Computers & Graphics 102 (2022) 502–520

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Survey Paper

A survey of visual analytics for Explainable Artificial Intelligence methods

Gulsum Alicioglu [a,*], Bo Sun [b]

[a] Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, 08028, USA
[b] Department of Computer Science, Rowan University, Glassboro, NJ, 08028, USA

ARTICLE INFO

Article history:
Received 10 February 2021
Received in revised form 25 August 2021
Accepted 6 September 2021
Available online 10 September 2021

Keywords:
Explainable Artificial Intelligence
Interpretable neural networks
Visual analytics
Black-box models

ABSTRACT

Deep learning (DL) models have achieved impressive performance in various domains such as medicine, finance, and autonomous vehicle systems with advances in computing power and technologies. However, due to the black-box structure of DL models, the decisions of these learning models often need to be explained to end-users. Explainable Artificial Intelligence (XAI) provides explanations of black-box models to reveal the behavior and underlying decision-making mechanisms of the models through tools, techniques, and algorithms. Visualization techniques help to present model and prediction explanations in a more understandable, explainable, and interpretable way. This survey paper aims to review current trends and challenges of visual analytics in interpreting DL models by adopting XAI methods and present future research directions in this area. We reviewed literature based on two different aspects, model usage and visual approaches. We addressed several research questions based on our findings and then discussed missing points, research gaps, and potential future research directions. This survey provides guidelines to develop a better interpretation of neural networks through XAI methods in the field of visual analytics.

1. Introduction

Machine learning (ML) techniques have achieved impressive performance in various domains such as medicine, finance, and autonomous vehicle systems with advances in computing power and technologies [1,2]. Neural Network (NN), as a sub-branch of ML, has become a powerful technique in finding complex patterns in high-dimensional datasets and providing high prediction accuracy in many domains [3]. However, NN-based models have a complex structure, which makes it difficult for them to be interpreted and understood. NNs are considered black-box models since their inner working and decision-making mechanisms are not understandable by a human. This reveals one of the most important issues in black-box models: transparency and explainability [4].

End-users often want to understand how a classifier makes predictions, particularly in sensitive domains, such as healthcare, transportation, defense, and finance, where decision making often has a critical impact. Explaining how predictions are made by ML models by clarifying their working mechanisms would increase trustworthy of ML models. To address this important need, interpretable ML algorithms has been developed rapidly in understanding the inner working mechanisms of black-box models [5]. One of the most important efforts is the development of a re-emerging field in eXplainable Artificial Intelligence (XAI) [6]. According to Defense Advanced Research Projects Agency (DARPA) technical report [6], XAI is defined as “a suite of machine learning techniques that enables human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners”. Although interpretability and explainability are often used interchangeably by the ML community, there are slight differences in the definition of interpretable ML and explainable AI. Miller [7] defines interpretability as “the degree to which an observer can understand the cause of a decision” and equates interpretability and explainability definitions. From the ML context, interpretability can be defined as understanding how the decision/prediction is given by machine learning algorithms with reasoning. The term explainability is more related to the internal working mechanisms of black-box models. Therefore, XAI reveals the internal functioning of black-box models and the rationale behind the decisions through various methods. While domain experts who are inexperienced in ML often want to understand through reasoning and cause–effect relationship why a certain decision has been made, ML scientists focus on the internal working mechanisms of ML models and try to understand how their components contribute to certain predictions. XAI aims to help end-users and domain experts to gain insight into how black-box models make predictions. It also helps

* Corresponding author.
E-mail addresses: alicio87@rowan.edu (G. Alicioglu), sunb@rowan.edu (B. Sun).

https://doi.org/10.1016/j.cag.2021.09.002

ML scientists with the model development process by explaining the decision-making process of the black-box models.

Visual analytics (VA) is inherent way to represent the data/model understandably, particularly to those who are inexperienced in ML. VA has been often used in providing interpretable ML models by understanding [1,5], diagnosing [8,9], and steering [10] the model and underlying data through an interactive visual interface. Combining the techniques of VA with XAI algorithms would present an ideal platform to clarify the black box structure of ML. However, there are only a few recent works combining VA with the current stage of XAI methods to provide explainable ML models to humans. So, we target the audience of this review to following groups: (1) VA scientists who would like to adopt XAI methods to interpret NN, (2) ML scientists, particularly in the field of XAI, who may need VA to interpret their work, and (3) end-users/domain experts who use NN for data classification and predications. This survey paper aims to find current trends and challenges of VA in interpreting black-box models by adopting XAI methods and present future research directions in this area. Within the study, we would like to discover and present how VA can support a better interpretation of NN models with XAI methods.

The explainability and interpretability of black-box models are recent hot topics, and many studies have been done in the field. Most of the studies have been focused on the interpretation of NNs due to their state-of-the-art performance in various domains. Therefore, we limit our study by reviewing papers that focus on the interpretation of NNs among black-box models. There are several literature reviews of XAI methods [4,11–14], and visual analytics on interpretable machine learning [2,15–20] respectively. However, to our knowledge, there is no literature review that focus on VA research combined with XAI methods. Such review will help to analyze potential future research directions to develop a better interpretation of neural networks through XAI methods in the field of visual analytics. Therefore, we reviewed 55 papers that contributed to the interpretation of NN models via visual analytics with and without XAI methods in terms of model usage and visual approach. ***Model usage*** refers to techniques that are used to explain NN models in the fields of VA and XAI respectively. The ***visual approach*** mainly focuses on analyzing how visualization techniques are used in data and architecture representations, performance analysis, and local and global explanations. Our main contributions are as follows:

- We present a review of VA research in interpreting deep learning with a focus on with and without adopting XAI methods.
- We reviewed the literature based on (1) model usage in visual interpretation and XAI algorithms respectively, and (2) visual approach where commonly used visual approaches are summarized.
- We highlight the current trends and limitations, and discuss future research directions of VA that adopts XAI for NN models.

The rest of the paper is organized as follows: Section 2 provides theoretical background about black-box models and XAI methods. Section 3 shows the methodology of this review. Section 4 reviews visual interpretation papers and Section 5 reviews visual-based XAI papers based on the model usage and visual approaches, respectively. Section 6 states the current trends and discusses future directions of VA for XAI. Section 7 concludes the paper.

2. Theoretical background

This section provides basic information about black-box models and definitions, concepts, and techniques related to XAI. The section emphasizes the need for explanations of black-box models through XAI methods.

Fig. 1. A representation of a black-box model.

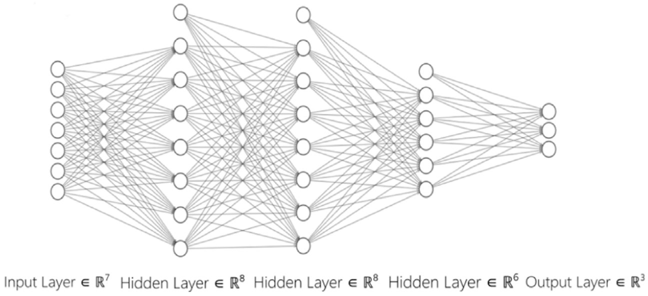


Fig. 2. A representation of a Deep Neural Network model [22].

2.1. Black-box models

ML algorithms work to enhance their performances, often by gradually improving a function to minimize a certain loss function. The loss function indicates how accurately it predicts outputs of unseen inputs [15]. ML algorithms such as regressions, decision trees, k-Nearest Neighbor, Bayesian classifier can be interpreted based on the model parameters, structure, and/or rules [4,12]. Although these methods provide interpretation, transparency, and explanations of their predictions, they lack good performance in terms of accuracy [4]. To tackle these performance issues, researchers have utilized more powerful models like neural networks, support vector machines (SVM), ensemble models, gradient boosted models, and boosted trees. These models have achieved good performance in terms of prediction accuracy in various domains [2]. Despite their impressive performance, it is difficult to interpret and explain what they learn during the training process, how these methods made a certain prediction, and their logic and inner working mechanisms [4,13]. Thus, these methods are considered as black-box models. Black-box models refer to a system that its internal functioning or logic is opaque and uninterpretable [14], as seen in Fig. 1. Especially in NN-based models, their hyperparameters, the number of hidden layers, and the number of neurons in each layer increase the complexity of their structure and opacities, and obstruct the interpretability. Therefore, it is essential to interpret these black-box models to domain users.

A deep NN model consists of multiple node layers that are inter-connected with adaptable weights [15], as seen in Fig. 2. Input layers that contain feature values forward data with random initial weights to the hidden layers. Hidden layers allow connections from input to output by performing nonlinear transformations via activation functions [21]. The values of the hidden nodes are the summation of the previous layer's nodes multiplied by their weights [3]. The output layer is the summation of the last hidden layer nodes, that are obtained through an activation function [15,21]. The obtained output is compared to the actual values and weights, and updated using optimization algorithms to minimize loss, which is called backpropagation [21].

Deep NNs are capable of modeling complex nonlinear systems through hidden layers and non-linear activation functions. Increasing the number of hidden layers enables the modeling of more complex relationships in the high dimensional data. They

Table 1
The Definitions of XAI.

