IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 September 2024, accepted 27 September 2024, date of publication 3 October 2024, date of current version 16 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3472872

SURVEY

Explainable Artificial Intelligence (XAI) for Methods Working on Point Cloud Data: A Survey

RAJU NINGAPPA MULAWADE[1], CHRISTOPH GARTH[2], AND ALEXANDER WIEBEL[1]
[1]Department of Computer Science, Hochschule Worms University of Applied Sciences, 67549 Worms, Germany
[2]Department of Computer Science, RPTU Kaiserslautern-Landau, 67663 Kaiserslautern, Germany

Corresponding author: Raju Ningappa Mulawade (mulawade@hs-worms.de)

This work was supported in part by the German Federal State of Rhineland-Palatinate (Forschungskolleg SIVERT); and in part by Hochschule Worms, Germany.

ABSTRACT In this work, we provide an overview of the XAI (Explainable Artificial Intelligence) works related to explaining the methods working on point cloud (PC) data. The recent decade has seen a surge in artificial intelligence (AI) and machine learning (ML) algorithms finding applications in various fields dealing with a wide variety of data types such as image and text data. Point cloud data is one of these datatypes that has seen an upward trend in the use of AI/ML algorithms. However, not all these AI algorithms are ''white box'' models that can be understood by humans easily. Many of them are hard to interpret or understand and thus, require methods to provide explanations for the decision-making process. These methods that attempt to provide explanations or insights into the working of AI models working on various datatypes are grouped under XAI. Even though the use of datatypes such as point clouds for AI models has seen an upward trajectory, we see a lack of survey works documenting the developments in the corresponding XAI field. This issue is addressed through our contribution. We classify the literature based on different criteria such as XAI mechanism used, AI models, their tasks, type of model learning and the type of point cloud data taken into consideration. This can help readers identify works that address specific tasks and obtain corresponding details easily. We also provide useful insights regarding the surveyed papers that highlight interesting aspects of the surveyed literature.

INDEX TERMS Artificial intelligence, explainable AI, point cloud data.

I. INTRODUCTION

In recent years, the technological advancement has made 3D data acquisition more accessible by making equipment such as sensors and cameras used for such data acquisition more affordable. This has led to an increase in the use of 3D data in various fields of application such as autonomous driving [16], healthcare [31], cultural heritage [57], and virtual reality [70]. Point clouds are one of the 3D datatypes represented by a discrete set of data points in 3D with each point represented by its spatial coordinates (x, y, z). They are finding widespread use in many applications due to their ability to capture structural information in 3D.

Apart from finding use in an increasing number of domains, point clouds recently have been increasingly processed and analyzed using machine learning (ML) which has also seen an exponential growth in the last decade. ML networks have the ability to capture complex features and perform tasks on various datatypes such as images, texts and time series with high accuracy. These properties of ML networks have led to the possibility of utilizing them for a variety of tasks in many areas of application. A major part of the ML research is related to ML networks working on image data. The amount of research work in the field of point-cloud-based ML is significantly less compared to image and text-based research work. This is probably attributed to the unstructured nature of the point cloud data which makes learning features extremely difficult compared

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

to structured data such as images. Another factor impacting the development of point-cloud-based learning algorithms is the scarcity of data as many ML algorithms are known to be ''data hungry''. Many efforts have been made in the last decade to address this [10], [76]. A brief overlook of point cloud datasets is provided by Camuffo et al. in [7].

Availability of large-scale point cloud datasets such as ModelNet [76] and ShapeNet [10] has enhanced the developmental works in ML intended for point cloud data [17]. With this increase in the number of learning algorithms working on point clouds, the need for understanding these algorithms has also become more important. This is mainly due to the risks involved in the processes that employ AI algorithms. For example, in autonomous driving applications or in applications in the healthcare field, minute errors by the AI algorithms could lead to the loss of or pose major danger to human life. Thus, it becomes extremely important to understand these AI algorithms before deploying them in real-world scenarios. This is addressed by XAI. XAI is a subfield of AI, and especially of ML, that corresponds to the development of methods attempting to provide explanations of AI and ML model's working. It is an important field of research given the exponential rise in AI and ML and its penetration into the application areas involving risks, as those mentioned above. Figure. 1 shows an overview of the XAI pipeline in point-cloud-based AI. The XAI methods make use of one or a combination of the three components (input point cloud data, AI model and prediction) to provide users with useful insights into the working of AI models.

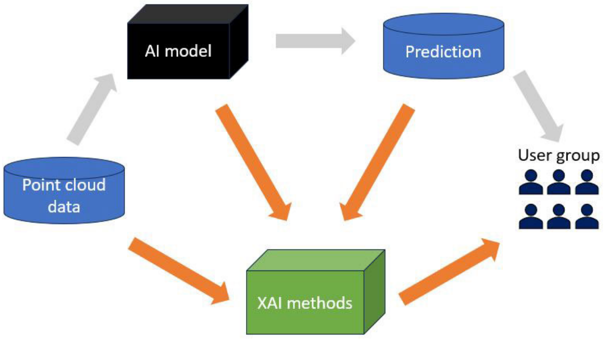


FIGURE 1. Illustration of XAI in point-cloud-based AI. Gray arrows indicate the general AI/ML process and the orange arrows indicate XAI process.

Even though there are some prominent point-cloud-based XAI works that attempt to explain or improve the transparency of ML models such as [80] and [82], the research area shows the absence of survey works documenting the latest developments in the field. Survey papers provide important information regarding the significant works and the latest developments in the corresponding field of research. In addition, they also highlight the pattern in the research literature collected and identify the gaps that need to be addressed by the researchers in the future [74]. Thus, they play an important role in every field of research. Even though the XAI field has survey works that cater to specific sub-fields such as cybersecurity related XAI [11], time series data-based XAI [48], and healthcare related XAI [9], [68], we observe the absence of such works in the point cloud-based XAI field. Therefore, we attempt to provide an overview of the XAI work in point-cloud-based AI models. This will allow the researchers working on point cloud data processing to easily find brief but important details about the past works and the current state of the art developments in the field and thus, help them in developing new methodologies or extending the current work. We review 45 papers that were selected from 81 papers collected at the beginning of the survey. The contributions of our paper are:

1) Detailed overview of XAI for point cloud data.
2) Classification of the literature based on various criteria to provide better insights.
3) Explanation of the basic XAI mechanisms employed or adapted by the surveyed papers.
4) Provide interesting insights from the surveyed literature that help readers understand the XAI work for point cloud data better.

The above mentioned contributions intend to help readers coming from various fields such as medical professionals looking for reliable AI applications for certain tasks in their domain. For example, they can be useful for ML developers trying to look for XAI methods that provide explanations for a specific type of point cloud data or ML model that relates to their work or for the readers from the visualization domain investigating the use of visualization tools in XAI for point cloud data.

A. ORGANIZATION OF THIS PAPER

The rest of the paper is organized into following sections: Section II provides an overview of the surveyed literature in the field of XAI in general and indicates the need for our survey paper. Section III provides a brief information about the methodology used for collecting and shortlisting of the literature. Section IV explains the classification of the collected literature based on the type of XAI explanation and the properties of the corresponding AI model and data used. Section V provides a detailed information about the XAI methods proposed in the selected literature and section VI provides interesting insights into the surveyed literature. Section VII contains the final remarks regarding our contribution.

II. RELATED WORK

With the rapid increase in the use of AI algorithms to perform diverse tasks, the need to understand these algorithms has also gained importance in recent years. This area is addressed by XAI methodologies providing insights into the working of these AI algorithms. These insights also lead to the increase in the trust value of these AI models among the end-users. Many XAI methods have been developed in the last decade that attempt to provide an explanation of the AI model's working. These are documented in many XAI surveys published in recent years. Adadi and Berrada [1] thoroughly reviewed explainability methods in AI. They explained in details the basic need for XAI and the classification of the literature based on (a) the complexity of ML model, (b) the scope of interpretability, and (c) the level of dependency on the ML model taken into consideration. Minh et al. [39] provided a detailed survey of XAI literature explaining the background of XAI and reviewing the XAI literature included in the survey. Similarly, Linardatos et al. [30] surveyed ML interpretability methods and provided sources of the programming implementations of these methods. The classification of the XAI literature in this work was based on four criteria: (a) Scope of the interpretability, (b) the level of model dependency, (c) purposes of interpretability, and (d) data types.

However, it is extremely difficult to provide a survey encompassing all the research work done in the field of XAI. This is mainly due to the diverse nature of AI models, input data, and model tasks. Thus, many surveys focus on providing an insight into the recent developments in specific sub-regions of the XAI field.

For example, Burkart and Huber [6] provided a survey of XAI methods intended for supervised ML which is a type of learning used in ML to train a model. They focused on XAI works corresponding to the classification and regression models of supervised learning domain. However, they do not take into consideration the type of input data being used for these ML models. Danilevsky et al. [12] focused on providing an overview of the XAI work in the field of natural language processing (NLP). The authors classified the literature into local vs global explanations, self-explaining NLP models vs post-hoc methods and provided detailed explanations for the same. Charmet et al. [11] provided an extensive literature review of the XAI methods intended for the cybersecurity field. Their classification of literature involved the exploration of both methods for explaining AI algorithms used for cybersecurity applications and the security analysis of XAI methods. Tjoa and Guan [68] surveyed the XAI literature in the medical field. They presented two major categories (perceptive interpretability and interpretability by mathematical structure) for the XAI literature which are further divided into subgroups based on the XAI mechanism used. Chaddad et al. [9] provided a detailed survey of XAI developments in the healthcare field. Here, the surveyed papers are categorized into four major groups based on the forms of explanations, types of interpretability, model dependency and scope of the explanations. The authors also take into consideration the modalities of the data acquisition when exploring the XAI methods. Similarly, Wells and Bednarz [75] surveyed XAI literature that attempt to explain reinforcement learning (RL) models. They classified the XAI literature for RL based on four main topics: subject domain the papers focused on, publication types such as conference or journals, year of publication, and the primary purpose of the XAI papers. Di Martino et al. [13] focused on surveying tabular and time series data-based XAI literature in the field of clinical and remote health applications. The collected papers are grouped into three main categories based on the input data type, model development stage and explanation assessments. These are further divided into subgroups to help users identify research papers based on their properties. Another XAI survey focusing on the explainability of AI algorithms working on time series data was by Rojat et al. [48]. The authors classified the literature into multiple groups based on the properties of XAI methods such as the scope of interpretability, ante-hoc vs post-hoc, model dependency, target audience, and whether the papers include evaluation of the explanations proposed.

However, to the best of our knowledge, there are no survey articles that focus on point cloud related XAI works. Therefore, through this survey, we attempt to close this gap in the literature.

III. SURVEY METHODOLOGY

We collected literature based on the *keyword* search and followed the references in the found literature by the snowballing approach. In the the keyword search process, we used the combination of keywords "XAI", "explainability", "interpretability" and "saliency map" with the keyword "point cloud" to obtain corresponding literature. We considered the ACM Digital Library, IEEE Explore and SpringerLink online databases along with the use of Google Scholar web search engine to look for relevant literature. We obtained 74 papers after this process. With the obtained papers as *start set* we used the snowballing approach to incrementally look for relevant references in the collected literature and for papers which reference the collected literature. This process resulted in a collection of 81 papers appearing to be relevant for the topic studies in this survey.

To filter the actually relevant papers from the collected literature, we followed the four-phase flow diagram of PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses) guideline proposed by Liberati et al. [29]. In addition to the papers that focus primarily on addressing the explainability part of the AI models, this filtering process also included the selection of papers that do not directly try to address the explainability issue but include concepts/methods that can be used for understanding the working of these AI models. We did not restrict our literature search to a specific time period as we intended to cover all the research work carried out in this specific field. This filtering process resulted in 45 papers for our analysis. Figure 2 shows the distribution of the selected literature over the years they were published. We observe an overall increasing trend which is promising for the research field in focus with 10 papers already published in the first five months of 2024.

IV. CLASSIFICATION

XAI methods consist of various types of explanations that attempt to explain AI models in different ways. In other words, XAI methods do not follow a standard way of

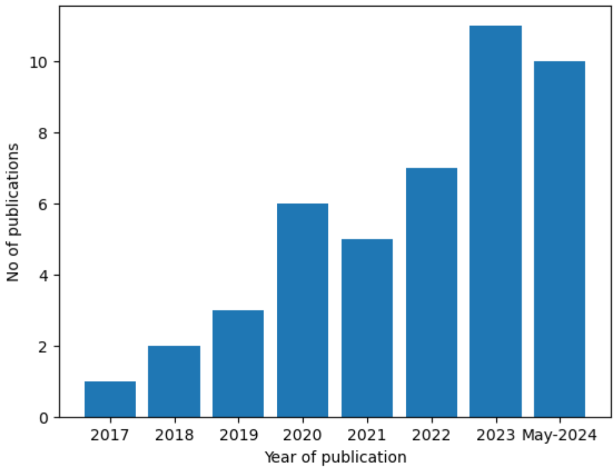


FIGURE 2. Number of articles published in point-cloud-based XAI over years. In general, numbers are increasing. There is a dip around 2021 which might be attributed to COVID-19. A decrease of non-COVID-related publications has been observed in some research fields [33], [46].

analyzing a model. Therefore, it is important for surveys to provide a classification of these methods based on their features. The point-cloud-based XAI methods collected in this work, in general, can be classified into multiple classes and sub-classes based on various criteria (see Figure 3). These criteria include the datatype being used as input, properties of the AI model used, and the characteristics of the XAI methods used. In this work, we classify these methods as follows:

A. TYPES OF EXPLAINABILITY

In this section, we classify the XAI literature based on the types of explainability (Ⓐ in Figure 3) they use to explain the working of the AI model. Here, we also include the research works that attempt to make these models partially or completely transparent (or intrinsic) and thus, addressing the explainability aspect of these models (see Table 1). Further, we identify which of these works include visualization as a part of the model analysis.

