class. This allows understanding, for each class, what are the most important features and their attention weights direction.

It is interesting to see how sensitive the model is with respect to the prediction of malignant class on the "texture_mean" feature as well as "symmetry_mean."

The information provided from an attention weights point of view is the same as in Fig. 4, but it adds the direction of how the weights W_{lz} behaves with respect to the current instance J_i. Thanks to this novel information, it is possible to visualize features that tend to co-behave.

Concerning the last analysis, related to the inspection of the nonlinear correlation among features that tend to co-behave in both the same and the opposite direction, the resulting correlation matrix is presented in (13), and its pictorial representation is shown in Fig. 6. In the specific case of the Wisconsin Breast Cancer dataset, it can be observed that "area_mean" and "perimeter_worst" features tend to co-behave: they have a large correlation coefficient, as well as "radius_worst" and "perimeter_worst."

B. *Example of the Use of Explainability Results*

Explainability can be useful not only to understand the behavior of the model but also to refine results and to engineer new systems. A simple example is the one related to the feature selection process that can be built upon the feature ranking provided by the adaptive multiscale attention network (see Fig. 4). In this case, the performance provided by these features can be compared with those provided by other standard approaches. To this aim, four different linear SVM models with default parameters have been trained with a tenfold cross-validation technique on the Wisconsin Breast Cancer dataset. Each SVM has been trained with the top three ranked features provided by different methods here considered:

1) adaptive multiscale attention—add: "symmetry_mean," "area_se," and "textrue_worst."
2) random forest: "radius_worst," "concave points_worst," and "perimeter_worst."
3) CatBoost: "Concave points_worst," "area_worst," and "texture_worst."
4) XGBoost: "perimeter_worst," "concave points_worst," and "radius_worst."

Results are reported in Table VII.

Features selected by the adaptive multiscale attention with Add operator can outperform all the others in terms of F1. In addition, its standard deviation (among different runs) is sensibly lower than all the others, which implies higher reliability and robustness.

The top three features selected by adaptive multiscale attention achieved the highest F1-score on this dataset in general, even when compared with F1-scores in Table II, which means that all other features do not contribute to increase the performance. Thus, adaptive multiscale attention was able to select the right top three features that saturate the average reachable accuracy on this dataset.

TABLE IV

AVERAGE EXECUTION TIME (IN S) ON CLASSIFICATION ALGORITHMS

	SVM RBF	DT	MLP	Random Forest	XGBoost	TabNet	CatBoost	Adaptive Multi-Scale Attention - Add	Adaptive Multi-Scale Attention - Avg	Adaptive Multi-Scale Attention - Mul	Adaptive Multi-Scale Attention - Conc
Arrhythmia	0.03	0.03	1.04	0.22	0.53	1.36	5.43	25.13	25.18	25.88	25.30
Winsconsin Breast Cancer	0.01	0.01	0.13	0.16	0.06	1.66	0.31	17.28	17.20	17.16	17.11
Cervical Cancer	0.06	0.03	0.18	0.12	0.60	2.90	1.17	16.90	17.20	17.04	17.24
Diabetic Retinopathy	0.03	0.01	0.21	0.22	0.14	2.91	0.25	16.92	16.84	16.54	16.63
Heart Disease	0.01	0.00	0.08	0.13	0.25	0.68	0.27	15.63	15.67	15.66	15.59
Click-Through Rate	30469	6.37	63.60	149.90	7.02	1324.00	3.90	2261.00	2264.00	2281.00	2305.00
Higgs Boson	infinity	700.00	337.00	8915.0	477.00	55899.00	53.00	26846.00	27210.00	27550.00	27765.00

TABLE V

ABLATION STUDY 1: F1 AVERAGE SCORES (± STANDARD DEVIATION), W MEANS WITH, W/O MEANS WITHOUT, EA MEANS ADAPTIVE MULTISCALE ATTENTION, AND TRA MEANS TRAINABLE ATTENTION