Domain	Study	XAI definition
AI	Gunning and Aha [6]	XAI will create a suite of ML techniques that enables human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners.
	Rai [23]	XAI is the class of systems that provide visibility into how an AI system makes decisions and predictions and executes its actions.
	Díaz-Rodríguez and Pisoni [24]	XAI aims at making state-of-the-art opaque models more transparent and defends AI-based outcomes endorsed with a rational explanation, i.e., an explanation that has as target the non-technical users.
	Moradi and Samwald [25]	XAI refers to systems that try to explain how a black-box AI model produces its outcomes.
Survey of XAI	Arrieta et al. [26]	XAI is a system that produces details or reasons to make its functioning clear or easy to understand.
	Schoenborn and Althoff [27]	An XAI enables a user to learn a transparent, relevant, and justified information at the right time using an appropriate size.
	Das and Rad [14]	XAI is a field of AI that promotes a set of tools, techniques, and algorithms that can generate high-quality interpretable, intuitive, human-understandable explanations of AI decisions.
	Adadi and Berrada [19]	XAI term tends to refer to the movement, initiatives, and efforts made in response to AI transparency and trust concerns, more than to a formal technical concept.

also have the ability to extract features from the raw data automatically, which eliminates feature engineering tasks. Both the complex structure of the NN models and their modeling ability make them uninterpretable compared to other ML models. Due to this challenge, a considerable amount of XAI methods [28–30] have been developed to provide explanations for such black-box models.

2.2. Explainable Artificial Intelligence methods

Explainable Artificial Intelligence is a new research field that aims to provide understandable artificial intelligence (AI) results for end-users [19]. More specifically, XAI techniques aim to develop machine learning techniques in providing understandable, trustworthy, and explainable rationales for decisions made by black-box models [6,31]. The definition of XAI has been improved and modified heavily based on domain-specific applications, use-case scenarios, and expertise by researchers. Therefore, there is a common agreement that there is no consensus on the definition of XAI [12,19,27], as seen in Table 1. While some studies [6,27] mathematically define explainability, other studies [21,23,24] emphasize that explainability should include more non-technical concepts to increase human-understandability. These two aspects that differentiate the definition of XAI mainly depend on the domain of the application and research goals. Whereas ML and AI communities focus on explanations to understand decision mechanisms of models, sensitive domains need explanations as to how a decision has been made for trust and risk-related issues [12]. Therefore, while ML and AI communities seek answers to how models behave through mathematical methods, experts in application domains want to understand why a prediction is made by classifiers. As a result, various domains, end-user goals, research communities, and case studies brought different definitions and concepts for XAI [14].

A considerable amount of XAI methods have been developed lately to explain the inner working mechanisms of black-box models and their decisions. The XAI methods can be divided into three main categories based on *explanation level, implementation level,* and *model dependency.* The explanation level indicates whether an XAI technique focuses on the entire model or a single instance. The subcategories of the explanation level of an XAI method are named ***global level***, which focuses on the explainability of the entire model, and its working and decision-making mechanisms, and ***local level***, which explains the decisions of a model for a single instance or subpopulation. While some XAI methods, such as Bayesian Rule Lists (BRL) [32], Generalized Additive Models (GAM) [33], Distillation technique [34], provide global level explanations for a whole model and its decision-making mechanism, other methods like Local Interpretable Model-Agnostic Explanations (LIME) [28], Shapley Additive Explanations (SHAP) [35], Gradient-weighted Class Activation Mapping (Grad-CAM) [29], Deep Learning Important FeaTures (DeepLIFT) [36] provide local explanations for instance data.

Implementation level mainly has two subcategories: intrinsic and post-hoc explanations. ***Intrinsic explanations***, such as Bayesian Rule Lists [32] and Mean Decrease Impurity (MDI) [37], are provided by the model itself in terms of how a prediction has been made through the model parameters, decision trees, and/or rules through the methods.

Post-hoc explanations reveal the internal functioning and decision mechanisms of black-box models. Post-hoc explanations can be done for either a pre-trained model or when the training process of a model is complete. Since post-hoc explainers convert black-box models to interpretable models, many post-hoc XAI techniques have been developed such as Grad-CAM [29], Layer-wise Relevance Propagation (LRP) [30], LIME [28], Integrated Gradients [38], and Saliency Maps [39].

Model dependency consists of model-specific and model-agnostic explainers. ***Model-specific*** XAI techniques can be adopted to explain only for a specific type of algorithm. Intrinsic explanations serve as model-specific techniques, i.e., they cannot be used for any model without re-changing its explanation mechanism [14]. ***Model-agnostic*** explanations work any type of model and do not depend on the architecture of a model. Since most of the

Table 2
Classification of the most popular XAI techniques in explaining neural networks.

XAI method	Explanation level		Implementation level		Model dependency	
	Global	*Local*	*Intrinsic*	*Post hoc*	*Agnostic*	*Specific*
ANCHORS [40]		✔		✔	✔	
LIME [28]	✔	✔		✔	✔	
SHAP [35]		✔		✔	✔	
LRP [30]	✔	✔		✔	✔	
Grad-CAM [29]		✔		✔	✔	
Saliency Maps [39]		✔		✔	✔	
Integrated Gradients [38]		✔		✔	✔	
DeepLIFT [36]		✔		✔	✔	
Bayesian Rule Lists [32]	✔		✔			✔
Distillation [34]	✔			✔	✔	
GAM [33]	✔		✔			✔
Mean Decrease Impurity [37]	✔	✔	✔			✔
CAM [41]		✔		✔	✔	

model-agnostic explainers also provide post-hoc explanations, these methods have been often used due to their flexibility [19]. ANCHORS [40], LIME [28], LRP [30] and SHAP [35] are the most popular examples of model-agnostic post-hoc explainers. Table 2 shows the XAI categories with the most popular XAI techniques.

To support of the understanding of the XAI methods, we presented a visual explanation of the most popular XAI methods based on their features in Fig. 3. The visual explanation is illustrated on the basis of a NN architecture. The most popular XAI technique used in the VA field is LIME [28], as seen in Fig. 3a. LIME learns an interpretable model locally around the selected instance by using a surrogate model such as linear or tree-based models. LIME generates samples around the selected instance, represented in Fig. 3a as a black star, by perturbing, which makes changes on the selected instance to create similar instances, illustrated as gray stars. The surrogate model is trained using these generated instances to obtain explanations through feature importance. End-users can easily observe both positive and negative feature importance values that contribute to the prediction through visualization. LIME method provides explanations locally on how a certain prediction has been made by approximating any complex model via surrogate models. Similarly, SHAP [35] explains complex models locally around a selected instance by calculating feature importance for the corresponding prediction. However, LIME [28] and SHAP [35] work quite differently. SHAP [35] adopts strategies of game-theory and treats features as team members of a game. It calculates the relative contribution of each feature to the individual prediction, which corresponds to the contribution of each team member to win the game. SHAP requires more computational cost than the LIME method.

ANCHORS [40] provides explanations through a local region in the feature space, shown as a rectangle around the selected instance (black star) in Fig. 3b. It presents high-precision explanation rules. This local region in the feature space anchors a prediction locally so that it guarantees to obtain the corresponding prediction even if the rest of the feature values change. It can also explain non-linear decision boundaries that exist. It consumes computational resources intensively until obtaining a local region to explain the model. The main difference between LIME and ANCHORS is the way to produce their explanations. While the LIME method produces explanations through a linear model around an instance, ANCHORS generates IF-THEN rules providing comprehensible explanations around a local region.

LRP [30], Bayesian Rule Lists [32], Saliency Maps [39], and Model Distillation [34] are other popular techniques that are often adopted by VA in explaining NN models. LRP [30] , as seen in Fig. 3c, provides explanations for deep neural networks by propagating the prediction backward in the network, where inputs can be images, videos, or text. It displays the feature importance based on pixel-wise contribution. LRP redistributes relevance into the lower layers in proportion to the contribution of each input to neuron activation. Fig. 3c shows connections where the color opacity indicates higher importance based on these relevance scores. LRP has a key property called relevance conservation that guarantees the received relevance will be redistributed to the lower layers [30].

Bayesian Rule Lists [32] is a generative model that creates IF-THEN rule sets to provide explainability. It produces a posterior distribution for possible IF-THEN rules using the Bayesian framework. BRL uses pre-mined rules to reduce the model space and then learns these rule sets and their orders.

Saliency maps [39] are a gradient-based explanation method that calculates feature importance based on gradients, and visualize and highlight important pixels or words that contribute to the final decision in convolutional networks. The salience values indicate the contribution of the features concerning the selected instance, shown in Fig. 3d as heatmaps. It aims to find the regions that the convolutional networks are focused on. Highlighted pixels through heatmaps help to explain these regions to the end-users.

Class Activation Mapping (CAM) [41] is another gradient-based explanation method focusing on interpreting the predictions of Convolutional Neural Network (CNN) models. It uses global average pooling to obtain class activation maps in convolutional networks, as shown in Fig. 3e. This global average pooling helps to localize the discriminative image regions in explaining predictions. Through CAM, the important regions of the image can be identified by reflecting the weights to the convolutional feature maps [41]. CAM requires a particular CNN architecture that does not contain any fully connected layers.