1) TRANSPARENT MODELS

This group corresponds to the explainability derived from the architecture of the AI models itself. In particular, the architecture of the model contains layers that are easily interpretable for humans. We further divide this group into two subgroups based on the extent of transparency.

Intrinsic models:These are the AI models that are completely transparent. This means, the model's architecture is inherently interpretable. Some examples of intrinsic models are decision trees, K-Nearest neighbors (k-NN) and linear regression models.

Hybrid interpretable models: These are the black box models that include some interpretable layers making them partially transparent.

2) XAI MECHANISMS AND POINT OF APPLICATION

The XAI methods are used to analyze models at different stages of of ML pipeline. In this work, the collected literature can be classified into two classes based on the point of application of XAI methods.

1) Training process analysis: Methods belonging to this class are utilized to analyze the model during the training process. These methods help humans in understanding how the model learns to detect features as the training process progresses.
2) Post-hoc: Post-hoc XAI methods refer to the group of methods that are used to explain a model after the completion of training process. These methods are further divided into following sub-groups:
   a) *Gradient-based*: These are the methods that utilize the gradients to produce saliency attributions to understand the decision-making process of the AI model.
   b) *Perturbation-based*: These methods introduce perturbations to the input point cloud data and compute the change in output value as the saliency attributions corresponding to the input variables altered.
   c) *Comparison of latent features*: Here, the features of input data instances are compared in the latent space to understand how the model differentiates input instances belonging to different classes.
   d) *Activation of intermediate layers*: Explanations that exclusively use the activations of the intermediate layers to generate saliency maps are categorized into this group.
   e) *Example-based explanation*: One of the unique methods of analyzing a model's performance is to compare multiple input data instances and interpret the decision-making process of the model. Such comparisons provide humans with the possibility to observe distinct patterns in the input data instances that relate to the output value.

When we try to use explainability methods to understand the working of an AI or ML model, it is important to answer some of the basic questions such as "Will the method provide me an overall explanation about the model's decision-making or will it be specific to an individual input instance?", "How easy is it to use the method to explain different AI models?" and "What tools does it require for explaining the model?". The answers to these basic but very important questions are provided by the properties of the explainability methods. Therefore, we also classified the surveyed papers based on these properties that help readers with these information.

3) SCOPE OF EXPLANATION

There are two types of explanations through which the models are analyzed. The global explanation, on one hand, attempts to explain the model by taking into consideration multiple input instances and providing an overall interpretation of the model behaviour. On the other hand, the local explanation

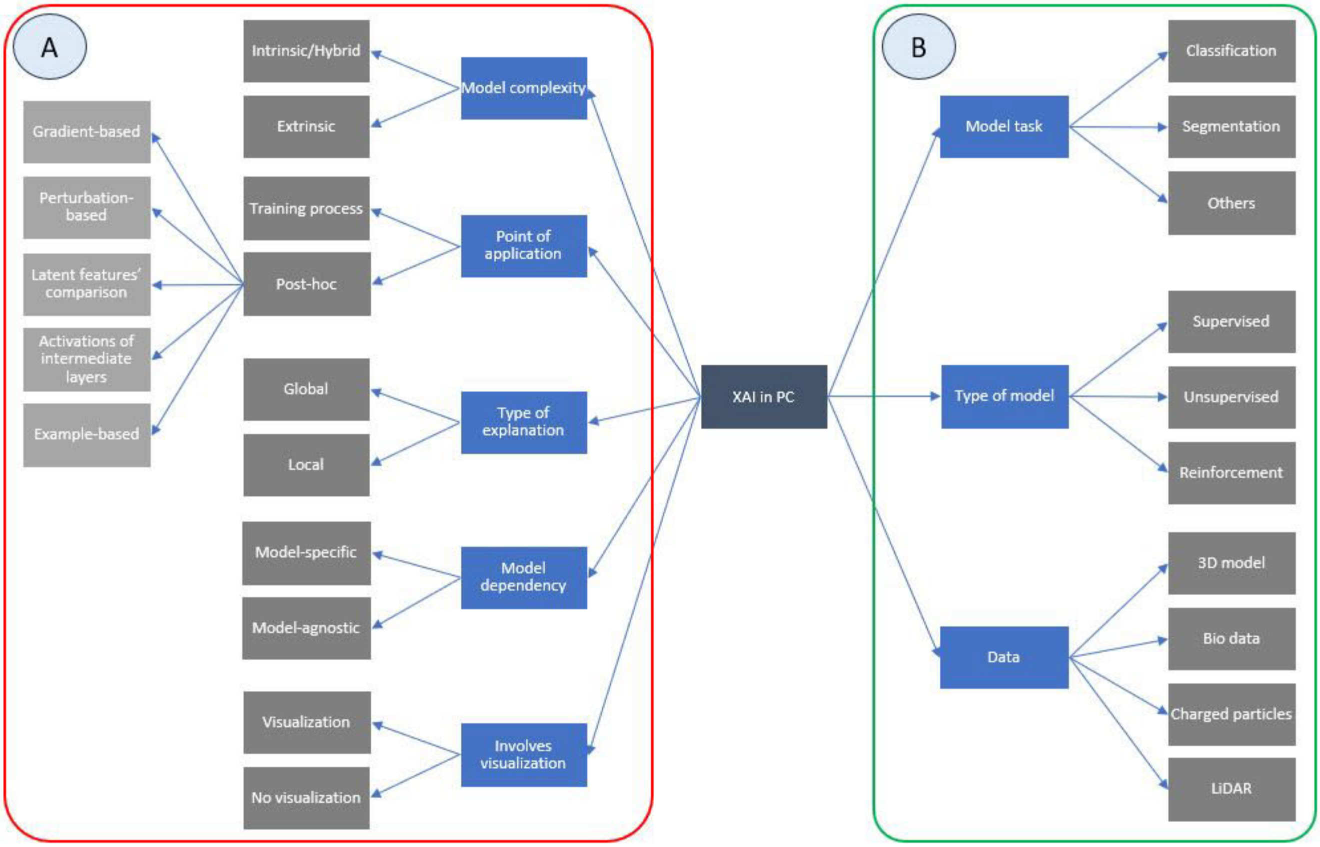


FIGURE 3. Classification of XAI methods in point cloud (PC) data. Ⓐ: Based on the type of explainability Ⓑ: Based on the model and data used.

tries to explain the decision-making process of the model for a particular input instance.

1) Global explanation
2) Local explanation

4) MODEL DEPENDENCY

Here, we take into consideration the dependency of the XAI methods on the models being analyzed. Some of the works attempt to explain particular models working on point cloud data. These are termed as model-specific methods. Methods that do not depend on the architecture of AI models being analyzed and can be easily applied to other models are grouped into model-agnostic class.

1) Model-specific
2) Model-agnostic

5) USE OF VISUALIZATION

Visualization is one of the important aspects of explainability as it assists humans in understanding the model. Therefore, we classify the literature into two groups based on whether the methods involve the use of visualization for explaining the model's decision-making process.

1) Included
2) Not included

The classification of literature based on all the criteria mentioned above in subsection IV-A is shown in Table 1. The affiliation of the papers to specific group is highlighted using a '•' mark. The rows are colour-coded based on the year of publication to help readers identify the initial XAI works in the point cloud-based XAI field and also the latest developments contributing to field. It is also visually helpful for readers in following the classification of individual literature across columns.

B. CLASSIFICATION OF XAI BASED ON MODEL AND DATA

Apart from the classification of XAI methods mentioned above, we also look into the type of point cloud data these methods work on and the type of models they target for explaining (Ⓑ in Figure 3). The *type of point cloud data* class refers to the different things that are represented by point clouds. The *type of models* class takes two criteria into consideration: 1) Type of training and 2) type of task these models perform. Table 2 shows this classification mechanism in the tabular form.

1) TYPE OF POINT CLOUD

The XAI literature can be classified based on the type of point clouds used as input for the AI model. Different types of point clouds encountered in our survey are as follows:

1) 3D models: These are the point clouds representing 3D models/objects such as tables, planes, chairs etc.

TABLE 1. Table containing collected literature and the characteristics of the XAI method used in these literature. The articles are ordered and color-coded based on the year of publication (from 2017 (top) to until May, 2024 (bottom)). "T" indicates that the method was used for explaining training process.

Paper	Model transparency		XAI mechanisms & point of application						Scope of explanation		Model dependency		Visualization	
	Intrinsic models	Hybrid interpretable models	Training process	Post-hoc					Global	Local	Model-specific	Model-agnostic	Yes	No
				Grad-based	Perturbation-based	Compare latent feat.	Activations	Example-based						
[45]							●			●	●		●	
[77]			●				●(T)			●	●		●	
[55]							●			●	●		●	
[82]					●					●		●	●	
[79]							●			●	●		●	
[67]							●			●	●		●	
[18]				●						●		●	●	
[80]		●								●	●		●	
[20]							●			●	●		●	
[81]							●			●	●		●	
[8]			●				●(T)			●	●		●	
[61]					●					●		●	●	
[53]					●					●		●	●	
[40]							●			●	●		●	
[19]								●		●		●	●	
[44]					●					●		●	●	
[59]						●				●	●		●	
[14]				●						●	●		●	
[35]				●		●				●	●		●	
[72]					●					●	●		●	
[21]				●						●	●		●	
[65]					●					●		●	●	
[42]				●						●		●	●	
[28]					●					●	●		●	
[38]	●									●	●			●
[2]		●								●	●		●	
[24]	●									●	●			●
[27]							●			●	●		●	
[34]				●		●				●	●		●	
[50]				●						●		●	●	
[5]						●				●	●		●	
[37]					●					●	●			●
[63]					●				●			●	●	
[62]		●			●				●			●	●	
[66]				●	●					●		●	●	
[41]				●						●		●	●	
[3]					●			●		●		●	●	
[78]							●			●	●		●	
[64]					●				●			●	●	
[54]					●					●		●		●
[15]	●									●	●			●
[22]							●			●	●		●	
[26]					●					●		●	●	
[23]				●						●		●	●	
[49]				●			●			●		●	●	
Total	3	3	2	11	15	4	13	2	3	42	26	19	40	5

TABLE 2. Classification of the literature based on the properties of AI model and data used. The articles are ordered and color-coded based on the year of publication (from 2017 (top) to until May, 2024 (bottom)).

Paper	Type of model			Type of model task								Type of point cloud			
	SL	UL	RL	Classif	Regression	Segm	Obj-det	Reconstr	Registr	Repre. learning	3D models	LiDAR, RGB-D	Particle traj	Med/Bio
[45]	●			●		●					●			
[77]		●						●			●			
[55]	●			●		●					●			
[82]	●			●							●			
[79]	●			●							●			
[67]	●			●		●					●			
[18]	●			●							●			
[80]	●			●							●			
[20]	●			●							●			
[81]	●			●							●			
[8]	●	●		●				●			●			
[61]	●			●							●			
[53]	●			●							●			
[40]	●			●									●	
[19]	●					●						●		
[44]	●			●										●
[59]	●	●								●	●			
[14]	●						●					●		
[35]	●			●							●			
[72]	●					●						●		
[21]	●								●			●		
[65]	●			●							●			
[42]			●					●					●	
[28]	●			●							●			
[38]	●					●						●		
[2]	●			●							●			
[24]		●				●						●		
[27]	●			●							●			
[34]	●					●						●		
[50]	●					●						●		
[5]	●	●		●				●						●
[37]	●			●									●	●
[63]	●			●							●			
[62]	●			●							●			
[66]	●			●							●			
[41]			●					●					●	
[3]	●			●								●		
[78]	●			●							●			
[64]	●			●							●			
[54]	●			●							●			
[15]	●			●		●					●			
[22]	●									●	●			
[26]	●				●							●		
[23]	●					●						●		
[49]	●	●		●				●			●			

2) LiDAR and RGB-D data: This data refers to the point clouds of buildings, train lines or other specific regions in the environment generated using LiDAR and/or RGB-D sensors.

3) Particle trajectories: This data refers to the trajectories of charged particles such as protons in the field of high energy physics (HEP) represented in the form of point clouds.