	Architecture W EA and W/O TRA	Architecture W/O EA, W TRA	EA – Add W/O TRA	EA – Average W/O TRA	EA – Multiply W/O TRA	EA – Concatenate W/O TRA
Arrhythmia [45]	0.62154 ± 0.039	0.4407 ± 0.089	0.5922 ± 0.074	0.568± 0.077	0.564± 0.052	0.590± 0.071
Winsconsin Breast Cancer [46]	0.9641 ± 0.011	0.9659 ± 0.004	0.968 ± 0.012	0.9701± 0.008	0.966± 0.010	0.963± 0.006
UCI Cervical Cancer [47]	0.982 ± 0.01	0.9890 ± 0.012	0.9928± 0.007	0.991± 0.008	0.993± 0.007	0.994± 0.005
Diabetic Retinopathy Deb. [48]	0.6595 ± 0.016	0.6807 ± 0.024	0.722± 0.028	0.739± 0.009	0.728± 0.012	0.716 ± 0.025
UCI Heart Disease [49]	0.5203 ± 0.069	0.5379 ± 0.065	0.535± 0.071	0.535± 0.062	0.526± 0.047	0.506 ± 0.081
Click-Through Rate [50]	0.4604 ± 0.001	0.4543 ± 0.001	0.464± 0.001	0.464± 0.001	0.464± 0.001	0.464 ± 0.001
Higgs Boson [51]	0.7184 ± 0.001	0.6922 ± 0.005	0.733± 0.001	0.732± 0.001	0.732± 0.001	0.735 ± 0.001

TABLE VI

ABLATION STUDY 2: MAE AVERAGE SCORES (± STANDARD DEVIATION), W MEANS WITH, W/O MEANS WITHOUT, EA MEANS ADAPTIVE MULTISCALE ATTENTION, AND TRA MEANS TRAINABLE ATTENTION

	Architecture W EA and W/O TRA	Architecture W/O EA, W TRA	EA – Add W/O TRA	EA – Average W/O TRA	EA – Multiply W/O TRA	EA – Concatenate W/O TRA
Echocardiogram [52]	16.042 ± 6.57	12.721 ± 1.61	9.899 ± 2.309	11.431 ± 3.97	14.81 ± 6.991	14.504 ± 5.578
Bike sharing hourly [53]	16.84 ± 13.37	69.34 ± 11.16	10.875± 5.05	9.901 ± 3.65	11.536 ± 5.181	7.248 ± 0.768
Year Prediction Million Song [54]	313.299 ± 8.71	20.97 ± 0.55	26.362 ± 0.26	26.613 ± 0.42	7.1576 ± 0.108	19.654 ± 0.297
Yahoo Learning to Rank [55]	309.58 ± 8.32	21.03 ± 0.48	26.401 ± 0.74	26.422 ± 0.719	8.041 ± 1.255	19.740 ± 0.642

TABLE VII

F1-SCORES OF SVM TRAINED ON TOP THREE FEATURES EXTRACTED FROM DIFFERENT ALGORITHMS

Technique for extracting top 3 Features	F1-Score	Standard Deviation
Adaptive Multi-Scale Attention - Add	**0.975553**	**0.0176**
Random Forest	0.939077	0.0253
CatBoost	0.959500	0.0274
XGBoost	0.940753	0.0243

VIII. CONCLUSION

In this work, the adaptive multiscale attention deep neural network architecture is proposed as a novel technique to be used with tabular data. Adaptive multiscale attention was capable of successfully learning features' attention and thus achieving high levels of F1-scores by exploiting parallel multilevel feature weighting, assembly, and recalibration. These results were achieved for classification tasks on seven different datasets of small, medium, large, and very large sizes. Adaptive multiscale attention also provided low MAE scores on four regression tasks of different sizes. In addition, their computation time is higher with respect to all the other techniques on small and medium-sized datasets, but it tends to be lower than SVM and TabNet on very large datasets.

By using multidimensional scaling for nonlinear dimensionality reduction in conjunction with kernel density estimation, it has been shown that the problems on which adaptive multiscale attention performs better than the other techniques are also the problems that show a noticeable but visible difference in the density distributions between the train and test sets showing, thus, that adaptive multiscale attention is a powerful technique capable of good generalization power. In addition, other benefits are the high accuracy, the low number of hyperparameters to setup, several options of interpretability, no preprocessing, and dimensionality reduction pipeline needed.