Grad-CAM [29], as seen in Fig. 3f, is a generalized form of the CAM [41] method. It removes the disadvantage of the CAM method which is the need for a particular architecture. Grad-CAM can be used to generate explanations of any CNN-based networks without altering their architecture [29]. Grad-CAM is a highly class-discriminative method that uses the gradients of any class concept. It flows gradients into the final layer to create a heat-map that highlights the significant pixels for classifying the related concept.

Model Distillation method [34], as seen in Fig. 3g, is a model compression technique used to obtain an interpretable model. It transfers the knowledge from pre-trained DNN, called teacher

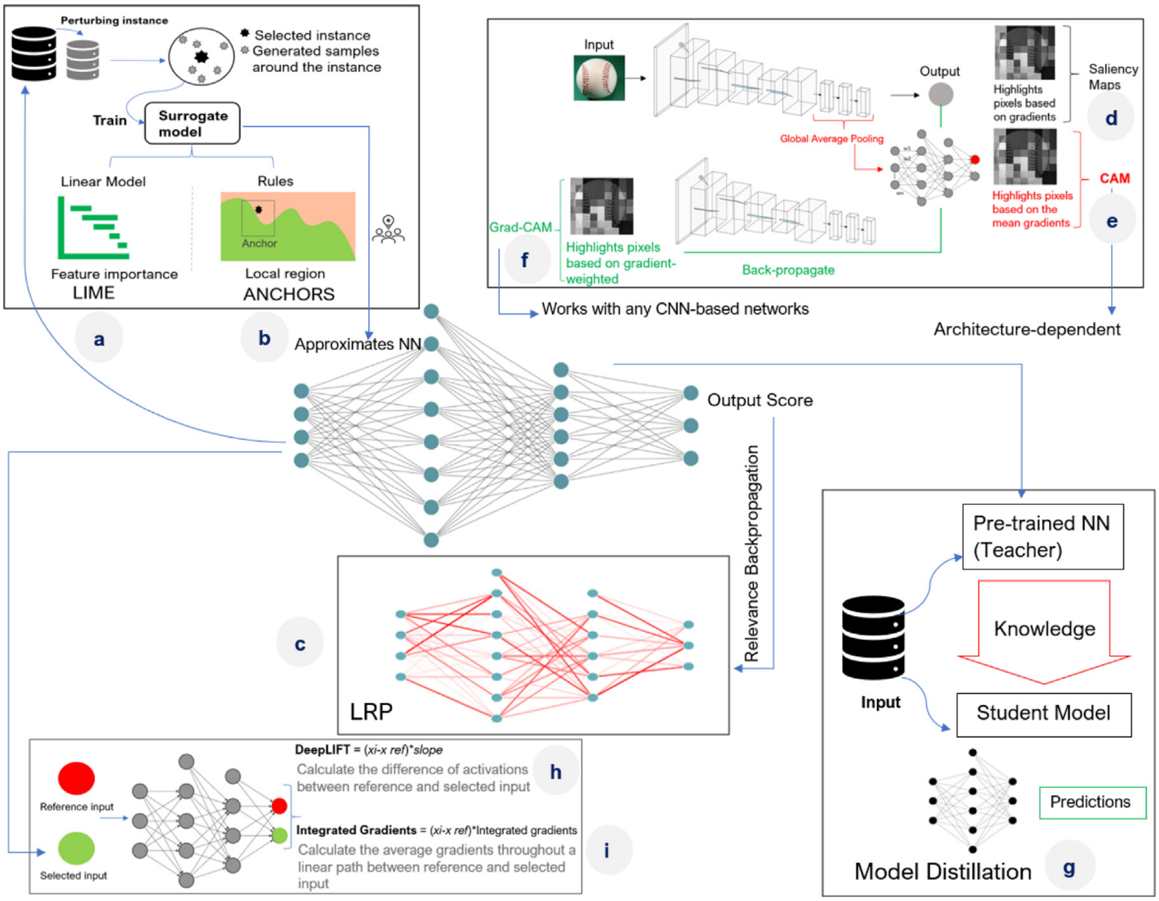

Perturbing instance
Selected instance
Generated samples around the instance
Train
Surrogate model
Linear Model
Rules
Anchor
Feature importance
LIME
Local region
ANCHORS
a
b
Approximates NN
Input
Output
Global Average Pooling
Highlights pixels based on gradients
Saliency Maps
d
Highlights pixels based on the mean gradients
CAM
e
Grad-CAM
f
Highlights pixels based on gradient-weighted
Back-propagate
Works with any CNN-based networks
Architecture-dependent
Output Score
Relevance Backpropagation
c
LRP
Pre-trained NN (Teacher)
Knowledge
Input
Student Model
Predictions
Model Distillation
g
Reference input
Selected input
DeepLIFT = (xi-x ref)*slope
Calculate the difference of activations between reference and selected input
h
Integrated Gradients = (xi-x ref)*Integrated gradients
Calculate the average gradients throughout a linear path between reference and selected input
i

Fig. 3. Visual explanations of XAI methods.

model, to small networks named as student. It learns the input–output behavior of the complex model and distills it to the interpretable model. Explanations are provided through the student model, which is interpretable. The student model needs to have a similar structure to the teacher model to provide good approximation [34]. It is used to simplify complex models into explainable models.

DeepLIFT [36], as shown in Fig. 3h, is an explanation method that computes the importance values of features in NN models. It propagates a 'reference input' which is generally a default or neutral input through the network to obtain reference output. The method compares activations of neurons to reference activation, then allocates importance score according to the difference between actual and reference output.

Integrated Gradients [38], as seen in Fig. 3i, is another explanation method that provides an interpretation of NN models through feature importance. It stands out with its simplicity and easy-to-use properties. It requires no modification to the architecture of the original deep NNs. It needs a few calls to the standard gradient operator to compute feature attribution by calculating the average gradients throughout a linear path between a selected input and baseline input.

Since GAM [33] and Mean Decrease Impurity [37] methods are model-specific, they are less preferred by VA scientists. GAM [33] is a linear model that aggregates unknown smooth functions. It provides feature importance by inferring the smooth functions and justify how these importance values affect the related prediction. MDI is used to obtain feature importance and split point for each node in tree-based models. It calculates the mean decrease impurity of features.

3. Methodology

This section presents the paper selection process and defines the strategies to classify the papers for our review.

3.1. Paper selection

We conducted a comprehensive paper selection using ScienceDirect, Scopus, IEEE, and Google Scholar respectively. To reach relevant papers, we searched several keywords, such as "explainable artificial intelligence", "visual analytics", "visualization", "interpretable machine learning", "black-box models", "neural network", and "deep learning models", and their different combinations. While searching for papers and reviewing the literature, we followed DARPA's [6] XAI definition the most. Using manual review process, we considered titles and keywords to identify the relevant papers and eliminated the irrelevant papers for the first round. Then we read abstracts of the papers to confirm the relevance. In any case that we are not sure about the relevance of a paper based on its title and abstract, we would go through the paper to finalize the selection. We categorized papers as survey papers, method papers, and implementation papers in the selection process. Survey papers cover reviews in the field of VA and XAI. Method papers include mainly XAI methods, their mathematical background, and application areas. We benefited from survey and method papers to establish a good background about XAI definitions and methods as seen in Section 2.2, then define review categories in the XAI and VA. VA papers with and without adopting XAI methods were included in the category of implementation papers since they provide

Table 3
The selected papers through reviewing process.

Paper categories			Papers
Interpretation of ML via VA	Explanation level	Feature selection	FeatureExplorer [42], FeatureInsight [43], INFUSE [44], TimeCluster [45–47]
		Performance analysis	CNNComparator [48], ComDia+ [49], CrossVis [50], DeepCompare [51], InstanceFlow [52], Squares [53], ConfusionWheel [54]
		Model and architecture understanding	ActiVis [1], CNNVis [5], LSTMVis [8], ReVACNN [9], ProtoSteer [10], REMAP [55], CNNSlicer [56], DGMTracker [57], CNNExplainer [58], CNNPruner [59], DeepEyes [60], DeepTracker [61], Deepvix [62], Manifold [63], RetainVis [64], SUMMIT [65], TopoAct [66]
VA adopting XAI methods		Global	RuleMatrix [67], iForest [68], ModelSpeX [69,70]
		Local	ExplainExplore [71], DeepVID [72–75], SENN [76–83]
		Both	Model Diagnostics [84], explAIner [85], MELODY [86], SUBPLEX [87,88], iNNvestigate [89,90], xDNN [91]

an interactive framework or system in interpreting ML models. Then, we divided implementation papers into subcategories and reviewed them to state the current situation and trends of the XAI in the field of visual analytics and present how VA can enhance interpreting NN models through XAI methods. After this selection process, 55 implementation papers were selected for this review, as seen in Table 3.

3.2. Paper classification

We classified the papers based on a newly defined XAI concept. We define XAI as a set of techniques that provide explanations of how decisions are made by black-box AI systems in an understandable, sense-making, and intuitive way for end-users. Unlike other definitions, our definition of XAI focuses on visual analytics perspective because we believe that the best way to present intuitive and sense-making explanations through meaningful representations for end-users is interactive visualization. With the awareness in mind, researchers have often utilized interactive visual analytics frameworks in ML applications to interpret feature selection [42–44], performance analysis [48,49,53], and model and architecture understanding [1,5,8,9,55]. However, most of these studies do not adopt XAI methods, which help to understand model behavior, inner workings, and understandable explanations of the prediction mechanism mathematically. But yet, these efforts lead us to the development of the field of XAI for black-box models from the visual analytics perspective. To ease the classification, we defined two new terms that used throughout the rest of the paper: visual interpretation (VI) and visual-based XAI (vXAI). ***Visual interpretation*** refers to usage of visualization techniques in an interactive framework to interpret NN models without using XAI methods for domain experts, data scientists and end-users. ***Visual-based XAI*** indicates the integration of visual interpretations and XAI methods in an interactive visual interface to provide a better interpretation of NN models.