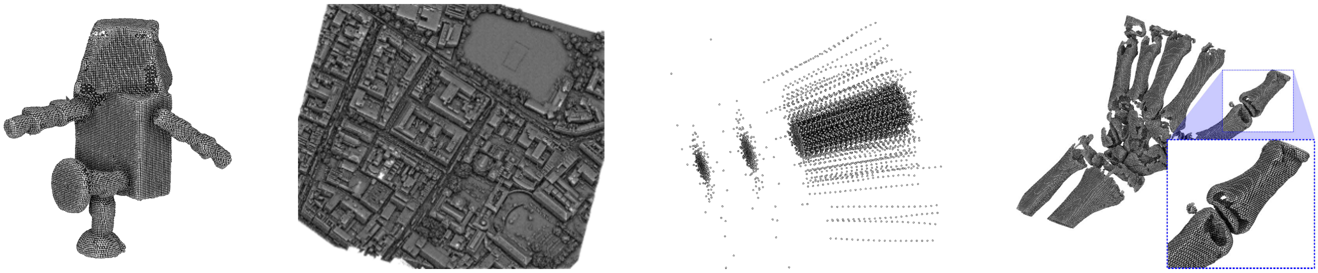


FIGURE 4. Point cloud types (form left to right): 3D model, LiDAR (data courtesy Debra Laefer [69]), particle trajectories (courtesy Bergen pCT Collaboration), medical data (hand CT data courtesy Tiani Medgraph). Visualized using OpenWalnut.

4) Medical/Biological data: These are the point cloud data that represent biological or medical elements such as brain or heart.

An example of the above mentioned point cloud types is shown in Figure 4. We observe in Table 2 that a large portion of the literature caters to the analysis of models working on point clouds representing 3D objects.

2) TYPE OF MODEL

In this group, the XAI methods are classified based on the type of training process used for training a model. We divide the literature into following three classes based on the models considered for analysis in these works.

1) Supervised learning (SL)
2) Unsupervised learning (UL)
3) Reinforcement learning (RL)

3) TYPE OF MODEL TASK

The AI models considered in the collected literature are tailored for specific tasks. Here, we classify the literature based on the tasks listed below.

1) Classification
2) Segmentation
3) Others (eg: Regression, Object detection, Reconstruction, Registration, Representation learning)

Similar to the representation of the literature classification in tabular format in subsection IV-A, we represent the classification of the same set of papers based on the properties of the AI model and the input data in Table 2. We also maintained the same colour-coding and affiliation highlighting methods to help readers identify specific works of interest across both the tables. These tabular representations help readers in identifying patterns and detecting outliers in the surveyed papers and thus, helping them in gaining a better overview of the XAI works in the concerned research field. We also derive interesting insights from these tabular representations which are described later in section VI.

V. SPECIFIC IMPLEMENTATIONS OF XAI MECHANISMS

In this section, ordered by the *type of explainability* (subsection IV-A), we discuss the specific implementations of the explainability mechanisms as introduced by the papers considered for review in this work. This section is the core of the paper because each surveyed paper is discussed in detail here.

A. TRANSPARENCY

Transparency as an explainability mechanism refers to the literature that attempt to make an AI model partially or completely transparent. This allows humans to partially or fully understand the process of detecting features and making a particular decision based on these detected features. In this section, we include models that belong to the *intrinsic models* and *hybrid interpretable models* mentioned in section IV.

Micheletti [38] proposed the idea of using group equivariant non-expansive operators (GENEOs) to develop transparent ML models dedicated for various tasks. A GENEO is a functional operator that performs data transformation. The author proposed using SCENE-net [25] (see Figure 5) for point cloud segmentation task. Here, the GENEOs that are in the form of convolutional operators detect some of the important geometrical features in the input data. The input data is a LiDAR point cloud data that captures energy transmission system with surrounding vegetation containing bushes and trees. The geometrical features are captured using three types of kernels in the GENEOs: 1) Cylindrical kernel to detect vertical structures, 2) Cone-cylinder kernel (cylinder with a cone on the top) to distinguish towers from trees, and 3) Spherical kernel to detect bushes and tree crowns. These features are used by the network to segment the point cloud data. Each GENEO unit $G_i^{\nu_i}$ in the first layer is associated with a parameter λ_i as seen in Figure 5 that is learned during the training and it indicates the importance of corresponding GENEO unit. The output of these GENEOs in the first layer and their corresponding parameters, λ_i, are used by another GENEO unit H (referred to as the "observer" in Figure 5) in the following layer that performs a convex combination and the output is transformed (by M) into the probability of whether a point belongs to a tower or not. At the end, a threshold operation is used to classify the data. Through the use of the interpretable kernel types, this pipeline makes the network simplified and therefore, more interpretable compared to the regular neural networks.

Kyuroson et al. [24] proposed an unsupervised learning network to perform point cloud segmentation for power lines inspection in a smart grid. The point cloud is captured

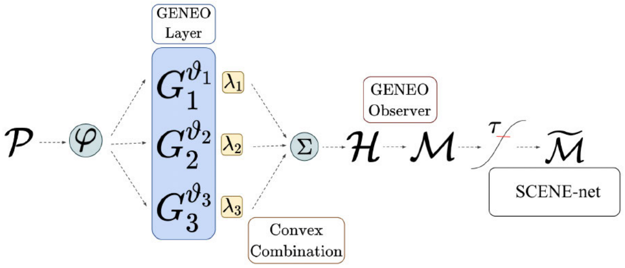


FIGURE 5. SCENE-net architecture. Image taken from [38].

by an unmanned aerial vehicle equipped with a LiDAR sensor. The authors make use of clustering algorithms to perform segmentation. In particular, the proposed method is a two-stage unsupervised hierarchical clustering method that is based on DBSCAN and Kd-tree to extract power lines from the LiDAR point cloud data with spatial coordinates. The initial stage involves extracting high-elevation points from the point cloud data using statistical analysis applying density criteria and histogram thresholding. The Kd-tree is used to structure and spatially organize the data to remove points from the set of high-elevation points that do not represent power lines. Using PCA [43], power lines are extracted from these high-elevation points. The following stage involves the segmentation of these extracted power lines using two-layered DBSCAN clustering to analyze the both the directions (orthogonal and parallel planes) of the power line span. This use of methods such as DBSCAN and Kd-tree whose working principles are transparent makes the network transparent compared to deep neural networks.

Zhang et al. [80] proposed PointHop, an explainable ML method for point cloud classification. The method consists of two stages. The first one corresponds to building attributes in a local-to-global fashion using iterative one-hop information gathering. The second stage corresponds to classifying the point cloud data based on the attributes generated. Figure 6 shows an overview of this method. Each point in the input point cloud data is represented by its spatial coordinates. The authors use these attributes corresponding to a point and its neighbors within one-hop distance to compute new attributes. These points in the neighborhood are determined using K nearest neighborhood method taking the distance measured by the Euclidean norm into consideration. As the number of hops becomes larger (leading to more points being considered as neighbors), the number of attributes increase as well covering a larger receptive field. To address the issue of large number of attributes, Saab transform is used in each PointHop unit as a dimensionality reduction technique. Subsequently, these attributes are aggregated using multiple aggregation schemes (M schemes in Figure 6) and used by a classifier model such as Random Forest (RF) to classify the data. This use of point-hop mechanism (top row in Figure 6) for building attributes adds transparency to the PointHop method. However, the entire method is not completely transparent due to the use of classifiers such as RF which do not fall into the transparent models group.

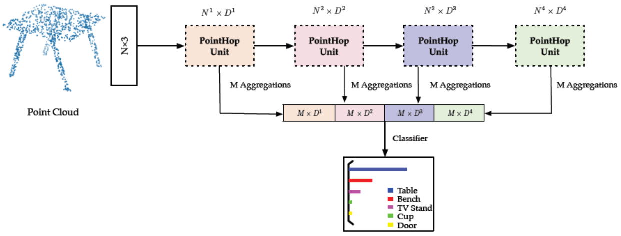


FIGURE 6. An overview of the PointHop method for point cloud classification. N^i and D^i refer to the number of points and number of attributes respectively. Image taken from [80].

Arnold et al. [2] proposed eXplainable Point Cloud Classifier (XPCC) method which is a prototype-based classifier. The network (see Figure 7) consists of a pre-trained kernel point convolutional neural network (KP-CNN) that is used to extract features from the input point cloud data that are then compared with the features of prototypes representing each class to determine which class the input point cloud data belongs to. In the local similarity layer, the extracted features of the input data are compared with the corresponding features of the prototypes of each class (each class has multiple prototypes). In the global similarity layer, the similarity score for each class is extracted from the most similar prototype in each class. In the following layer, the classes' similarity score is weighted by the input data's similarity to the respective classes' compound prototype. The compound prototype corresponding to a class consists of all the prototypes belonging to that class superimposed. In the last layer, a softmax function is applied and hard classification is performed to determine the predicted class. This use of prototypes and similarity criteria makes the model more transparent but the use of KP-CNN for feature extraction excludes it from being considered as a completely transparent model.

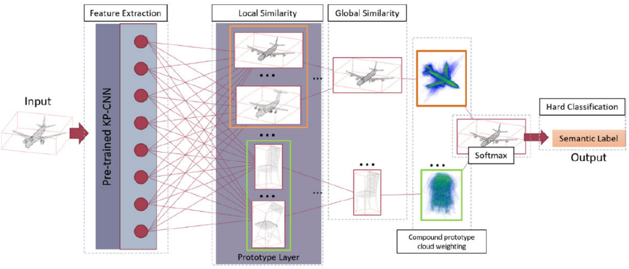


FIGURE 7. XPCC architecture for point cloud classification. Image taken from [2].

Tan [62] proposed Fractual Projection Forest, a pipeline that utilizes fractal features to help ML models perform point cloud classification tasks. The concept is similar to the PointHop method [80] where features are generated from the input data in an interpretable way and a classifier such as a RF model is used to classify the point clouds based on these generated features. In this work, the features are generated from the projections (1D and 2D projections) of the input point cloud data. The point cloud is projected on each of the x, y, z axes and xy, yz and xz planes. Relevant features are generated from these projections using multiple windows of varying size as shown in Figure 8. A Gaussian distribution is fit onto each of these windows to obtain corresponding Gaussian parameters. All these generated features are concatenated and used as input for the classifier model. We added this model to the *hybrid interpretable model* group due to the use of the explainable feature generation process and the use of RF classifier in that is opaque.

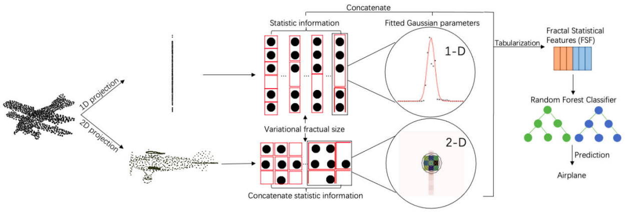


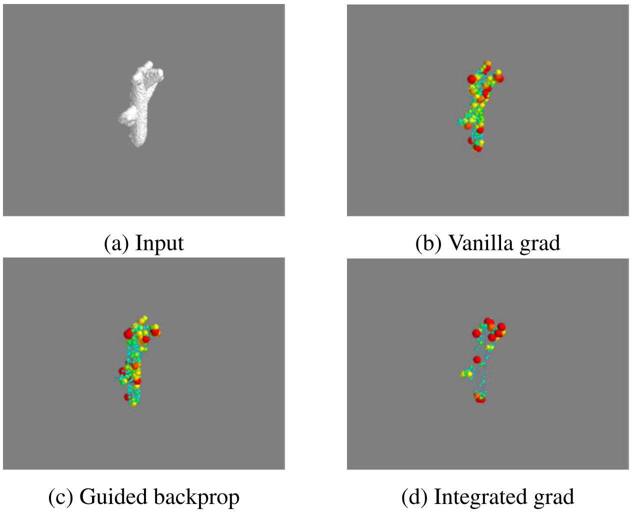
FIGURE 8. **An overview of the Fractual Projection Forest architecture for point cloud classification. Image taken from [62].**

Feng et al. [15] proposed an interpretable classifier called Interpretable3D for point clouds classification. It is based on the prototype comparison mechanism where the label (or class) of the input sample is determined by assigning the label corresponding to the most similar prototype. The similarity is computed using cosine similarity function. To represent each class effectively, certain number of prototypes are selected instead of using one prototype for each class. The classifier consists of two parts where one part learns the underlying representations of the samples and the other part predicts the labels.

Transparency is the most preferred type of explainability. However, it becomes more difficult to implement as the model tasks get more complex and thus, leading to complex model architecture. In our survey, only six surveyed papers attempted to provide explainability through a transparency mechanism. Three of these papers proposed hybrid interpretable models that contain only certain layers in the architecture that are transparent. The remaining three papers proposed intrinsic models made up of transparent layers.

B. GRADIENT-BASED METHODS

Gradient-based explainability methods rely on the computation of gradients through backpropagation to generate explanations for neural networks. Backpropagation is a method to estimate gradients by propagating errors backward in neural networks. These gradients are used to update network parameters (weights) during the training process. Some of the major works in this area are SmoothGrad [56], Integrated gradients [60], and Grad-CAM [51] which were mainly developed for ML models working on image data. However, many researchers have attempted to adapt these methods to explain ML models working on point cloud data. One of the earliest works in this direction was by Gupta et al. [18]. They adapted two of the most widely used gradient-based XAI methods (Guided backpropagation [58] and integrated gradients [60]) to point cloud data and visualized the results along with vanilla gradients as shown in Figure 9. The points coloured red indicate their high influence on the output value.