In general, adaptive multiscale attention provided high generalization capabilities with respect to the size of the dataset and the type of problem (regression or classification) with a low standard deviation of results from different folds when trained in tenfold validation. This shows the robustness of the proposed algorithm.

In addition, the adaptive multiscale attention here proposed has been able to provide four levels of explainability: 1) interpretability of excitation layers: gives insights into which layers are more pivotal for specific classes by calculating the attention weight of each excitation layer; 2) global magnitude feature attention: reveals the overall attention of each feature across all instances. Computed as the mean of the absolute values obtained from the network; 3) global feature attention concerning class and behavior analysis: provides an understanding of the learned feature attention with respect to each class, allowing a deeper analysis of feature attention and behavior for particular classes; and 4) nonlinear spearman correlation among learned features: identifies nonlinear correlations among features that co-behave, giving valuable information for dataset dimensionality reduction and increasing interpretability. Feature weighting was not computed post hoc; thus, it concretely represented the feature attention given by the network while training. Thanks to these interpretability levels, it is possible to successfully use adaptive multiscale attention as a technique for feature selection and both to increase accuracy and to get important insights about the dataset at hand.

The potential limitations of the study on the adaptive multiscale attention network for tabular data include the following.

1) *Data Availability:* The need for large amounts of data for gradient-based optimization may limit its performance on smaller datasets.
2) *Data Quality Challenges:* Handling missing values, outliers, and inconsistencies in tabular data are still in charge of the researcher.
3) *Computational Complexity:* Despite the sublinear growth, the network may require significant computational resources for training and optimization.
4) *Generalizability:* The network's performance on a wide range of real-world tabular datasets with unique or challenging characteristics remains to be evaluated.
5) *Architecture Variability:* The large number of variations of the proposed architecture may lead to different results.

In future research, adaptive multiscale attention will be applied to other tabular datasets with a focus on health datasets for the purpose of Public Administration application and a novel merging layer can be defined. In addition, its nature of being explainable will be stressed and explored: various other correlations can be applied for analyzing feature co-behavior.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/NATURE14539.

[2] D. Zhang, C. Yin, J. Zeng, X. Yuan, and P. Zhang, "Combining structured and unstructured data for predictive models: A deep learning approach," *BMC Med. Informat. Decis. Making*, vol. 20, no. 1, pp. 1–11, Dec. 2020, doi: 10.1186/S12911-020-01297-6/FIGURES/3.

[3] R. Wannamaker, B. Buck, and K. Butcher, "Multimodal CT in acute stroke," *Current Neurol. Neurosci. Rep.*, vol. 19, no. 9, pp. 1–13, Jul. 2019, doi: 10.1007/s11910-019-0978-z.

[4] S. O. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," 2019, *arXiv:1908.07442*.

[5] H. Luo, F. Cheng, H. Yu, and Y. Yi, "SDTR: Soft decision tree regressor for tabular data," *IEEE Access*, vol. 9, pp. 55999–56011, 2021, doi: 10.1109/ACCESS.2021.3070575.

[6] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *JSTOR*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001. [Online]. Available: https://www.jstor.org/stable/2699986

[8] M. P. Neto and F. V. Paulovich, "Explainable matrix–visualization for global and local interpretability of random forest classification ensembles," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 1427–1437, Feb. 2021, doi: 10.1109/TVCG.2020.3030354.

[9] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2017, pp. 6638–6648.

[10] R. E. Schapire, "Explaining AdaBoost," in *Empirical Inference*, N. V. Vladimir, Ed. Berlin, Germany: Springer, 2013, pp. 37–52, doi: 10.1007/978-3-642-41136-6_5.

[11] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.

[12] T. M. Hehn, J. F. P. Kooij, and F. A. Hamprecht, "End-to-end learning of decision trees and forests," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 997–1011, Apr. 2020, doi: 10.1007/s11263-019-01237-6.

[13] B. Feng, Y. Wang, and Y. Ding, "Uncertainty-aware attention graph neural network for defending adversarial attacks," in *Proc. 35th AAAI Conf. Artif. Intell.*, vol. 8B, Sep. 2020, pp. 7404–7412.

[14] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, Oct. 2017, pp. 1–12, doi: 10.1007/978-3-031-01587-8_7.