The papers are initially classified into VI and vXAI. We then further reviewed all papers based on model usage and visual approach. In visual interpretation, model usage refers feature selection, performance analysis, and model understanding. We then summarized commonly used visual approaches in data representation, architecture understanding, and performance comparison. In vXAI, model usage covers rule, feature, propagation based and other XAI methods. Visual approaches used in data representation, global and local explanations are then concluded for vXAI paper. Fig. 4 presents the paper categorization scheme. With this categorization, we analyze, identify, and present the results of how visual approaches and model usage help to enhance the interpretability and transparency of NN models. The study is one of the first attempts to review the VA papers that combined with XAI methods and analyze how VA can be designed and utilized in explaining NN models through XAI. The study states current challenges and future directions in this area, as well.

4. Visual interpretation

This section focuses on techniques and visualization approaches to explain NN models without adopting XAI.

4.1. Model usage

Model usage covers common techniques used in explaining NNs via feature selection, performance analysis, and model and architecture understanding.

4.1.1. Feature selection

Feature selection is often used in ML to determine significant features, and remove irrelevant and redundant features from datasets in improving model performance and reduce noise [42]. Feature selection and evaluation techniques have been adopted in many visual analytics frameworks to identify informative and non-informative features by displaying feature contributions. VA supports interactions to filter, add and remove features to see how predictions are affected quickly, especially when the data are high dimensional [92]. Thus, researchers can interpret ML models by observing changes during the training and prediction phase through feature selection strategies.

One of the most common feature selection approaches is subset selection. Feature subset selection helps to remove irrelevant and redundant features from the feature space to improve the model performance and prediction accuracy. Moreover, feature subset selection enables the interpretation of the model and data by observing the contribution of different feature subset groups to the predictions. Feature subset selection through an interactive visual analytics framework helps to establish relationships between different feature and instance groups. For example, FeatureExplorer [42] presents a dynamic feature subset selection with integrated regression models for high dimensional datasets,

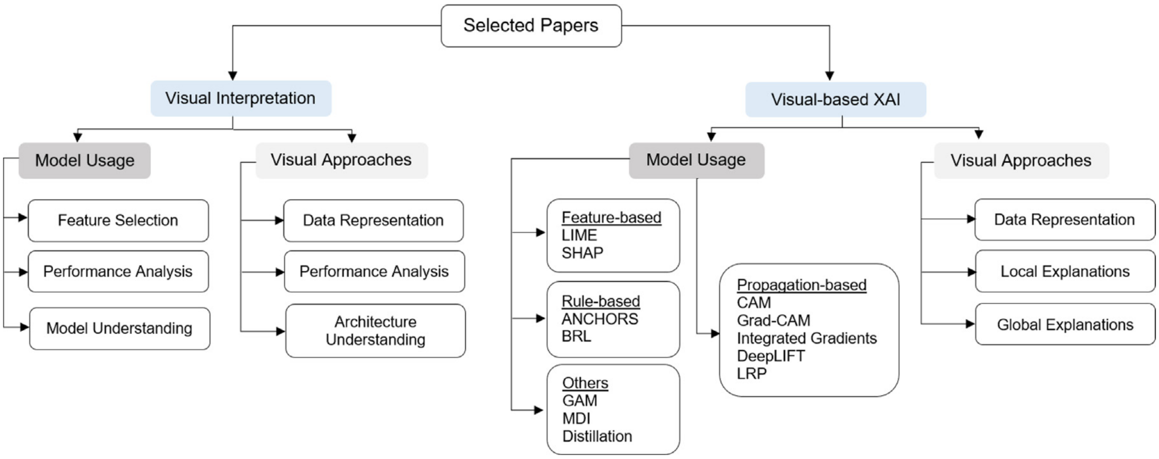
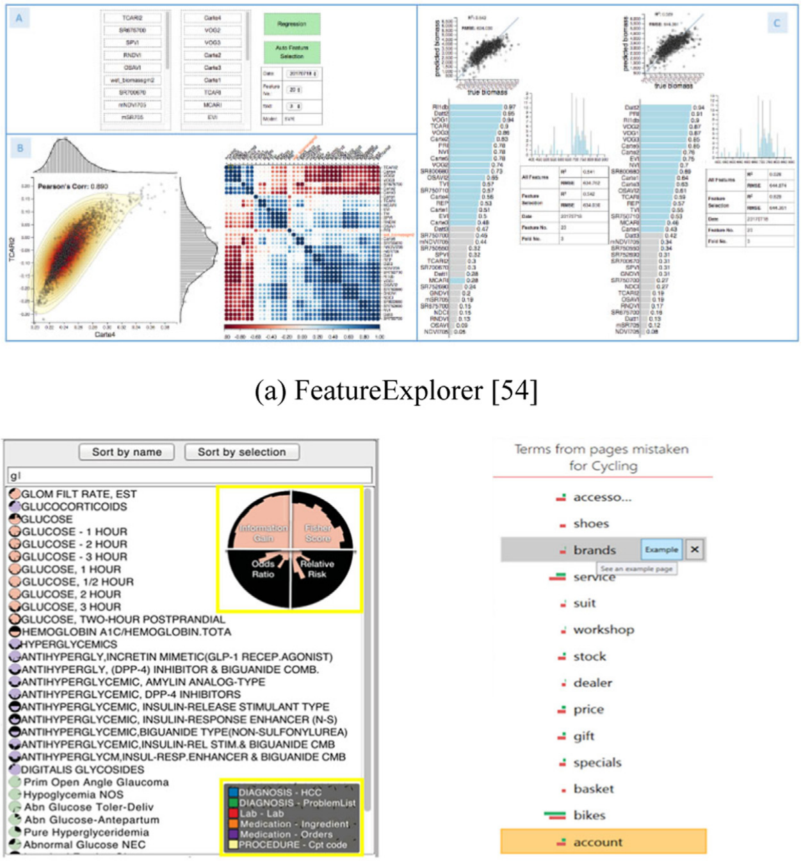


Fig. 4. The paper categorization scheme.



(a) FeatureExplorer [54]

(b) INFUSE [56]

(c) FeatureInsight [55]

Fig. 5. Visual analytics tools for feature selection strategies.

as shown in Fig. 5a.*A*. Linear relationships between features are shown in a correlation map with scatterplot enhanced by kernel density estimation visualizations, as seen in Fig. 5a.*B*. Feature importance are obtained using *Support Vector Regressor* and *Recursive Feature Elimination* method. The ranking results are presented via horizontal bar charts, as seen in Fig. 5a.*C*, to ease the feature subset selection process for users by comparing model performance.

Similarly, INFUSE [44] adopts various feature selection algorithms including *information gain, Fisher score, odds ratio* and *relative risks*, to select the most informative features by comparing the results of these algorithms through circular glyphs, as seen in Fig. 5b. It displays sorted lists of all features and quality scores. Circular glyphs are divided into four parts that represent feature selection algorithms, as highlighted in the top right of Fig. 5b. The color of circular glyphs indicates the subtype of features, as highlighted in the bottom right of Fig. 5b. INFUSE [44] supports feature reorder, filter, and select features to increase the model performance and interpretation through visual interactions. SmartStripes [47] presented subset selection in a matrix-based layout by ranking the features based on relevance. It also allows users to investigate the dependency between features and instance subsets. Hohman et al. [46] presented a rank-by-feature framework for feature subset selection by using ranking criteria such as *normality*, *Pearson's correlation coefficient*, and *entropy*. The framework shows feature ranks in a correlation map and table to ease the subset selection process for users.

Another feature selection strategy is feature extraction, which creates new features from the existing feature space to improve the model performance and perceive the data. FeatureInsight [43] examines the effect of a *set of errors* and its *visual summaries* to support feature extraction in the text classifiers. It uses ranked and annotated word lists to focus on important features and error sets of text documents. Fig. 5c shows the ranked lists of words to summarize and emphasize errors of keywords of bicycling web pages. TimeCluster [45] supports various dimensionality reduction (DR) techniques, such as principal component analysis (PCA) [93], Uniform Manifold Approximation and Projection (UMAP) [94]. The system permits DR comparison to enhance feature extraction for time series data. Combination of the visualization and feature extraction with dimensionality reduction techniques in TimeCluster [45] highlighted internal logic of ML models. It provides interactions such as zooming, filtering, and brushing to increase the efficiency of the system by allowing details-on-demand views for users.