(a) Input

(b) Vanilla grad

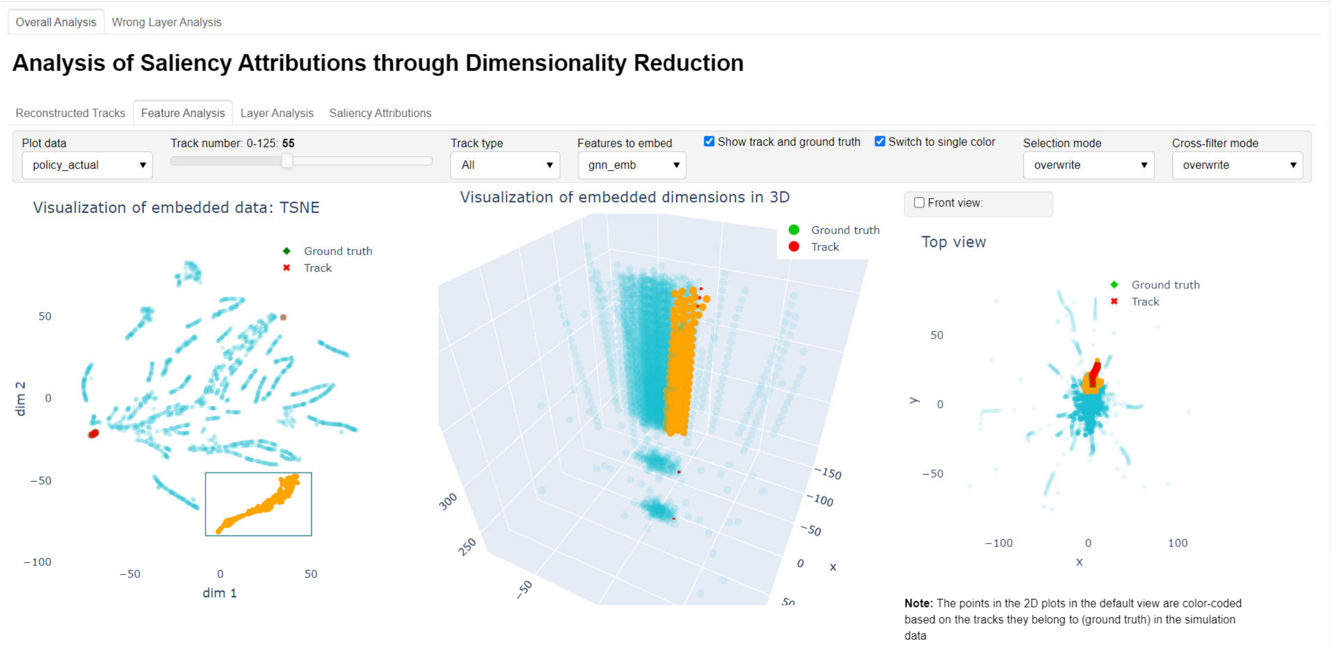
(c) Guided backprop

(d) Integrated grad

FIGURE 9. **Saliency maps for Pointnet++ model working on point clouds representing an airplane. The saliency attributions are represented by a red (large) to blue (small) heatmap. Image taken from [18].**

Mulawade et al. [42] adapted two gradient-based methods to point cloud data representing particle trajectories. The authors adapt SmoothGrad [56] and integrated gradients to a deep RL model tracking charged particles using point cloud data as input. The model is a multi-input multi-output model and the initial visual analysis of the model's working included the adaptation of the above-mentioned gradient-based methods to a subset of the input data. They extend this work to include all the input features for analyzing the deepRL model in [41]. Here, they analyzed both the adaptations to determine the method that provides better insights, extended its adaptation to the remaining features in the input data and designed a visual analytics (VA) system for the thorough analysis of the model's decision making process. To address the issue of visualization of high-dimensional saliency attributions, they made use of dimensionality reduction technique t-SNE [71]. The visual analytics system consists of multiple parts (or tabs) containing interactive tools that address specific requirements. Figure 10 shows one of these tabs from the VA system consisting of visualization of embedded saliency attributions (left), 3D visualization of the corresponding point cloud data (center) and its 2D projection (right) with all the three plots linked to each other for better exploration of the data.

In addition to the above mentioned papers, Schwegler et al. [50] also extended the integrated gradients method to the ML models working on semantic segmentation of point


Overall Analysis | Wrong Layer Analysis

Analysis of Saliency Attributions through Dimensionality Reduction

Reconstructed Tracks | Feature Analysis | Layer Analysis | Saliency Attributions

Plot data: policy_actual ▾
Track number: 0-125: 55
Track type: All ▾
Features to embed: gnn_emb ▾
☑ Show track and ground truth
☑ Switch to single color
Selection mode: overwrite ▾
Cross-filter mode: overwrite ▾
□ Front view:

Note: The points in the 2D plots in the default view are color-coded based on the tracks they belong to (ground truth) in the simulation data

FIGURE 10. Visual analytics system for understanding a deepRL-based particle tracking network. Image taken from [41].

cloud data. They generated a baseline with high entropy and interpolated it with the input point cloud data that consists of coordinates and RGB values of each point. These interpolated point cloud data instances are used as input and their corresponding gradients are computed for all the classes in the segmentation to generate saliency attributions. These attributions are then projected back onto the actual input point cloud data to indicate the importance of each input feature on the output.

Dworak et al. [14] adapted the Grad-CAM method to object detection architectures working on LiDAR point cloud data in automotive perception systems. They proposed an object detection architecture for point cloud data and adapted the Grad-CAM method for the same. The point cloud data is voxelized using a voxelization method and features are extracted using a Voxel Feature Extractor (VFE) network. These features are processed by a series of convolutional layers producing a multi-dimensional output tensor, with the first two dimensions corresponding to the 2D grid of cells that the whole region of interest is divided into. The third dimension corresponds to the anchor boxes of different sizes and the last dimension is a per-cell (or anchor) vector of predicted feature values.

Matrone et al. [35] proposed BubblEX adapting Grad-CAM for point cloud classification networks. Figure 11 illustrates this adaptation where the activations corresponding to the last convolutional layer *conv*5 and the gradients computed by backpropagation of the target class to the *conv*5 layer are utilized. The product of these activations and gradients is used as the saliency map to indicate the importance of each point in the input point cloud data. The authors extended this adaptation to provide explanation for networks designed for point cloud semantic segmentation [34]. In this adaptation, the authors utilize the penultimate convolutional layer for generating gradients and activations for implementing Grad-CAM as the output layer is also a convolutional layer. The gradients are computed for individual class in the segmentation and a saliency map is generated to highlight the importance of input features for the corresponding class.

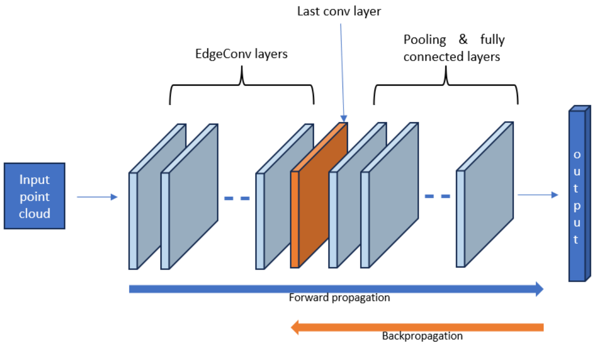


FIGURE 11. Adaptation of Grad-CAM for point cloud classification network in [35].

Huang et al. [21] proposed an interpretable module called descriptor activation mapping (DAM) that is inspired by Grad-CAM but includes modifications to suit the point cloud registration model. The registration process aims to estimate the transformation matrix between two point clouds using the corresponding descriptors generated by the registration model. Here, the individual values in the output descriptor vector (or channel) are used as the loss for backpropagating to the target layer to compute gradients, unlike the target class value in the original implementation. The descriptor activation map is computed for each descriptor element that describes the contribution of each data point in the input for the same. The final descriptor map is generated by adding the descriptor activation maps corresponding to all the descriptor elements in the output.

Kuriyal et al. [23] proposed pGS-CAM (point Grad-Seg Class Activation Mapping) that is inspired by the Grad-CAM method for generating saliency maps (or explanations) for neural networks performing semantic segmentation of point cloud data. The authors use an aggregation operation to capture the collective gradient impact across all logits corresponding to a class at an intermediate activation layer. These gradients are aggregated for each filter in the intermediate layer and multiplied with the corresponding activations of the filters. The proposed method also provides the possibility to focus on a subset of points in the input data for which the explanations can be computed by aggregating the gradients of the logits corresponding to this subset. This helps in exploring the explanations for specific segments in the output data.

Romaneli et al. [49] used gradients to analyze a point-cloud-based transformer model intended for tasks such as classification and reconstruction of point clouds representing 3D models. The input data is divided into multiple patches and these patches are encoded (or embedded) using a neural network before using them as input for the transformer model that produces a feature vector for each of these patches. To understand the importance of input data in generating the feature vector for a particular patch, gradients are computed by backpropagation from its feature vector to the embeddings highlighting the importance of patches on the feature vector.

Gradient-based methods have gained popularity in the last decade. This is mainly due to their attempt to provide explanations for complex neural networks that are extremely difficult to understand. Almost a quarter of the surveyed works (11 out of 45) proposed gradients-based methods reflecting the aforementioned trend in point-cloud-based XAI.

C. PERTURBATION-BASED EXPLANATION

Perturbation-based explanations involve modifying the input data to observe changes in the output value taken into consideration. The amount of change in the output value is interpreted as the extent of influence the perturbed input features have on the output value. Some of the significant XAI works in perturbation-based methodologies are SHapley Additive exPlanations (SHAP) [32] and Local Interpretable Model-agnostic Explanations (LIME) [47]. Some of the surveyed point-cloud-based XAI literature proposed explainability methods that use this mechanism to analyze ML model behaviour.

Shen et al. [53] proposed using Shapley values [52] to evaluate the representation quality in a deep neural network (DNN) performing point cloud classification. The point cloud data is divided into multiple regions and sensitivity metrics such as the regional rotation sensitivity, the regional translation sensitivity, the regional scale sensitivity, three types of regional structure sensitivity (sensitivity to edges, surfaces, and masses) are used to evaluate the DNN model taken into consideration. The perturbation is introduced by resetting the coordinates of the subset of points to the center of the point cloud and recording the corresponding change in the output class value. This generates a sensitivity map indicating how certain regions in the input point cloud data influence the output value.

Zheng et al. [82] used a similar perturbation method to analyze ML models working on point cloud classification. They consider the spherical coordinate system and perturbation is induced by moving the point towards the center of the system and observing the change in the corresponding output class value. This point shifting mechanism is assumed to have a similar effect to the point dropping mechanism where a particular point is dropped from the input point cloud and the change in output class is used as the saliency attribution for the dropped point. Figure 12 shows an example of the point cloud saliency map generated in this work.

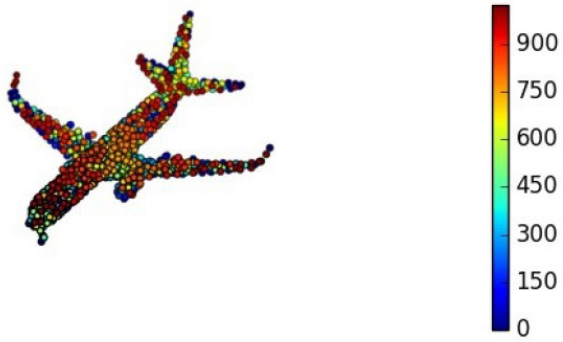


FIGURE 12. Saliency map for input point cloud classification network. The color-coding is based on their score-rankings (higher value indicates higher importance). Image taken from [82].

Verbung [72] proposed a perturbation method to analyze a ML model working on point cloud semantic segmentation. They focus on explaining the PointNet++ model that is used for segmenting point cloud data representing catenary arches. The perturbations are introduced by modifying certain objects in the input point cloud data and observe their effect on the segmentation result. They change insulator shape, insulator location and shape of the pole (introducing different amounts of holes in the pole) in the input data and study their effect on the output.

Tan and Kotthaus [65] adapt LIME to explain DNNs working on point cloud classification. LIME is a local surrogate model-based explainability method where multiple perturbed input instances are created from the given input data and a surrogate linear model is trained on these perturbed instances to learn the decision boundary. Since a linear model is used, it is easily explainable due to its transparency.

In the adaptation, the authors divide the input point cloud data into multiple regions using a clustering algorithm and introduce perturbations based on these regions. These perturbed instances are fed to the classification model to record their corresponding output class scores. Then, they trained a linear regressor that approximated the output class score of the classification model taken into consideration.

Another interesting work by Tan addressing the explainability issue using the perturbation methodology is presented in [63]. Here, the author proposed a generative model-based activation maximization (AM) method to explain a point cloud classification model. We included this work to the perturbation-based explanation category as it involves modifying input data to maximize a particular class score in the output vector. Initially, an autoencoder is trained to reconstruct point cloud instances in the real data and thus learning their distribution. Later, a latent vector (size $(k * 1)$) (as shown in Figure 13) is initialized and the decoder part (generator) is utilized to generate point cloud data from it. This generated point cloud data is then fed to the autoencoder model to produce a point cloud data that is close to the real ones. This data is used as the input for the ML model. The target class value in the output is optimized via backpropagation by modifying the latent vector and generating corresponding input instances as shown in Figure 13. The optimization leads to generation of a point cloud data that maximizes the corresponding target class value and thus, providing the information regarding the geometrical structure that produces maximum value for the target class considered. Comparing other input instances with this point cloud provides an explanation regarding their corresponding output values.