[15] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," 2019, *arXiv:1909.06312*.

[16] A. Hernández and J. M. Amigó, "Attention mechanisms and their applications to complex systems," *Entropy*, vol. 23, no. 3, p. 283, Feb. 2021, doi: 10.3390/e23030283.

[17] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 23, 2022, doi: 10.1109/TNNLS.2022.3229161.

[18] S. Chatterjee and P. Zielinski, "On the generalization mystery in deep learning," 2022, *arXiv:2203.10036*.

[19] A. Gretton, K. M. Borgwardt, M. Rasch, and A. J. Smola, "A kernel method for the two-sample problem," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, May 2008, pp. 513–520, doi: 10.7551/mitpress/7503.003.0069.

[20] A. Pérez, P. Larra naga, and I. Inza, "Bayesian classifiers based on kernel density estimation: Flexible classifiers," *Int. J. Approx. Reason.*, vol. 50, no. 2, pp. 341–362, 2009, doi: 10.1016/J.IJAR.2008.08.008.

[21] H. Wang, D. Mirota, and G. D. Hager, "A generalized kernel consensus-based robust estimator," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 178–184, Jan. 2010, doi: 10.1109/TPAMI.2009.148.

[22] E. Schubert, A. Zimek, and H.-P. Kriegel, "Generalized outlier detection with flexible kernel density estimates," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2014.

[23] B. Neyshabur et al., "Exploring generalization in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

[24] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[25] Z. Li, Y. Sun, L. Zhang, and J. Tang, "CTNet: Context-based tandem network for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9904–9917, Dec. 2022, doi: 10.1109/TPAMI.2021.3132068.

[26] Z. Li, J. Tang, and X. He, "Robust structured nonnegative matrix factorization for image representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1947–1960, May 2018, doi: 10.1109/TNNLS.2017.2691725.

[27] H. Zhang et al., "Multiscale visual-attribute co-attention for zero-shot image recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6003–6014, Sep. 2023, doi: 10.1109/TNNLS.2021.3132366.

[28] H.-C. Li, W.-S. Hu, W. Li, J. Li, Q. Du, and A. Plaza, "A3 CLNN: Spatial, spectral and multiscale attention ConvLSTM neural network for multisource remote sensing data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 747–761, Feb. 2022, doi: 10.1109/TNNLS.2020.3028945.

[29] M. Shi et al., "LMFFNet: A well-balanced lightweight network for fast and accurate semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3205–3219, Jun. 2023, doi: 10.1109/TNNLS.2022.3176493.

[30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[31] L. E. Melkumova and S. Y. Shatskikh, "Comparing ridge and LASSO estimators for data analysis," *Proc. Eng.*, vol. 201, pp. 746–755, Jan. 2017.

[32] B. Peters, V. Niculae, and A. F. T. Martins, "Sparse sequence-to-sequence models," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, May 2019, pp. 1504–1519, doi: 10.18653/v1/p19-1146.

[33] A. Abutbul, G. Elidan, L. Katzir, and R. El-Yaniv, "DNF-Net: A neural architecture for tabular data," 2020, *arXiv:2006.06465*.

[34] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421, doi: 10.18653/v1/d15-1166.

[35] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Jan. 2016, pp. 551–561, doi: 10.18653/v1/d16-1053.

[36] A. Vaswani, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 5999–6009.

[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, Sep. 2018, pp. 3–19.

[38] S. Xu, Y. Cheng, K. Gu, Y. Yang, S. Chang, and P. Zhou, "Jointly attentive spatial–temporal pooling networks for video-based person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4743–4752, doi: 10.1109/ICCV.2017.507.

[39] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–14.

[40] A. Shah, E. Kadam, H. Shah, S. Shinde, and S. Shingade, "Deep residual networks with exponential linear unit," in *Proc. ACM Int. Conf. Ser.*, vols. 21–24, Sep. 2016, pp. 59–65, doi: 10.1145/2983402.2983406.

[41] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[42] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Dec. 2013, pp. 1–11.