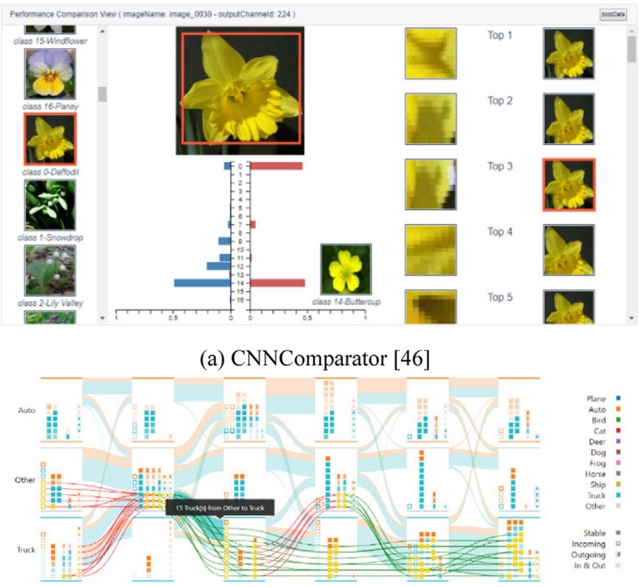
4.1.2. Performance analysis

Performance analysis is an essential process to select appropriate ML models. Summary statistics such as accuracy, precision, recall provide a good perspective in understanding model performance and prediction. However, the traditional statistics are not enough to explore the model performance when a complex algorithm is applied, particularly for multiclass classifiers. Therefore, interactive VA systems, such as seen in [50,52–54], have been developed to interpret prediction results and compare ML performance by creating customized dashboard visualizations.

Due to the inconsistent and messy nature of multiclass problems, many VA platforms adopted instance-level exploration to enlighten multiclass classifier performance and underlying decision-making mechanisms. For example, Squares [53] conducts performance analysis on multiclass classifiers and supports common performance metrics like accuracy, true and false prediction rates with instance-level information. Classes are color-coded as columns in a similar way to parallel coordinates. Squares on both sides of these columns highlight the instance level prediction results. Similarly, Alsallakh et al. [54] developed a confusion wheel using a circular chord diagram. A colored histogram that shows class probabilities enables users to identify misclassified instances with prediction probability scores.

Another performance analysis strategy is to compare the performance of multiple classifiers after training process. For example, ComDia+ [49] targets performance analysis of up to 10 ML models at both class and instance levels in finding the reasons for misclassification in image recognition. It helps to enhance the model performance and interpretation by comparing, diagnosing, and improving multiple models through matrix views and bar charts. CNNComparator [48] and DeepCompare [51] tools focus on comparing two different NN architectures by linking the model structures. These interactive visualization tools aim to understand the reasons for misclassified instances. CNNComparator [48] shows a comparison of a trained CNN model taken after 10 and 100 epochs by displaying classification results via bar charts and highlighting high impact feature, as seen in Fig. 6a.

To increase the interpretation of the black-box models, researchers recently focused on performance analysis during the training process considering the temporal evolution of the models. For example, InstanceFlow [52] has been developed to analyze the algorithms and their performance during the training process. It aims to analyze the model behavior over time by providing a fully temporal analysis on both class and instance level through a Sankey diagram, which visualizes the epochs and helps to follow instances, as seen in Fig. 6b.



(a) CNNComparator [46]

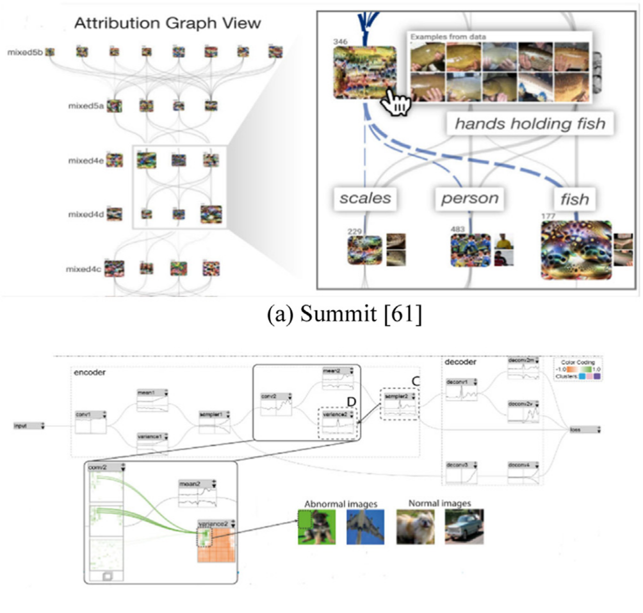
(b) InstanceFlow [57]

Fig. 6. Performance analysis and comparison on deep learning models through visual analytics.

4.1.3. Model and architecture understanding

Model and architecture understanding through visual analytics helps to gain more insights into the structure of NN models and their decision-making mechanisms. Many attempts have been done to *explore model architecture* [1,5,8,9,56,58,62,65], *refine* [10,55,59,60,64], and *diagnose* [57,61,63,66] NN models through interactive VA in making their inner working mechanisms transparent.

Understanding a model and its architecture through a visual interface allows both end-users and ML experts to gain insight on how a certain prediction has been made and how the architecture flows the data through layers [95]. It also helps data scientists on the model development process by displaying the layers and nodes to understand and explore the topology of a model [15]. For example, ActiVis [1] visualizes the structure of large-scale neural networks in a graph-based representation and shows their inner working mechanisms and neuron activations at both instance and subset levels through multiple integrated views. Tools like CNNVis [5] and ReVACNN [9] visualize CNN architectures and explore the activations in the layers and nodes. These tools illustrate the roles of the layers and nodes in the decision process. CNNExplainer [58] helps users to understand how CNN works by showing mathematical operations between layers and neurons. It visualizes neurons as heatmaps and integrates multiple level views to enhance the interpretation of CNN. Similarly, CNNSlicer [56] is developed to understand information distillation of CNNs by exploring various components of CNN, such as convolutional layers, pooling layers presented in heatmaps and matrix views. Unlike other studies, LSTMVis [8] and Deepvix [62] focus on interpreting and exploring the architecture of Long Short Term Memory (LSTM) models via visual analytics



(a) Summit [61]

(b) DGMTracker [44]

Fig. 7. VA tools for understanding architecture of NN models.

tools. LSTMVis [8] focuses on the hidden states over texts through parallel coordinates. Deepvix [62] visualizes nodes inside layers as heatmaps and connects these heatmaps by displaying weights on the links between layers. It provides detailed views of LSTM to understand its components, structure and learning process. Furthermore, it allows users to perform what-if analysis to enlighten decision-making mechanism of LSTMs.

Feature visualization is another important issue in interpreting the NN models. Feature visualization helps to understand what features are learned during the training process and how they impact the predictions. Summit [65] summarizes the features learned by a deep learning model to support the model understanding process of NNs. It explores important neurons in terms of activations and their relationships through an attribution graph, as seen in Fig. 7.a.

Manifold [63], DeepTracker [61], DGMTracker [57] and TopoAct [66] are other popular VA tools targeting diagnosing processes in model training by visualizing the training dynamics. Manifold [63] is a model-agnostic VA tool that diagnoses and interprets ML models by inspecting and explaining instances and then refining the model. It presents a scatterplot-based visual summary to display the outcomes of the models and feature attribution. Users can create new models to improve the performance by diagnosing and exploring these views. TopoAct [66] focuses on complex topological structures and by summarizing them to discover activations across multiple layers for deep learning classifiers. It provides a graph-based summary view and feature visualizations to enhance discovery and diagnose of image classifiers. DGMTracker [57] monitors and diagnoses the training process of deep generative models by visualizing activation changes over time to show how data neurons contribute to other neurons, as seen in Fig. 7.b. Similarly, DeepTracker [61] is proposed to visualize the training dynamics of CNN models from different levels of detail. They introduced a cube-style visualization to provide correlations among training data, iterations, and neuron weights.

Model refining and steering focus on the improvement of model performance through an interactive system. Many studies have been conducted to increase the performance of NN models via pruning, steering and architecture searching using VA techniques. For example, REMAP [55] explores NN architectures by adding, removing, or replacing hidden layers. It also permits quick hyperparameters searching to enhance performance of NN models. It allows interactive model generation by either searching the neighbors of a selected model or removing layers from the model one by one. Similarly, DeepEyes [60] explores the architectures of NN models by analyzing them during the training process. It removes the disadvantage of hyperparameter searching , which is time-consuming process through manual trial-and-error, by allowing model design during the training process. It presents layers in an activation heatmap and visualizes filter map as scatterplots. Another example is ProtoSteer [10] that provides interactive human-guided refinements for Recurrent Neural Networks (RNN) to improve the model performance via prototypes. It allows adding, editing, and removing prototypes through multiple connected views.

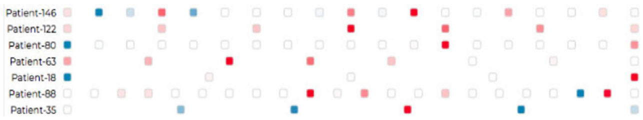
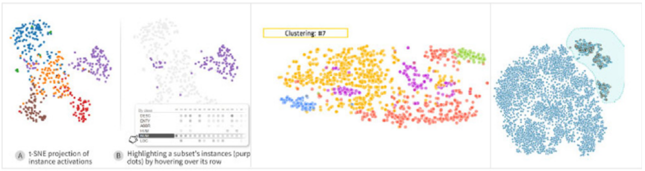


Fig. 8. RetainVis [64].


t-SNE projection of instance activations
Highlighting a subset's instances (purp dots) by hovering over its row
Clustering: 87

(a) ActiVis [1]

(b) ClusterVision [96]

(c) RetainVis [59]

Fig. 9. Data representation through scatterplots supported by dimensionality reduction techniques.