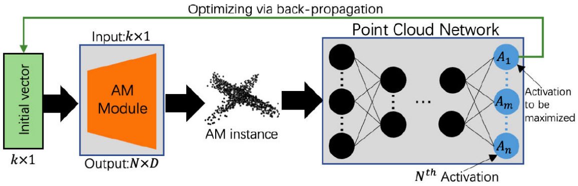


FIGURE 13. General overview of the architecture for point cloud activation maximization (AM). Image taken from [63].

In addition to the activation maximization-based XAI method mentioned above, the author also proposed activation-flow based AM method called Flow AM [64] that generates global explanations for the classification network taking the activations of the intermediate layers into account instead of using a generative model. The neurons in these intermediate layers are forced (through the regularization process) to align their activation values to the ones corresponding to the real objects during the activation maximization process while maximizing the target class value in the output. The initial input for the AM process is defined by computing the average of the points in the test dataset for each class. The positions of the points are regularized by limiting the expansion of outlying critical points and minimizing the distance between the neighboring points.

Tan also proposed a grouped feature ablation method in [62] to understand the decision-making process of a classification network. Here, a set of features are removed from the whole dataset, the network is retrained and the testset accuracy is recorded. This accuracy is compared to the accuracy value obtained from the unablated testset. The change in accuracy is used as the attribution for the features removed.

Pölsterl et al. [44] proposed Shapley Value Explanation of Heterogeneous Neural Networks (SVEHNN) that attempts to explain Alzheimer's diagnosis made by a DNN using multi-input data consisting of the 3D point cloud of the neuroanatomy and tabular biomarkers. They created baselines for both the inputs and used these values to replace corresponding features and compute the change in output value. Zero is used as the baseline for tabular biomarkers and for the point cloud data, they created a hull containing all point clouds in the dataset from which the matching point is selected as a replacement in the input instance.

Tan and Kotthaus [66] proposed two adversarial attack-based explainability methods, One Point Attack (OPA) and Critical Traversal Attack (CTA) for ML networks working on the classification of point clouds representing 3D objects. The integrated gradients [60] method is used to determine the critical points that are used to induce perturbation into the input instance. Since the adaptation of integrated gradients is not explained in the paper, we do not explain it in subsection V-B. However, the adaptation of gradient-based method is indicated in Table 1. In the OPA method, the point with highest attribution (generated by integrated gradients method) is selected and shifted with an iterative optimization process until the output prediction changes. In the CTA method, the process starts with one critical point and later, other points are added to the critical points set based on their integrated gradients attributions. This addition of points to the critical set is carried out if the perturbation of the current set does not alter the prediction class.

Taghanaki et al. [61] proposed PointMask which is a model-agnostic method used for attribution in point cloud models. PointMask learns to mask out input points that have negligible contribution to the model's output. This is implemented by introducing a differentiable layer before the encoder part of the classification network that maximizes the mutual information between the masked points and the class labels. The masking is defined by a threshold that is applied on the attribution values of the point cloud. Figure 14 shows the use of thresholding to identify points that have high influence on the output value. The masking mechanism learns to mask points during the training process of the classification model. This is achieved by introducing a regularization term in the loss function consisting of classification loss. Since this method of masking input data is a type of perturbation-based mechanism, we have included this work in the *perturbation-based explanation* subsection.

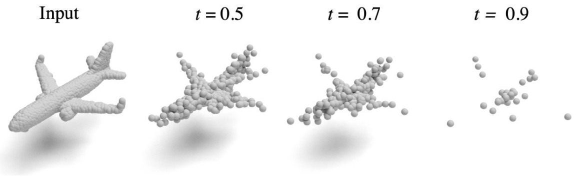


FIGURE 14. Thresholding in the PointMask method. Increasing the threshold value (*t*) leads to the detection of points in the input data that are more influential on the classification output. Image taken from [61].

Miao et al. [37] proposed a method called Learnable Randomness Injection (LRI) that detects points in the input point cloud data that influence the output class label. The method is proposed for geometric deep learning (GDL) models working on the classification of point clouds belonging to the high-energy physics and biochemistry field. The idea of LRI method is to learn injecting randomness to the input data along the training process. The architecture of the network consists of an interpreter g and a classifier f as shown in Figure 15. The interpreter learns to encode the input data and generate randomness to induce perturbation to the data. The classifier learns to classify the perturbed data using the output of the interpreter. The authors use two types of randomness. The Bernoulli randomness is added to measure the *existence importance* of points (top row in Figure 15) and the Gaussian randomness is used to analyze the *location importance* of points in the input data (bottom row in Figure 15).

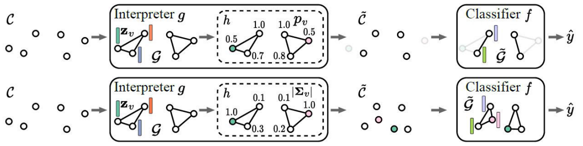


FIGURE 15. The architectures of the LRI method using Bernoulli and Gaussian randomness. Image taken from [37].

Atik et al. [3] adapted SHAP for a point-cloud-based classification network and also proposed the use of filter-based feature selection algorithms to determine the importance of input features. They mainly focused on the explanation of ensemble ML models performing the point cloud classification task. The filter-based feature selection algorithms use two types of methods to generate importance for input features. The first method computes feature's importance by utilizing the network's prediction on the input data containing all the features and the perturbed input data that does not contain the feature taken into consideration. The second method is described in subsection V-D as it uses example-based explanation mechanism.

Shen et al. [54] proposed a set of generic rules for modifying a point-cloud-based neural network to a rotation-equivariant quaternion neural network (REQNN). To evaluate the performance of REQNNs with respect to different rotation angles of the input point cloud data, the authors compute Shapley values for comparison. Here, the input point cloud data is uniformly divided into n regions and the Shapley values for each region are computed. They perturb the input data by moving the points belonging to the regions to be excluded to the center of the input point cloud data instead of removing them from the input point cloud data. The stability of these regional attributions is calculated by computing the cosine similarities between input point clouds with different rotation angles.

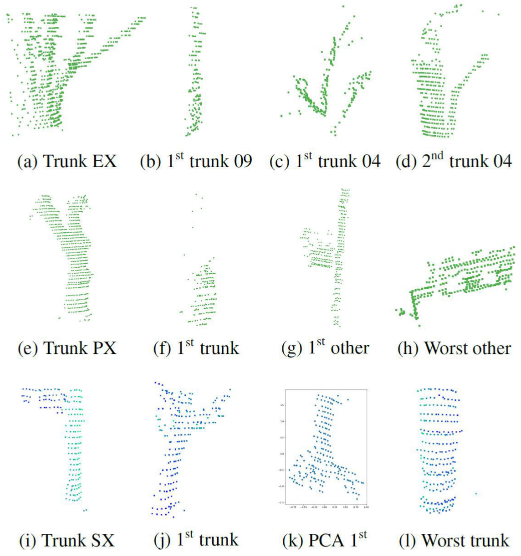
Lavasa et al. [26] proposed an AI-based method to predict the point-wise accuracy of laser scanners across the surface of the object represented by point cloud data. In other words, the method outputs one value for each point in the input point cloud data to indicate how accurately a certain laser scanner has captured it. This process is performed for obtaining accuracy values for points along each of the three axes (x, y, z). They developed a model that predicts the accuracy of laser scanning devices and also informs the users about the features that influence these predictions. To provide the explanation, the authors adapted the SHAP method (based on the cooperative game theory) to the input point cloud data and computed Shapley values that highlight the importance of individual points on the prediction.

The large number of papers falling into this category (15 out of 45) could result from the fact that the basic mechanism of perturbation-based methods does not rely on the network architecture. This makes it easier to adapt methods devised for other data types to models dealing with point clouds.

D. EXAMPLE-BASED EXPLANATION

As explained in subsection IV-A, example-based explanations utilize the concept of input instances' comparison to provide an explanation for the decision-making process of the model. Heide et al. [19] utilized this mechanism to generate explanations for the semantic segmentation of point cloud data and proposed an approach named X^3Seg. The method provides explanations by selecting the most similar and most dissimilar point sets with respect to the input point cloud being examined. It consists of three methods: Encompassing (EX), Selective (SX), and Predictive X^3Seg (PX). The encompassing method selects examples from the whole training dataset whereas the selective method selects from a smaller subset that represents the whole dataset. An example of the explanation is shown in Figure 16.

Atik et al. [3] proposed two filter-based feature selection algorithms to generate explanations for ensemble ML networks working on point cloud classification. One of the methods involves the perturbation mechanism and is explained in subsection V-C. The other method determines the importance of features by comparing the samples that are similar to the input instance and another set of randomly selected samples from the training set. The importance of a feature is reduced if it has different values (in the input instances) in the similar input samples selected. In the


(a) Trunk EX (b) 1st trunk 09 (c) 1st trunk 04 (d) 2nd trunk 04

(e) Trunk PX (f) 1st trunk (g) 1st other (h) Worst other

(i) Trunk SX (j) 1st trunk (k) PCA 1st (l) Worst trunk

FIGURE 16. Example-based explanations with encompassing (a-d, EX), predictive (e-h, PX), and selective X^3Seg (i-l, SX) (S indicates the similarity score): (a), (e), (i) coherent 3D point sets for explanation; (b), (c), (d), (f), (j) best same-class prototypes from prototype database ($S_b = 0.113, S_c = 0.178, S_d = 0.192, S_f = 0.016$); (g) best different-class prototypes ($S_g = 0.014$, pole); (h), (l) worst-matching prototypes (criticism; $S_h = 4.154$, building). Image taken from [19].

dissimilar samples, if the feature has different values, then its importance value is increased.

Example-based explanation is a unique type of explanation as it uses the concept of similarity between input data instances to provide explanations. In this survey, only two papers used this mechanism to provide explanations for the point-cloud-based AI models. One of the papers ([19]) relies on human interpretation to determine important features in the input instances that are similar to the input instance under consideration as it does not produce saliency maps from the comparison. However, the other paper ([3]) computes saliency attributions for input features based on the comparison of the input instance with similar and dissimilar data instances selected for explanation.

E. ACTIVATIONS OF INTERMEDIATE LAYERS

The use of activations of intermediate layers has been one of the most common methodologies of analyzing and understanding the working of AI models. Layer activations provide humans with the information pertaining to feature detection in corresponding layers. The explanation of the AI model using the layer activations relies on the analysis of these activation values. Some of the literature selected in this survey use this mechanism to provide insights into the working of ML models.

Zhang et al. [80] visualize the activations of the first layer (see Figure 6 for the architecture) to indicate the features learned by the corresponding layer. The authors show that the first layer learns to detect common patterns such as corners in the input point cloud data.

Qi et al. [45] used the activations of an intermediate layer to indicate critical points in the input point cloud data that influence the decision-making process of the ML model. The authors make use of the max pooling operation on the preceding layer in the network to determine these critical points (see Figure 17). These are then identified in the input data to highlight their importance. Figure 18 shows some examples of the critical points identified in four input data instances.

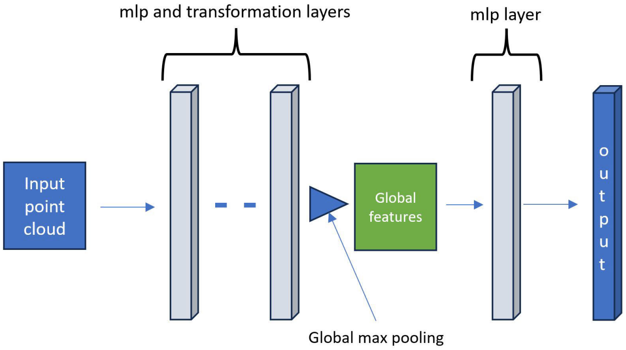


FIGURE 17. An overview of the PointNet architecture for point cloud classification proposed in [45].

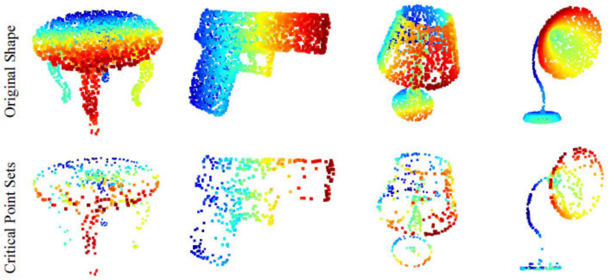


FIGURE 18. Critical points in the input data that influence the output. The color-coding is based on the depth information. Image taken from [45].

Levi et al. [27] extended this concept of critical points [45] by introducing discrete and continuous measures to rank the points in the point cloud based on their importance. The ranking is performed on the activations of the intermediate layer that is subjected to max pooling operation in [45]. The discrete measure assigns ranking based on how many features corresponding to each point (in the output of the intermediate layer) contribute to the global feature vector when subjected to the max pooling operation. The smooth measure takes all the activation values (per point features) corresponding to each point into consideration and ranks the points based on the aggregate of these per point features obtained from the intermediate layer that precedes the pooling layer.