[43] A. Courville, D. Erhan, Y. Bengio, and P. Vincent. (2009). *Visualizing Higher-Layer Features of a Deep Network Visualizing Higher-Layer Features of a Deep Network Département d'Informatique et Recherche Opérationnelle*. Accessed: Oct. 14, 2022. [Online]. Available: https://www.researchgate.net/publication/265022827

[44] H. A. Guvenir, B. Acar, G. Demiroz, and A. Cekin, "A supervised machine learning algorithm for arrhythmia analysis," in *Proc. Comput. Cardiol.*, Sep. 1997, pp. 433–436, doi: 10.1109/CIC.1997.647926.

[45] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," *Proc. SPIE*, vol. 1905, pp. 861–870, Jul. 1993, doi: 10.1117/12.148698.

[46] . Sobar, R. Machmud, and A. Wijaya, "Behavior determinant based cervical cancer early detection with machine learning algorithm," *Adv. Sci. Lett.*, vol. 22, no. 10, pp. 3120–3123, Oct. 2016, doi: 10.1166/asl.2016.7980.

[47] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowl.-Based Syst.*, vol. 60, pp. 20–27, Apr. 2014, doi: 10.1016/j.knosys.2013.12.023.

[48] R. Detrano et al., "International application of a new probability algorithm for the diagnosis of coronary artery disease," *Amer. J. Cardiology*, vol. 64, no. 5, pp. 304–310, Aug. 1989, doi: 10.1016/0002-9149(89)90524-9.

[49] *SIGKDD: KDD Cup 2012 (Track 2): Predict the Click-Through Rate of Ads Given the Query and User Information*. Accessed: Nov. 9, 2021. [Online]. Available: https://www.kdd.org/kdd-cup/view/kdd-cup-2012-track-2

[50] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Commun.*, vol. 5, no. 1, p. 4308, Jul. 2014, doi: 10.1038/ncomms5308.

[51] L. Chen, "Curse of Dimensionality," in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. Boston, MA, USA: Springer, 2009, pp. 545–546, doi: 10.1007/978-0-387-39940-9_133.

[52] R. Artusi, P. Verderio, and E. Marubini, "Bravais-pearson and Spearman correlation coefficients: Meaning, test of hypothesis and confidence interval," *Int. J. Biol. Markers*, vol. 17, no. 2, pp. 148–151, Apr. 2002, doi: 10.1177/172460080201700213.

[53] S. Salzberg, *Exemplar-Based Learning: Theory and Implementation*. Cambridge, ma, usa: Harvard University Center for Research in Computing Technology Aiken Computation Laboratory, 1988.

[54] F.-T. Hadi and J. Gama, "Event labeling combining ensemble detectors and background knowledge," *Progr. Artif. Intell.*, vol. 2, pp. 113–127, Jun. 2014, doi: 10.1007/S13748-013-0040-3/TABLES/7.

[55] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere. *The Million Song Dataset*. Accessed: Nov. 11, 2021. [Online]. Available: http://www.infochimps.com/

[56] B. M. Marlin and R. S. Zemel, "Collaborative prediction and ranking with non-random missing data," in *Proc. 3rd ACM Conf. Recommender Syst.*, Oct. 2009, pp. 5–12, doi: 10.1145/1639714.1639717.

[57] R. R. Bouckaert and E. Frank, "Evaluating the replicability of significance tests for comparing learning algorithms," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2004, pp. 3–12.

[58] X. Chai, L. Deng, Q. Yang, and C. X. Ling, "Test-cost sensitive naive Bayes classification," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 51–58.

[59] G. Brown, "Diversity in neural network ensembles," Ph.D. dissertation, School Comput. Sci., Univ. Birmingham, 2004.

[60] K. Huang et al., "Biased minimax probability machine for medical diagnosis," in *Proc. AI&M*, 2004, pp. 1–8.

[61] S. Bock and M. Weiß, "A proof of local convergence for the Adam optimizer," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8852239.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[63] I. Borg and P. Groenen, "Modern multidimensional scaling: Theory and applications," *J. Educ. Meas.*, vol. 40, no. 3, pp. 277–280, 2003, doi: 10.1111/J.1745-3984.2003.TB01108.X.

[64] J. Kim and C. D. Scott, "Robust kernel density estimation," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2529–2565, Sep. 2012. [Online]. Available: www.eecs.umich.edu/