4.2. Visual approaches

Visual approaches adopted in visual analytics tools to represent data, architecture and assist in performance analysis are presented in the following subsections.

4.2.1. Data representation

Presenting data and classes in a dataset through visualizations enable users to understand data features and hidden patterns behind it. Many VA tools provides raw data access for performance analysis. For example, Squares [53] and CrossVis [50] display numeric data in a table form in interpreting the classification results. LSTMVis [8] and DeepCompare [51] visualize text samples in a list form to ease interpretation of ML models. REMAP [55], CNNExplainer [58] and TopoAct [66] show actual images to increase the understanding of the model decisions.

When data is high dimensional, displaying all samples in a VA system can be challenging since it could cause performance issues and visual clutters. Therefore, many VA tools use DR techniques to diminish the size of the data, then visualize them using scatterplots. For example, ClusterVision [96], ActiVis [1], DeepEyes [60], Summit [65], and RetainVis [64] visualize data instances via scatterplots enhanced by DR techniques. RetainVis [64] also visualizes individual instances through a customized row of rectangles view, as seen in Fig. 8. Users can select or group data samples through direct manipulations for further examination to understand data and features in detail. Fig. 9 shows some examples of data representations through scatterplot supported by DR techniques.

4.2.2. Architecture understanding

Visualizing the architectures of deep neural networks allows users to understand the structure of models and gain more insight on the working mechanisms of these black-box models considered as black-box. Recently, many VA tools [1,9,59] were developed to visualize NN structures. For example, ActiVis [1] visualizes large-scale network architectures and neuron activations in computational graphs by formulating the network as directed acyclic graph (DAG). It utilizes both instance-level and subset-level neuron activities by integrating coordinated views. Similarly, CNNVis [5] illustrates architectures as a directed acyclic graph to reveal the facets of each neuron and the interactions between them.

Another common approach to visualize NN architecture is to represent them as directed network with nodes and links. For example, Summit [65] proposed a customized view on attributes, where it displays network architecture using vertices and edges that representing highly activated neurons and their connections, respectively. ReVACNN [9] visualizes CNN using a node-link representation. It allows users to gain insights by observing the training dynamics of the model.

When the input data and the number of layers and neurons are high-dimensional, illustrating NNs as node-link representation may cause visual clutter. Therefore, many studies have adopted heatmaps for input and neuron illustrations and parallel coordinate plots (PCP) to connect layers. For example, Deepvix [62] presents neurons at each layer as a heatmap and connect them through links to interpret the working mechanism of LSTMs over time. Similarly, CNNExplainer [58] illustrates CNN architecture as a square that represents each neuron with a heatmap. It also supports detail-on-demand visualization to provide interpretations on different levels. LSTMVis [8] also displays the activations of the hidden states through a parallel coordinate plot. DeepTracker [61] adopts a neural network visualizer into their systems along with a heatmap to display activations in layers.

Fig. 10 shows some different ways in illustrating NN structure, active neurons, and hidden layers. Fig. 10(a–b) shows convolutional neural network architectures and hidden layers formulated as DAG. Fig. 10c is an example of visualization of NN as node-link. Fig. 10(d–f) display NN via heatmaps and parallel coordinate plots.

4.2.3. Performance analysis

Traditional summary tools of ML models, such as confusion matrix, accuracy, and prediction scores, are not enough to explore the model performance, particularly for multiclass problems. These traditional tools do not provide performance analysis on individual instances and subset levels. Therefore, VA scientists have been improving traditional summary tools and using customized interactive visualization, as seen in [48–50,52], to do performance analysis and comparison. Alsallakh et al. [54] proposed ConfusionWheel using a circular chord diagram, a colored histogram that shows class probabilities to identify misclassified instances with prediction probability scores. The confusion wheel provides feature analysis view to illustrate data features and help feature selection. Ren et al. [53] introduced Squares that mainly focuses on multiclass classifiers' performance analysis. Squares [53] supports common performance metrics like accuracy, true and false prediction rates with instance-level exploration through color-coded squares in a parallel coordinate plot. ComDia+ [49] proposed a view of customized overall performance comparison through connected circles to display accuracy scores at the class level, shown in Fig. 11. CNNComparator [48] provides a side-by-side performance comparison of two different CNNs through a horizontal bar chart for a selected instance. Although these studies developed a customized visualization for performance analysis in multiclass classifiers, several other VA tools such as CNNSlicer [56], CNNPruner [59], and REMAP [55], adopted some traditional visual summary tools such as confusion matrix, bar charts, and line charts, for performance analysis and comparison.

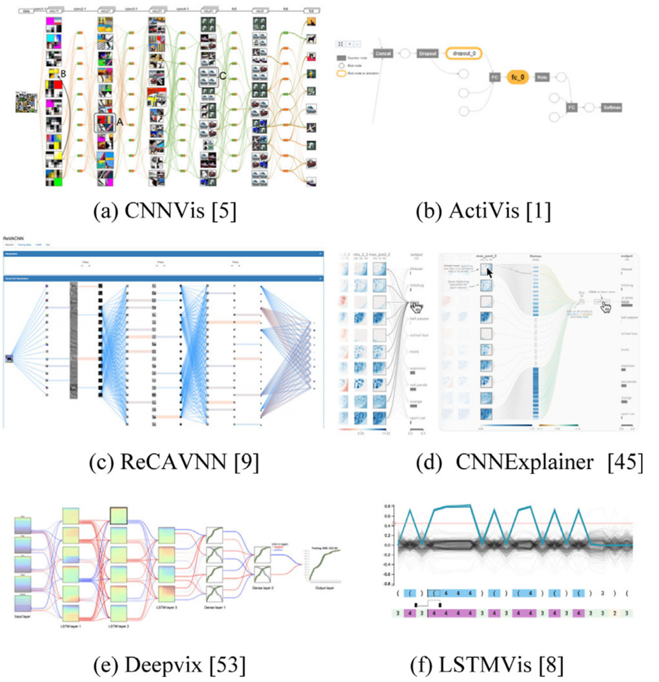

(a) CNNVis [5]
(b) ActiVis [1]
(c) ReCAVNN [9]
(d) CNNExplainer [45]
(e) Deepvix [53]
(f) LSTMVis [8]

Fig. 10. Different examples in illustrating NN structure, neurons and hidden layers.

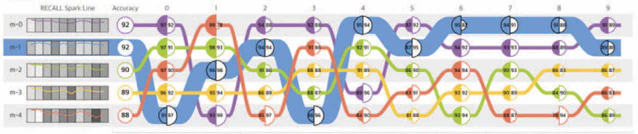
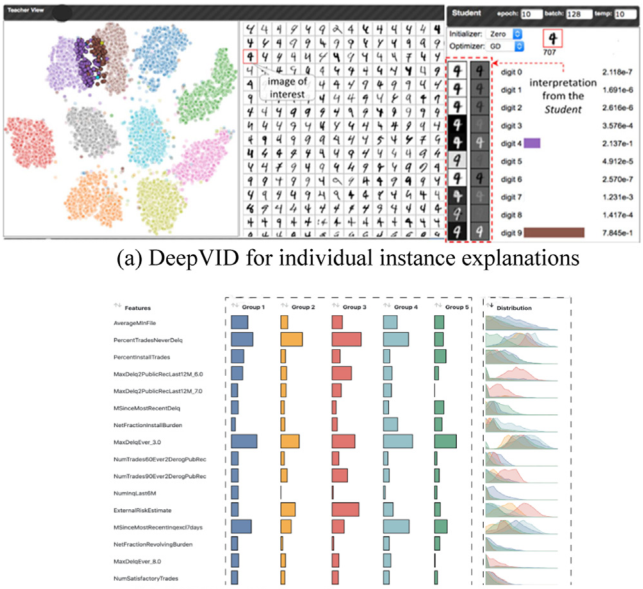


Fig. 11. A customized performance comparison view: ComDia+ [49].

Table 4
A summary of visual approaches adopted by visual interpretation papers.

	Visual approach	Papers
Data representation	Actual data	LSTMVis [8], REMAP [55], CNNExplainer [58], CrossVis [50], DeepCompare [51], Squares [53], TopoAct [66]
	Scatterplots	ActiVis [1], DeepEyes [60], RetainVis [64], Summit [65]
Architecture understanding	DAG	ActiVis [1], CNNVis [5]
	Node-Link	ReVACNN [9], Summit [65]
	Heatmap + PCP	LSTMVis [8], CNNExplainer [58], DeepTracker [61], Deepvix [62],
Performance analysis	Customized visualizations	ActiVis [1], CNNComparator [48], ComDia+ [49], Squares [53], ConfusionWheel [54],
	Traditional visualizations	ReVACNN [9], REMAP [55], CNNSlicer [56], CNNPruner [59], Deepvix [62]

Table 4 summarizes the visual approaches adopted by visual interpretation papers in data representation, architecture understanding and performance analysis.