Huang et al. [20] proposed the class attentive interpretable mapping (CLAIM) approach to understand the decision-making process of the PointNet network proposed in [45]. They replace the global max pooling layer (shown in Figure 17) with global average pooling layer in the network.

Then, they use the input and the ouput of this pooling layer along with the network weights associated with the following layer and the output layer to generate saliency maps for the input point cloud data. Such saliency maps can be generated for individual classes in the output vector to study the influence of each feature in the input data on the output classes.

Zhao et al. [81] proposed an interpretation network that is based on 3D point cloud deep neural networks to get better insights into the 3D convolutional operations. Figure 19 shows an overview of the interpretation network with a dynamic graph CNN (DG-CNN) [73] as the basic line network. The network's input is a point cloud of 3D objects containing n points with their x-, y-, and z-coordinates. The input data goes through multiple DG-CNN layers and multi-layer perceptron (MLP) layer before it splits into two classification networks as shown in Figure 19. The *internal consistency network* is used to obtain the feature map of each filter in the convolutional layers that are the intermediate layers. The authors determine fully activated and partially activated filters by adding up all the features produced by each filter and exploring the mean and standard deviation of these features. The *external consistency network* is used to generate attributions (class activation maps) of each point in the input to the classification output. These activation maps are generated from the $n \times C$ output of the intermediate layer.

Yang et al. [77] proposed a folding-based autoencoder called FoldingNet that reconstructs point clouds representing 3D objects. Here, the folding refers to a decoding operation proposed by the authors. They visualize the outputs of intermediate layers to understand how the network learns during the training process as shown in Figure 20.

The folding operation is implemented by introducing an m-by-2 matrix containing m (the number of points in output point cloud) grid points to the latent space features produced by the encoder. The latent space features are replicated m times and the $m \times 2$ matrix is concatenated with these features. The output of this concatenation operation is processed by a three layer perceptron (first folding operation) producing $m \times 3$ output. The replicated features are concatenated to this output and a folding operation is applied on it again producing $m \times 3$ matrix. The outputs of these folding operations are used to understand how the network learns to capture the structural information in the input point cloud data over the training process.

Cao et al. [8] adapted this folding mechanism to provide explanations for the working of deep learning networks working on point cloud classification. They adapt it by introducing the decoder network to the classification. The decoder takes the latent feature vector computed by one of the intermediate layers (max pooling layer) as input and reconstructs the input point cloud using folding operations. Visualizing the outputs of these folding layers (as shown in Figure 20) provides insights (as described in the previous paragraph) into the learning process of the network during the training process. The authors also extended the 3D class activation mapping mechanism proposed by Huang et al. [20] for understanding PointNet [45] network (the CLAIM approach) to understand the decision-making processes of other types of point cloud classification networks.

Thomas et al. [67] presented Kernel Point Convolution (KPConv) that performs convolution operations directly on the point cloud without any intermediate representations. To understand the classification network built using KPConv layers, the authors visualized activations of the intermediate KPConv layers by projecting them on the input point cloud data to highlight the features detected by these layers when classifying 3D objects represented by point cloud. This visualization demonstrated that the layers in the initial part of the network detect low-level features such as corners or vertical and horizontal planes and in the latter layers, the network detects complex features such as cones and stairs.

Shen et al. [55] proposed the use of a kernel correlation layer and a graph-based pooling layer to capture local geometric structures with a clear geometric interpretation. The kernel here refers to a set of learnable points whose positions are modified through backward propagation as the network learns to detect specific features in the input point cloud data. The authors use the visualization of the features captured by these kernels to provide an insight into what the network has learnt during the training process.

Zhang et al. [79] proposed methodologies to understand the working of a PointNet network used for classification purpose. They proposed visualizing point functions and generating class-attentive response maps to understand the network's decision-making process. The point functions correspond to the global features produced in the PointNet network shown in Figure 17. Thus, visualizing the point functions helps us understand what each global feature represents in the input data. To get additional insights into how the input features influence each class, they modified the PointNet architecture to generate class-attentive global features and class-attentive response maps. An overview of this architecture can be seen in Figure 21. The modification is performed by removing the max pooling layer in PointNet and adding an MLP layer to reduce the dimension ((to obtain class attentive features) of the per-point features to the number of classes, C, and perform global average pooling operation to generate an output vector.

Mokhtar et al. [40] adapted the layerwise-relevance propagation (LRP) [4] method for machine-learned partical flow (MLPF) network that works on charged particle track reconstruction. The input data consists of a set of detector (that detects charged particles passing through) elements such as the energy of the particles and the azimuthal angle and the ML model predicts the corresponding set of particle flow candidates. The input data is converted into a graph data before feeding it to the MLPF network which is a graph neural network (GNN). Thus, the aggregation of messages from neighboring nodes in the graphs is also taken into account by the authors when applying LRP. They distribute relevance scores per node using the weights and layer activations in

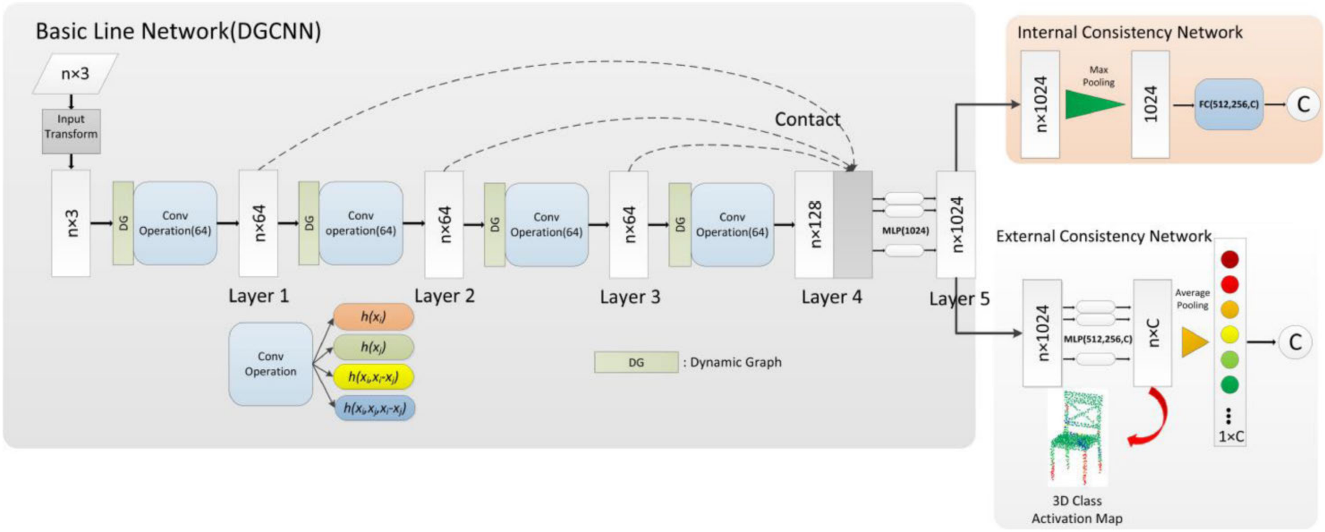


FIGURE 19. An overview of the interpretation network. Image taken from [81].

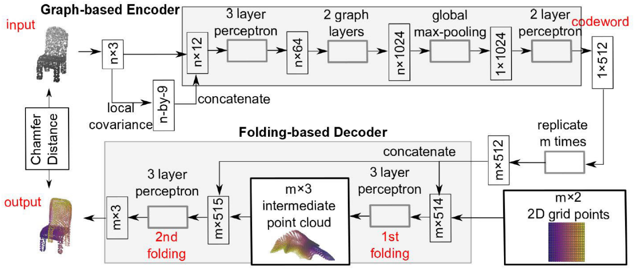


FIGURE 20. FoldingNet architecture. The color gradient is used to illustrate the correspondence between the 2D grid and the reconstructed point clouds after folding operations. Image taken from [77].

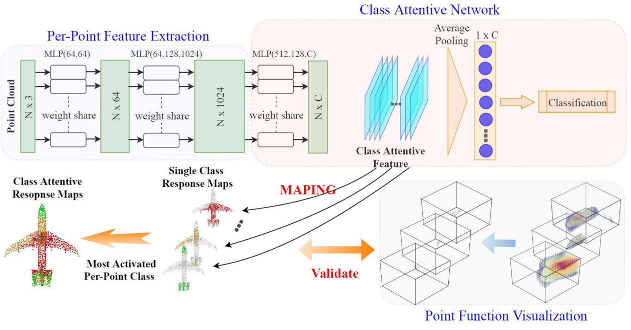


FIGURE 21. An overview of the network architecture and the explainability mechanisms proposed by Zhang et al. in [79]. Image taken from the same literature.

the network, and thus indicating the node's contribution to the relevance of the node in the following layer. This method utilizes the activations and weights associated with the intermediate layers to generate saliency attributions for the input data.

Liang et al. [28] proposed a method to compute saliency maps for point-cloud-based classification networks. The method intends to find several non-contribution factors in the input space. Moving any point in the point cloud to these non-contributing factors' positions leads to change in the output value similar to the point dropping method where the influence of the point is nullified by dropping it from the input data. The authors perturb the input by releasing several free factors in the target space and compare the pooled features (features obtained after using a pooling layer in the network) and determine the regions that do not contribute to the output value.

Levi et al. [78] proposed a method called Feature Based Interpretability (FBI) that uses the output values of a specific intermediate layer to generate explanations for the working of a point-cloud-based classification network. They use the per point features computed by the network with an architecture similar to the one shown in Figure 17 before they are subjected to a max pooling operation. The L1 norm of these features is computed and is used as an importance indicator of the individual points in the input point cloud data. We classify this method as a model-specific XAI method as it relies on the per point feature computation for generating explanations.

Katageri et al. [22] proposed a representation learning (or embedding) mechanism that learns Wasserstein embeddings from 3D point cloud data. The network utilizes MLP layers in the initial stages to capture features in the input data and these layers are followed by a max pooling operation. The output of the max pooling layer indicates the set of points (known as critical points) that contribute to the global embeddings. The authors visualize these critical points to provide an explanation to the embedding process.

Romaneli et al. [49] used attention visualization to provide explanations for the working of point-cloud-based transformer model apart from the gradient-based explanation described in subsection V-B. Here the authors use the parameters (or weights) and activations of the intermediate layers to compute attention scores for the input point cloud patches and thus indicating their importance for the output value.

The use of activations of intermediate layers is one of the most widely used methods to understand the working of AI models. This is also highlighted in the point-cloud-based XAI as 13 out of the 45 surveyed papers use this mechanism to generate explanations for the working of the AI models. Two of the surveyed papers ([8], [77]) that attempted to analyze the point-cloud-based models during the training process also used this mechanism to generate explanations.

F. COMPARISON IN LATENT SPACE

Comparison of data in latent space can provide some important insights into how an AI model learns to distinguish between input instances belonging to different classes. It also helps model developers to determine input samples that are hard to classify for the AI model. In our survey, we identified three papers that utilize this mechanism to provide an explanation for the decision made by the AI model.

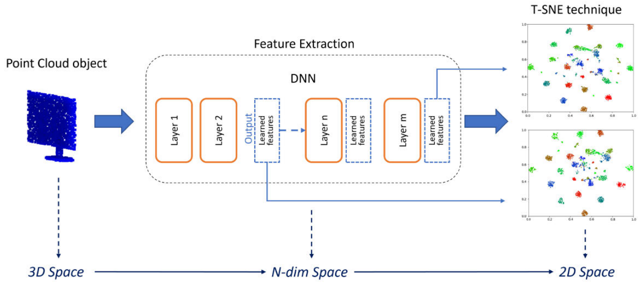


FIGURE 22. **Comparison of input data instances in latent space using dimensionality reduction techniques. The points are colour-coded based on the class these data instances belong to. Image taken from [35].**

Matrone et al. [35] proposed BubblEX that utilizes, in addition to the gradient-based method (explained in subsection V-B), the comparison of input instances in latent space as an XAI mechanism. This mechanism is illustrated in Figure 22. The features learned by the model from the input point cloud data are extracted from the hidden (or intermediate) layers of the AI model. These features can be extracted at any intermediate layer. However, AI models tend to capture complex features in the deeper layers while the initial layers capture basic features. To address the issue of high-dimensionality of the learned features, the authors make use of dimensionality reduction techniques such as t-SNE [71] and UMAP [36] for visual analysis in 2D. Each point in the 2D plot represents one input data instance. This comparison of input data instances in latent space allows us to understand if and how well the network is discriminating different classes of the dataset within its architecture. The authors extended BubblEX to understand the working of AI models working on heritage point cloud data intended for semantic segmentation tasks [34]. They used both t-SNE and UMAP methods for dimension reduction to visualize the data in 2D. However, this visualization differs from the original implementation as it visualizes the learned features of only one input instance. Here, each point in the 2D plot actually represents one of the points in the input data and the points are colour-coded based on the prediction or the segmentation label of the input data. The clusters indicate how successful the model has been in identifying objects of different classes in the input heritage point cloud data.

In addition to the above mentioned two papers, Beetz et al. [5] also utilized the mechanism of comparing input data instances in latent space to provide explainability for the multi-objective (reconstruction and classification) AI model. The latent space encodings of the input point cloud data are computed using the encoder branch of the trained model. These latent space encodings are then visualized on a 2D plot using the Laplacian eigenmap algorithm, a non-linear dimensionality reduction method, that reduces the dimensions of these encodings to two (see Figure 23).

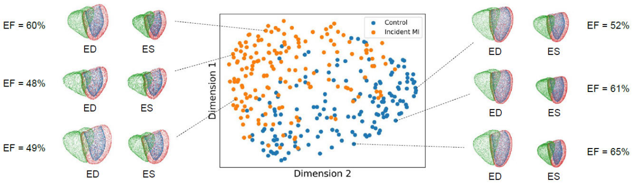


FIGURE 23. **Comparison of the end-diastolic (ED) and end-systolic (ES) input instances (shown with their corresponding left ventricular ejection fraction (EF) values) using their latent space features. Image taken from [5].**

Su et al. [59] proposed a deep learning framework that extracts interpretable latent representation of pose and bodytype information from the human point cloud data. The framework is an encoder-decoder architecture with two additional encoders (E_p, E_b) that extract specific latent features corresponding to pose and bodytype respectively from the output of the primary encoder, E. This is enforced by deploying a classifier to each of the branches of E_p and E_b to enforce the learning of these specific representations. To verify the representation learned by the network, the authors visualize the latent features produced by E_p and E_b in 2D using t-SNE. The visualization shows that data points corresponding to data belonging to the same categories or representing similar poses lie close to each other.

In our survey work, only four of the surveyed papers used the methodology of comparing latent features for model analysis. This is probably due to the inability of this mechanism to provide explanations for a single input instance and indicate important features in it. However, it helps in understanding how well the network has learned to distinguish input instances belonging to different classes. It can also be used to understand how well the intermediate layers in a neural network distinguish input instances belonging to different classes.

VI. SURVEY INSIGHTS

In this section, we provide valuable insights that can be observed from the classification of the collected literature. Since the focus of this survey paper is on the point-cloud-based XAI methods, we first look into the *type of point cloud* class in Table 2. We observe that a large number of literature (62%) is dedicated to the explanation of AI models working on point clouds representing 3D objects such as plane, table and chair (see Figure 24). This trend may be attributed to the development of methods and hardware that make the generation and measurement of point clouds from 3D models easier.

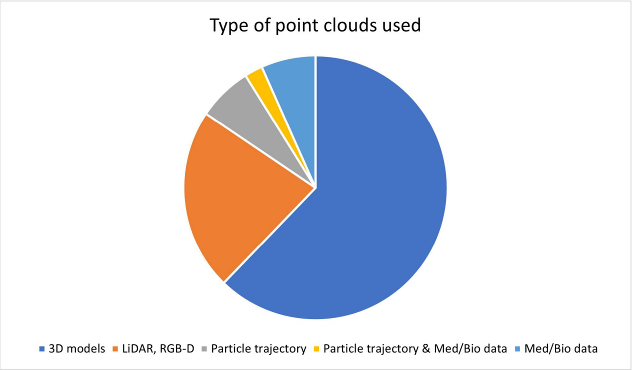


FIGURE 24. Representation of the collected literature based on the type of point cloud data used for analysis.

This is followed by the point clouds that represent outdoor areas such as train lines and archaeological sites. However, it is interesting to note that there are four papers (as indicated in Table 2) that deal with point clouds in the field of particle physics. These point clouds represent charged particle trajectories which is an interesting type of point cloud data compared to the more general point cloud data that represent 3D objects.

A. TRADE-OFF BETWEEN PERFORMANCE AND INTERPRETABILITY

The exponential rise in the use of AI models to perform numerous tasks is due to their ability to learn complex features and thereby leading to better performance. But this ability to learn complex features needs larger models and this comes with a drawback which is the lack of transparency. Therefore, the use of AI models involve a trade-off between performance and interpretability. This depends on the type of task being performed and the margin of error allowed when performing this task. Tasks that involve a very low margin of error are generally critical tasks involving high risk factors. This makes explainability an important requirement for application in real-world. In our surveyed literature, we observed that there are only six papers that proposed fully or partially transparent models. The remaining literature cater to the explanation of complex models which highlights the trade-off between performance and interpretability during the development of these AI models. In addition, it also underlines the need for more XAI work focused on explaining complex AI models.

B. ADAPTATIONS OF EXISTING XAI METHODS

One of the important observations in our list of surveyed papers is the adaptation of many existing XAI methods that were originally developed for AI models working on different datatypes. This makes sense given the fact that these XAI methods are well-known XAI methods providing explanations for models working on datatypes such as image data. In our survey, 12 of the surveyed papers ([14], [34], [35], [50], [21], [41], [42], [53], [3], [44], [54], [65]) adapted well-known XAI methods such as SHAP, integrated gradients, SmoothGrad and LIME.

C. VISUALIZATION

One of the major difficulties in analyzing the results of XAI methods is the lack of ground truth. This also makes it hard to compare the results of XAI methods to determine the method providing better explainability. Therefore, the quality of results produced by these XAI methods rely a lot on the human interpretation. Visualization is an important tool used for this purpose. It helps humans in exploring the output provided by these methods and finding hidden insights. Some of the prominent examples of XAI works utilizing visualization tools are the ones that produce saliency maps (for example: Grad-CAM [51] and Integrated gradients [60]) to help humans understand the decision-making process of a model. A large section of the surveyed literature makes use of visualization tools to highlight feature learning and feature importance as shown in Table 1. Since the input data is point clouds, most of these works use 3D visualization to provide explanations. One of the interesting works that uses interactive tools is by the authors of [41] who proposed an interactive visual analysis tool that allows the users to explore a large amount of XAI data. However, it is important to note that some XAI mechanisms do not completely rely on visualization. One such mechanism is the transparency mechanism where the functioning of individual layers in the model architecture is transparent and thus, easily understandable without the need for visualization. However, in some cases, authors do use visualization to highlight the features detected by these layers to the users.

D. USE OF DIMENSIONALITY REDUCTION TECHNIQUES

The use of high dimensional data makes it possible for the data to carry very complex information. However, it becomes more and more difficult to explore such data due to difficulty in visualizing high dimensional data. It also leads to high computational costs when used in algorithms such as ML-based algorithms. Point clouds are already represented in 3D and additional features associated with individual points in it lead to difficulty in visualizing them in 3D space. Thus, many of the works involve using dimensionality reduction techniques to address these issues. In the surveyed papers, seven of the papers make use of dimensionality reduction techniques to address these issues. Three of them use it to visualize the latent features of input point cloud instances in 2D [5], [34], [35], [59]. Zhang et al. [80] utilized the dimensionality reduction techniques to reduce the number of features computed by previous layers in the ML network. Mulawade et al. [41] and Kuriyal et al. [23] used dimensionality reduction techniques to visualize high-dimensional saliency attributions in 2D and 3D.

E. MODEL-RELATED OBSERVATION

A lot of work in AI is related to supervised learning algorithms that learn to produce an output by training on the data and corresponding ground truth values. This is also reflected in the XAI methods developed for point clouds related XAI works as shown in Table 2. Only two papers (i.e. [42] which is further extended in [41]) deal with XAI for RL models working on point cloud data. In addition, we also observe a similar trend (with respect to the research work related to the supervised learning algorithms) in the type of model task involved in generating explanations with 31 out of the 45 surveyed papers containing the classification-based models compared other task based models. This observation reflects the amount of work going into the supervised learning and classification oriented AI model development.

Regression is one of the basic and widely used techniques in ML. However, there are not many works in point-cloud-based AI and XAI that make use of this task. Majority of the works cater to the classification and segmentation tasks. This can be observed in Table 2 for point-cloud-based XAI works. In our surveyed papers, only one paper (i.e. [26]) attempted to provide explanations for a point-cloud-based model performing regression task.

VII. CONCLUSION

In this paper, we provided an overview of the recent developments in the field of XAI focused on point cloud data. We classified the literature based on multiple criteria making it easier for the readers to distinguish XAI methods and identify specific works that fit their use case. We explained the methods in detail and highlighted the fundamental mechanisms employed by these methods to explain AI and ML models. In addition, we also provided interesting insights such as the importance of visualization for interpretation and the need for dimensionality reduction techniques to address high dimensionality problems in these papers. We believe that there is a high potential in the field of point-cloud-based XAI as point cloud data becomes more mainstream in industrial applications. Moreover, XAI, in general, is an important sub-field of AI. However, as per our observation, there are no dedicated conferences or journals for XAI. We hope the coming years will see more focus on XAI and thus, see more dedicated journals and conferences that delve deeper into the field.

REFERENCES

[1] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[2] N. I. Arnold, P. Angelov, and P. M. Atkinson, "An improved explainable point cloud classifier (XPCC)," *IEEE Trans. Artif. Intell.*, vol. 4, no. 1, pp. 71–80, Feb. 2023, doi: 10.1109/TAI.2022.3150647.

[3] M. E. Atik, Z. Duran, and D. Z. Seker, "Explainable artificial intelligence for machine learning-based photogrammetric point cloud classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 5834–5846, 2024, doi: 10.1109/JSTARS.2024.3370159.

[4] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140, doi: 10.1371/journal.pone.0130140.

[5] M. Beetz, A. Banerjee, and V. Grau, "Multi-objective point cloud autoencoders for explainable myocardial infarction prediction," *Medical Image Computing and Computer Assisted Intervention—MICCAI 2023*, H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, Eds., Cham, Switzerland: Springer, 2023, pp. 532–542, doi: 10.1007/978-3-031-43895-0_50.

[6] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, Jan. 2021, doi: 10.1613/jair.1.12228.

[7] E. Camuffo, D. Mari, and S. Milani, "Recent advancements in learning algorithms for point clouds: An updated overview," *Sensors*, vol. 22, no. 4, p. 1357, Feb. 2022, doi: 10.3390/s22041357.

[8] Y. Cao, M. Previtali, and M. Scaioni, "Understanding 3D point cloud deep neural networks by visualization techniques," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLIII-B2-2020, pp. 651–657, Aug. 2020, doi: 10.5194/isprs-archives-xliii-b2-2020-651-2020.

[9] A. Chaddad, J. Peng, J. Xu, and A. Bouridane, "Survey of explainable AI techniques in healthcare," *Sensors*, vol. 23, no. 2, p. 634, Jan. 2023, doi: 10.3390/s23020634.

[10] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[11] F. Charmet, H. C. Tanuwidjaja, S. Ayoubi, P.-F. Gimenez, Y. Han, H. Jmila, G. Blanc, T. Takahashi, and Z. Zhang, "Explainable artificial intelligence for cybersecurity: A literature survey," *Ann. Telecommun.*, vol. 77, nos. 11–12, pp. 789–812, Dec. 2022, doi: 10.1007/s12243-022-00926-7.

[12] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A survey of the state of explainable AI for natural language processing," 2020, *arXiv:2010.00711*.

[13] F. Di Martino and F. Delmastro, "Explainable AI for clinical and remote health applications: A survey on tabular and time series data," *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 5261–5315, Jun. 2023, doi: 10.1007/s10462-022-10304-3.

[14] D. Dworak and J. Baranowski, "Adaptation of grad-CAM method to neural network architecture for LiDAR pointcloud object detection," *Energies*, vol. 15, no. 13, p. 4681, Jun. 2022, doi: 10.3390/en15134681.

[15] T. Feng, R. Quan, X. Wang, W. Wang, and Y. Yang, "Interpretable3D: An ad-hoc interpretable classifier for 3D point clouds," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 2, pp. 1761–1769, doi: 10.1609/aaai.v38i2.27944.

[16] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and LiDAR in autonomous vehicle environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018, doi: 10.1109/TII.2018.2822828.

[17] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021, doi: 10.1109/TPAMI.2020.3005434.

[18] A. Gupta, S. Watson, and H. Yin, "3D point cloud feature explanations using gradient-based methods," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9206688.

[19] N. F. Heide, E. Müller, J. Petereit, and M. Heizmann, "X3SEG: Model-agnostic explanations for the semantic segmentation of 3D point clouds with prototypes and criticism," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3687–3691, doi: 10.1109/ICIP42928.2021.9506624.

[20] S. Huang, B. Zhang, W. Shen, and Z. Wei, ''A CLAIM approach to understanding the PointNet,'' in *Proc. 2nd Int. Conf. Algorithms, Comput. Artif. Intell.*, New York, NY, USA, Dec. 2019, pp. 97–103, doi: 10.1145/3377713.3377740.

[21] X. Huang, W. Qu, Y. Zuo, Y. Fang, and X. Zhao, ''IMFNet: Interpretable multimodal fusion for point cloud registration,'' *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12323–12330, Oct. 2022, doi: 10.1109/LRA.2022.3214789.

[22] S. Katageri, S. Sarkar, and C. Sharma, ''Metric learning for 3D point clouds using optimal transport,'' in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2024, pp. 552–560.

[23] A. Kuriyal and V. Kumar, ''Towards explainable LiDAR point cloud semantic segmentation via gradient based target localization,'' 2024, *arXiv:2402.12098*.