Student epoch: 10 batch: 128 temp: 10

Initializer: Zero

Optimizer: GD

digit 0	2.118e-7
digit 1	1.691e-6
digit 2	2.616e-6
digit 3	3.576e-4
digit 4	2.137e-1
digit 5	4.912e-5
digit 6	2.570e-7
digit 7	1.231e-3
digit 8	1.417e-4
digit 9	7.845e-1

(a) DeepVID for individual instance explanations

(b) SUBPLEX for subgroup explanations

Fig. 12. VA tools for both instance and subgroup level explanations by the LIME method.

5. Visual-based XAI (vXAI)

vXAI is an emerging area of research in the field of VA. Comparing with visual interpretation based studies, vXAI papers are very limited This section summarizes how visual approaches and model usage are used in visual-based XAI papers to make NN models more transparent by adopting XAI techniques. The subsections are created based on the paper categorization scheme as shown in Fig. 4.

5.1. Model usage

Under model usage, we grouped XAI methods in the subsections based on feature-based, rule-based, propagation-based, and other methods.

5.1.1. Feature-based methods

Feature-based XAI methods produce the importance values of features to show their contributions to predictions in explaining black-box models. LIME method [28] is the most popular model-agnostic XAI method often adopted in vXAI. LIME [28] generates training samples by perturbing neighbors around the user-selected instance, then uses a surrogate model to explain the instance locally. Surrogate models are simple interpretable models, such as linear models, to approximate black-box models. These surrogate models provide explanations for black-box models by obtaining feature importance via coefficients of linear models.

LIME method is often used in various data domains such as tabular [75,79,82], image [77], and text [83] due to its easy-to-use feature. Several vXAI works, such as ExplainExplore [71], SUBPLEX [87], and DeepVID [72], modified and integrated the LIME method into their interactive VA platforms to explain NN models locally. ExplainExplore [71] modified the LIME method to increase the explainability by using not only linear models but also shallow tree-based models as surrogate models. ExplainExplore [71] provides explanations locally for a selected instance by displaying generated samples interactively and allowing direct manipulations. Another extension for the LIME method has been done by DeepVID [72] by improving the training sample generation process through deep generative networks. It allows users to select a sampling region around one or more selected instances and provides local explanations. Fig. 12.a displays DeepVID that explains the NN model at instance level using the LIME method. While the LIME method usually has been used to explain a selected instance, SUBPLEX [87] presents the average feature contributions to provide explanations for a selected subgroup by modifying the LIME method, as seen in Fig. 12.b.

SHAP [35] is the other popular feature-based XAI method used for feature explanation. Li et al. [73] adopts SHAP in their VA system to interpret feature selection on clinical data. They present local explanations of both individual and clustered instances interactively. The tool also allows multiple model comparison by displaying the similarities of ML models based on local feature importance as seen in Fig. 13, by using a scatterplot obtained by t-Distributed Stochastic Neighbor Embedding (t-SNE) [97], which helps understand the rationale of different models.

5.1.2. Rule-based methods

Rule-based explanations are another often seen XAI method used in vXAI study. The method provides logical statements, i.e., IF…THEN…ELSE, to present rule lists that explain the predictions. RuleMatrix [67] displays a graphical representation of rule-lists obtained by Bayesian Rule Lists [32], which produces a posterior distribution for possible IF-THEN rules using the Bayesian framework. RuleMatrix visualized IF-THEN rules in the form of a matrix as seen in Fig. 14, for a group that satisfies the related rule(s). xDNN [91] provides prototype-based IF-THEN rules to explain NNs by displaying similarities between validation and selected image. Model Diagnostics [84] also adopts a rule extraction method to present local and group level explanations for binary classifiers. Similarly, ModelSpeX [69] employs a rule extraction technique to provide rule-based explanations for decisions. However, Model Diagnostics [84] and ModelSpeX [69] did not visualize IF-THEN rules but focused on explanations of logic representations.

5.1.3. Propagation-based methods

XAI methods, such as LRP [30], CAM [41], Grad-CAM [29], back-propagate predictions to obtain feature relevance in explaining ML models at both global and local levels. Lauritsen et al. [88] utilized LRP to reveal feature importance in explaining predictions at both instance level and population-based level for acute critical illness records. Saliency maps are often used to highlight the contributions of features for each sample through propagation. Li et al. [81] used saliency maps to explain contributions and negations of words to the predictions obtained by LSTMs. CAM and Grad-CAM methods are proposed to improve the saliency. Kim et al. [78] utilized CAM method to obtain salience values for text classification. Saliency values are visualized using gray-scale saliency maps to explain CNNs.

The majority of VA research only integrated a specific type of XAI method into their systems. Since there are many XAI techniques, it is difficult to implement each one of them and compare their explainability individually. To overcome this challenge, iNNvestigate [89], a python package, has been developed to compare not only XAI methods but also NN models. It compares how different NN models would behave on the same instance through visualizations of feature contribution by using a user selected XAI method. The package supports a variety of propagation based XAI methods, such as Saliency Maps [39], Integrated Gradient [38], and LRP [30]. Similarly, explAIner [85] and MELODY [86] support various propagation-based and feature-based XAI methods, such as LIME [28], LRP [30], SHAP [35], Saliency Maps [39], DeepLIFT [36], to provide explanations through vXAI. These tools provide both local and global explanations for individual and subgroup instances.

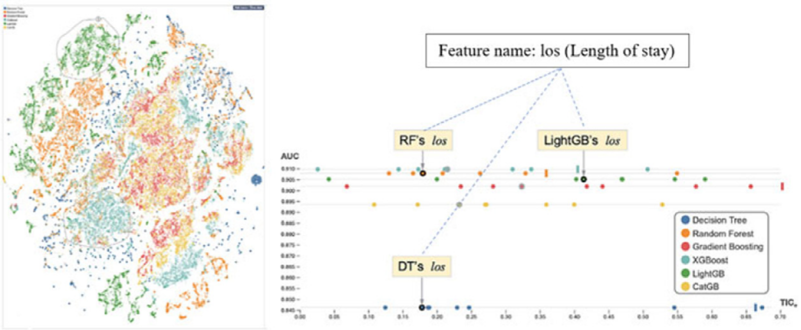


Fig. 13. Model comparison via each feature's consistency values [73].

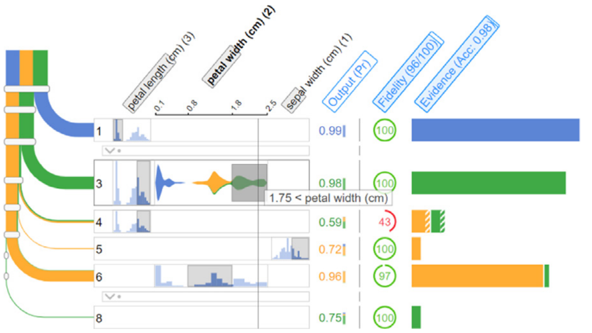


Fig. 14. RuleMatrix: If-Then rules visualization in a matrix form [67].

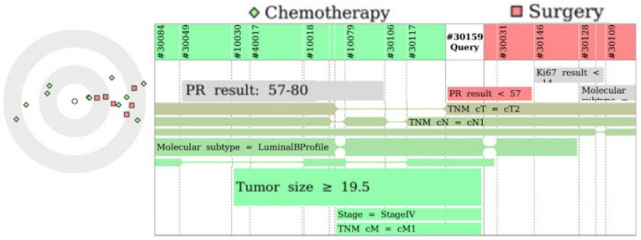


Fig. 15. Case-based reasoning adopted by VA [80].

5.1.4. Other methods

Case-Based Reasoning (CBR), another type of XAI method, helps to classify and explain a decision based on similarities of previous cases. It is often used in medicine domains [70,80]. Studies presented by Lamy and Tsopra [70] and Lamy et al. [80] have used a case-based reasoning approach to provide explanations for clinical datasets through rainbow boxes, as seen in Fig. 15. Rainbow boxes aim to explain the rationale a decision via weights and case similarities. The scatterplot on the left shows similar cases in color-coded classes and their distance to the query. The height of the rainbow boxes shows the mutual information between the query and similar cases. The color of rainbow boxes is determined in proportion by weighting the cases based on their similarity to the query. iForest [68] uses the MDI method [37], which is a model-specific method for Random Forest, to present feature-based explanations. iForest [68] reveals the working mechanism of Random Forest by summarizing decision paths based on feature ranges. It supports case-based reasoning by adopting similarity measures of both data similarity and decision path.

Table 5 presents a summary of the reviewed vXAI papers based on model usage. We summarized vXAI papers based on data domain (i.e., tabular, text, and image), explanation and implementation level, and dependency of adopted XAI methods. Most of the vXAI tools adopt model agnostic XAI methods. Among these methods, LIME is the most utilized method to provide local explanations. The most common data domain used in the vXAI tools is tabular data.

5.2. Visual approaches

This section summarizes visual approaches used in data representation, local and global explanations adopted by vXAI research.