[24] A. Kyuroson, A. Koval, and G. Nikolakopoulos, ''Autonomous point cloud segmentation for power lines inspection in smart grid,'' *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11754–11761, 2023, doi: 10.1016/j.ifacol.2023.10.562.

[25] D. Lavado, ''Detection of power line supporting towers via interpretable semantic segmentation of 3D point clouds,'' NOVA School Sci. Technol., NOVA Univ. Lisbon, Lisbon, Portugal, Tech. Rep., 2022.

[26] E. Lavasa, C. Chadoulos, A. Siouras, A. E. Llana, S. R. Del Rey, T. Dalamagas, and S. Moustakidis, *Toward Explainable Metrology 4.0: Utilizing Explainable AI to Predict the Pointwise Accuracy of Laser Scanning Devices in Industrial Manufacturing*. Cham, Switzerland: Springer, 2024, pp. 479–501, doi: 10.1007/978-3-031-46452-2_27.

[27] M. Y. Levi and G. Gilboa, ''Robustifying point cloud networks by refocusing,'' 2023, *arXiv:2308.05525*.

[28] A. Liang, H. Zhang, and H. Hua, ''Point cloud saliency maps based on non-contribution factors,'' in *Proc. 3rd Int. Conf. Control, Robot. Intell. Syst.*, New York, NY, USA, Aug. 2022, pp. 194–198, doi: 10.1145/3562007.3562045.

[29] A. Liberati, D. G. Altman, J. Tetzlaff, C. Mulrow, P. C. Gøtzsche, J. P. Ioannidis, M. Clarke, P. J. Devereaux, J. Kleijnen, and D. Moher, ''The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration,'' *Ann. Internal Med.*, vol. 151, no. 4, p. 65, 2009, doi: 10.1136/bmj.b2700.

[30] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, ''Explainable AI: A review of machine learning interpretability methods,'' *Entropy*, vol. 23, no. 1, p. 18, Dec. 2020, doi: 10.3390/e23010018.

[31] B. Liu, ''3D point cloud-based visual prediction of ICU mobility care activities,'' *Proc. 3rd Mach. Learn. Healthcare Conf.*, vol. 85, Aug. 2018, pp. 17–29. [Online]. Available: https://proceedings.mlr.press/v85/liu18a.html

[32] S. M. Lundberg and S. Lee, ''A unified approach to interpreting model predictions,'' in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4765–4774. https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[33] H. Maisonneuve, ''COVID-19 as a source of poor publications,'' *Joint Bone Spine*, vol. 89, no. 6, Nov. 2022, Art. no. 105427, doi: 10.1016/j.jbspin.2022.105427.

[34] F. Matrone, A. Felicetti, M. Paolanti, and R. Pierdicca, ''Explaining AI: Understanding deep learning models for heritage point clouds,'' *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. X-M-1-2023, pp. 207–214, Jun. 2023, doi: 10.5194/isprs-annals-x-m-1-2023-207-2023.

[35] F. Matrone, M. Paolanti, A. Felicetti, M. Martini, and R. Pierdicca, ''BubblEX: An explainable deep learning framework for point-cloud classification,'' *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6571–6587, 2022, doi: 10.1109/JSTARS.2022.3195200.

[36] L. McInnes, J. Healy, N. Saul, and L. Großberger, ''UMAP: Uniform manifold approximation and projection,'' *J. Open Source Softw.*, vol. 3, no. 29, p. 861, Sep. 2018, doi: 10.21105/joss.00861.

[37] S. Miao, Y. Luo, M. Liu, and P. Li, ''Interpretable geometric deep learning via learnable randomness injection,'' 2022, *arXiv:2210.16966*.

[38] A. Micheletti, ''A new paradigm for artificial intelligence based on group equivariant non-expansive operators,'' *Eur. Math. Soc. Mag.*, vol. 128, no. 128, pp. 4–12, Apr. 2023, doi: 10.4171/mag/133.

[39] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, ''Explainable artificial intelligence: A comprehensive review,'' *Artif. Intell. Rev.*, vol. 55, no. 5, pp. 3503–3568, Jun. 2022, doi: 10.1007/s10462-021-10088-y.

[40] F. Mokhtar, ''Explaining machine-learned particle-flow reconstruction,'' 2021, *arXiv:2111.12840*.

[41] R. N. Mulawade, C. Garth, and A. Wiebel, ''Visual analytics system for understanding DeepRL-based charged particle tracking,'' *Vis. Comput.*, pp. 1–24, Mar. 2024, doi: 10.1007/s00371-024-03297-3.

[42] R. N. Mulawade, C. Garth, and A. Wiebel, ''Saliency clouds: Visual analysis of point cloud-oriented deep neural networks in DeepRL for particle physics,'' in *Machine Learning Methods in Visualisation for Big Data 2022*. Eindhoven, The Netherlands: The Eurographics Association, 2022, doi: 10.2312/mlvis.20221069.

[43] K. Pearson, ''LIII. On lines and planes of closest fit to systems of points in space,'' *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901.

[44] S. Pölsterl, C. Aigner, and C. Wachinger, ''Scalable, axiomatic explanations of deep Alzheimer's diagnosis from heterogeneous data,'' in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2021*. Cham, Switzerland: Springer, 2021, pp. 434–444, doi: 10.1007/978-3-030-87199-4_41.

[45] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, ''PointNet: Deep learning on point sets for 3D classification and segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85, doi: 10.1109/CVPR.2017.16.

[46] M. Raynaud, ''Impact of the COVID-19 pandemic on publication dynamics and non-COVID-19 research production,'' *BMC Med. Res. Methodol.*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12874-021-01404-9.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin, '''Why should i trust you?': Explaining the predictions of any classifier,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[48] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, ''Explainable artificial intelligence (XAI) on timeseries data: A survey,'' 2021, *arXiv:2104.00950*.

[49] I. Romanelis, V. Fotis, K. Moustakas, and A. Munteanu, ''ExpPoint-MAE: Better interpretability and performance for self-supervised point cloud transformers,'' *IEEE Access*, vol. 12, pp. 53565–53578, 2024, doi: 10.1109/ACCESS.2024.3388155.

[50] M. Schwegler, C. Müller, and A. Reiterer, ''Integrated gradients for feature assessment in point cloud-based data sets,'' *Algorithms*, vol. 16, no. 7, p. 316, Jun. 2023, doi: 10.3390/a16070316.

[51] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ''Grad-CAM: Visual explanations from deep networks via gradient-based localization,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[52] L. S. Shapley, ''A value for n-person games,'' in *Contributions to the Theory of Games*, vol. 2. Princeton, NJ, USA: Princeton University Press, 1953, pp. 307–318, doi: 10.1515/9781400881970-018.

[53] W. Shen, Q. Ren, D. Liu, and Q. Zhang, ''Interpreting representation quality of DNNs for 3D point cloud processing,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8857–8870.

[54] W. Shen, Z. Wei, Q. Ren, B. Zhang, S. Huang, J. Fan, and Q. Zhang, ''Interpretable rotation-equivariant quaternion neural networks for 3D point cloud processing,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3290–3304, May 2024, doi: 10.1109/TPAMI.2023.3346383.

[55] Y. Shen, C. Feng, Y. Yang, and D. Tian, ''Mining point cloud local structures by kernel correlation and graph pooling,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4548–4557, doi: 10.1109/CVPR.2018.00478.

[56] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, ''Smooth-Grad: Removing noise by adding noise,'' 2017, *arXiv:1706.03825*.

[57] S. Spina, K. Debattista, K. Bugeja, and A. Chalmers, ''Point cloud segmentation for cultural heritage sites,'' in *Proc. VAST Int. Symp. Virtual Reality, Archaeology Intell. Cultural Heritage*. Eindhoven, The Netherlands: Eurographics Association, 2011, doi: 10.2312/VAST/VAST11/041-048.

[58] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, ''Striving for simplicity: The all convolutional net,'' 2014, *arXiv:1412.6806*.

[59] F.-G. Su, C.-S. Lin, and Y. F. Wang, ''Learning interpretable representation for 3D point clouds,'' in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 7470–7477, doi: 10.1109/ICPR48806.2021.9412440.

[60] M. Sundararajan, A. Taly, and Q. Yan, ''Axiomatic attribution for deep networks,'' in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, Sydney, NSW, Australia, 2017, pp. 3319–3328.

[61] S. A. Taghanaki, K. Hassani, P. K. Jayaraman, A. H. Khasahmadi, and T. Custis, ''PointMask: Towards interpretable and bias-resilient point cloud processing,'' 2020, *arXiv:2007.04525*.

[62] H. Tan, "Fractual projection forest: Fast and explainable point cloud classifier," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4240–4249, doi: 10.1109/WACV56688.2023.00422.

[63] H. Tan, "Visualizing global explanations of point cloud DNNs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4730–4739, doi: 10.1109/WACV56688.2023.00472.

[64] H. Tan, "Flow AM: Generating point cloud global explanations by latent alignment," 2024, *arXiv:2404.18760*.

[65] H. Tan and H. Kotthaus, "Surrogate model-based explainability methods for point cloud NNs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2927–2936, doi: 10.1109/WACV51458.2022.00298.

[66] H. Tan and H. Kotthaus, "Explainability-aware one point attack for point cloud neural networks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4570–4579, doi: 10.1109/WACV56688.2023.00456.

[67] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6410–6419, doi: 10.1109/ICCV.2019.00651.

[68] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Nov. 2021, doi: 10.1109/TNNLS.2020.3027314.

[69] L. Truong-Hong and D. F. Laefer, "Quantitative evaluation strategies for urban 3D model generation from remote sensing data," *Comput. Graph.*, vol. 49, pp. 82–91, Jun. 2015, doi: 10.1016/j.cag.2015.03.001.

[70] D. Valenzuela-Urrutia, R. Muñoz-Riffo, and J. Ruiz-del-Solar, "Virtual reality-based time-delayed haptic teleoperation using point cloud data," *J. Intell. Robotic Syst.*, vol. 96, nos. 3–4, pp. 387–400, Dec. 2019, doi: 10.1007/s10846-019-00988-1.

[71] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

[72] F. M. Verburg. (Feb. 2022). *Exploring Explainability and Robustness of Point Cloud Segmentation Deep Learning Model By Visualization*. [Online]. Available: http://essay.utwente.nl/89440/

[73] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.

[74] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quart.*, vol. 26, no. 2, pp. 8–23, 2002. [Online]. Available: http://www.jstor.org/stable/4132319

[75] L. Wells and T. Bednarz, "Explainable AI and reinforcement learning—A systematic review of current approaches and trends," *Frontiers Artif. Intell.*, vol. 4, May 2021, doi: 10.3389/frai.2021.550030.

[76] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[77] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, USA, Jun. 2018, pp. 206–215, doi: 10.1109/CVPR.2018.00029.

[78] M. Y. Levi and G. Gilboa, "Fast and simple explainability for point cloud networks," 2024, *arXiv:2403.07706*.

[79] B. Zhang, S. Huang, W. Shen, and Z. Wei, "Explaining the PointNet: What has been learned inside the PointNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2019, pp. 71–74.

[80] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop: An explainable machine learning method for point cloud classification," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1744–1755, Jul. 2020, doi: 10.1109/TMM.2019.2963592.

[81] B. Zhao, X. Hua, K. Yu, W. Tao, X. He, S. Feng, and P. Tian, "Evaluation of convolution operation based on the interpretation of deep learning on 3-D point cloud," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5088–5101, 2020, doi: 10.1109/JSTARS.2020.3020321.

[82] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "PointCloud saliency maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1598–1606.

RAJU NINGAPPA MULAWADE received the Bachelor of Engineering (B.E.) degree in mechanical engineering from the Sri Jayachamarajendra College of Engineering, Mysore, India, in 2013, and the M.S. degree in mechatronics from Technische Universität (TU), Hamburg, in 2021. He is currently pursuing the Ph.D. degree in computer science with the Hochschule Worms University of Applied Sciences and Rheinland-Pfälzische Technische Universität (RPTU) Kaiserslautern-Landau. His research interests include artificial intelligence (AI) with a focus on explainable AI (XAI) and visual analytics.

CHRISTOPH GARTH received the Ph.D. degree in computer science from Technische Universität (TU), Kaiserslautern, in 2007. After four years as a Postdoctoral Researcher with the University of California, Davis, he rejoined RPTU Kaiserslautern-Landau, where he is currently a Full Professor of computer science. His research interests include largescale data analysis and visualization, situ visualization, topology-based methods in visualization, and interdisciplinary applications of visualization.

ALEXANDER WIEBEL received the Ph.D. degree from Universität Leipzig, in 2008, for research on flow visualization. From 2013 to 2015, he was a Professor of visual computing with Coburg University. Currently, he is a Professor with Hochschule Worms University of Applied Sciences, where he is also co-heading the research group User Experience and Visualization (UX-Vis) and the Deputy Scientific Director of the Center for Technology and Transfer (ZTT). He is a Postdoctoral Researcher with the Max Planck Institute for Human Cognitive and Brain Sciences, and Zuse Institute Berlin (ZIB), he conducted research in interactive visualization of 3D MRI data with a focus on intuitive selection of structures in direct volume renderings. During this time, he lectured with Universität Leipzig and Freie Universität Berlin. His current research interests include scientific visualization, XAI, mixed, augmented, and virtual reality.