5.2.1. Data representation

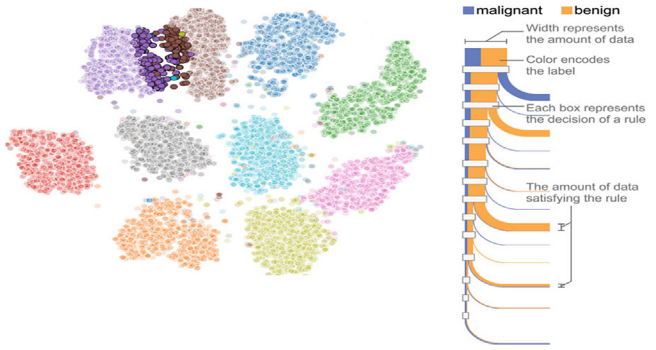
Through visual representations, data can be more accessible to help users engage in exploration and analysis [98]. In a VA framework, datasets and/or their classes are often presented visually to users. Instance exploration would enhance prediction explanation. Therefore, many vXAI papers adopted actual data representation in their system. For example, RuleMatrix [67] displays actual data in a table form. Thus, users can investigate data by filtering in the data table and focus on specific instance or subgroups. Similarly, iForest [68] also adopted table form to represent raw data. iForest [68] allows browsing and selecting an instance or subpopulation from the table for exploration through connected views. For image datasets, actual images are often presented to explain the feature importance, as seen in explAIner [85], Meske and Bunde [77], and DeepVID [72]. Moreover, Kim et al. [78] and Li et al. [81] presented text data by displaying each instance in a text box.

For high dimensional datasets, data representation via scatterplots using DR techniques have been adopted by many vXAI tools as well to reduce visual clutter. For example, Botari et al. [75], Baptista et al. [79], Lauritsen et al. [88] presented tabular data enhanced by DR techniques in a scatterplot. iForest [68] and DeepVID [72] represent data supported by DR techniques through color-coded classes in scatterplot along with the raw data to increase data exploration. In Fig. 16a, DeepVID [72] displays color-coded data instances in a scatterplot obtained by t-SNE [82]. In this way, users can select and analyze a single instance and/or subgroups to understand the predictions. For example, SUBPLEX [87] uses scatterplots with a DR technique — Local Affine Multidimensional Projection (LAMP) [99]- to show subgroups in the data. LAMP maps the attributions in a 2D plane by presenting the subgroups produced by the LIME method.

Another data representation approach is to use Sankey diagrams. Sankey diagrams show how data and instances flow through overall data feature distribution. For example, RuleMatrix [67] uses a color-coded Sankey diagram to display data flow through rule lists for tabular data, as seen in Fig. 16b. With the Sankey diagram, users can follow the amount of data satisfying a rule by observing the width of the Sankey diagram flow. MELODY [86] uses a Sankey diagram to visualize the data flow from each class to different instances clusters for various domains, i.e., tabular, text and image. Examples of data representations in vXAI through scatterplots and Sankey diagrams are shown in Fig. 16.

Table 5
A summary for vXAI papers based on model usage.

XAI method		Papers	Explanation level	Implementation	Dependency	Domain
Feature-based	*LIME*	ExplainExplore [71]	Local	Post-Hoc	MA	TB
		SUBPLEX [87]	Both	Post-Hoc	MA	TB
		DeepVID [72]	Local	Post-Hoc	MA	IM
		Botari et al. [75]	Local	Post-Hoc	MA	TB
		Meske and Bunde [77]	Local	Post-Hoc	MA	IM
		Baptista et al. [79]	Local	Post-Hoc	MA	TB
		Islam et al. [82]	Local	Post-Hoc	MA	TB
		So [83]	Local	Post-Hoc	MA	TXT
	SHAP	Li et al. [73]	Local	Post-Hoc	MA	TB
Rule-based	*BRL*	RuleMatrix [67]	Global	Post-Hoc	MA	TB
	Rule extraction	Model diagnostics [84]	Both	Post-Hoc	MA	TB
		ModelSpeX [69]	Global	Post-Hoc	MA	TB
		xDNN [91]	Both	Post-Hoc	MA	IM
Propagation-based	*LRP*	Lauritsen et al. [88]	Both	Post-Hoc	MA	TB
	CAM	Kim et al. [78]	Local	Post-Hoc	MA	TXT
	Saliency maps	Li et al. [81]	Local	Post-Hoc	MA	TXT
Others	*CBR*	iForest [68]	Local	Intrinsic	MS	TB
		Lamy and Tsopra [70]	Global	Post-Hoc	MA	TB
		Lamy et al. [80]	Local	Post-Hoc	MA	TB
	Diverse	explAIner [85]	Both	Both	Both	IM
		MELODY [86]	Both	Both	Both	ANY
		iNNvestigate [89]	Both	Both	Both	ANY



(a) DeepVID [72] (b) RuleMatrix [73]

Fig. 16. Data representations through scatterplots (a) and Sankey diagram (b).

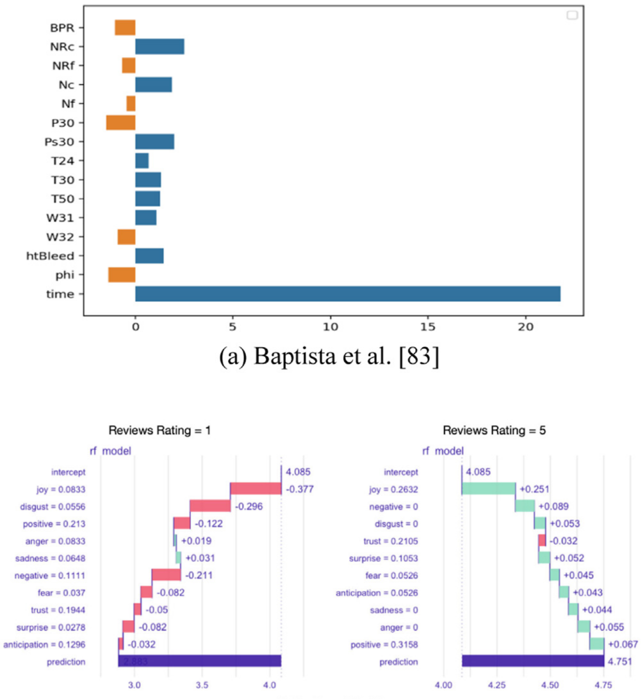
5.2.2. *Local explanations*

Local explanations overcome challenges of explainability related to complex models by providing local explainability through selected instances or surrogate models to understand the behavior of the whole model [24]. Many VA tools integrate local explanations into their systems, as seen in Model Diagnostics [84], explAIner [85], iForest [68], and Li et al. [73]. Displaying local feature contribution for selected instances helps users to understand which features contribute the most to the corresponding prediction. It explains why a certain prediction is made by a classifier and helps users to understand model behavior for similar instances.

The most popular visualization technique to present local explanations is bar charts. For example, ExplainExplore [71], SUBPLEX [87], Model Diagnostics [84], RuleMatrix [67] and iForest [68] use horizontal and vertical bar charts to display local feature contributions to the related predictions. Bar charts show feature value distribution and allow a quick comparison among their values so users can easily notice the most contributing feature groups in a prediction. Another popular visual approach for local explanations is breakdown plots (BDP) that show positive and negative contributions of each feature for a selected instance's prediction. Some studies [69,76,79,83] use a color-coded breakdown plot to present positive and negative contributions of each feature for a certain prediction. These contributions are often obtained via feature-based XAI methods such as LIME method [28]. Bar charts and breakdown charts are generally preferred in local explanations because they are straightforward to interpret the results of feature selection. Additionally, they can be applied to any data domain such as tabular, text, and image. Fig. 17 shows breakdown plots for tabular and text datasets utilized in studies [79] and [83], respectively.

Heatmaps are another popular visual approach used in local explanation, especially for image datasets. Local explanations are often highlighted in the pixels and/or regions that contribute to the prediction over the actual image with a heatmap.

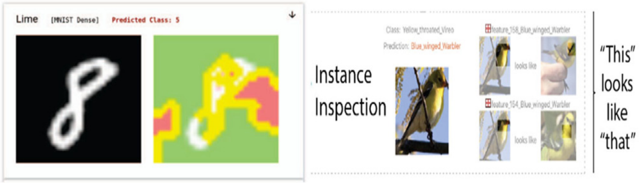
These contributions are often obtained through XAI methods. For example, DeepVID [72] uses LIME method to obtain feature contributions and reflects these values over the image by highlighting pixels based on these scores. explAIner [85] and MELODY [86] support various XAI methods in their systems so that they can use different XAI methods to display heatmaps on the actual images. Fig. 18a shows the pixel contributions to predict digit 8 through local explanations using LIME method in explAIner [85]. In Fig. 18b, MELODY [86] shows instance inspection by comparing and highlighting similar patches between a selected image and other instance. Besides image data, heatmaps have been also used in text data to display words that contribute to the predictions. For example, Li et al. [81] and Kim et al. [78] use Saliency map and CAM methods respectively to reflect feature/word contributions to text classification via heatmaps. Heatmaps allow users to gain insights on how a model recognizes certain images or words using input features.



(a) Baptista et al. [83]

(b) So [91]

Fig. 17. Local feature contributions via breakdown plots.



(a) Explainer [68]

(b) MELODY [70]

Fig. 18. Common visual approaches for local explanations using heatmaps.

5.2.3. Global explanations

Global explanations focus on the explanation of the entire model, and its working and decision-making mechanisms. Providing global explanations of NN models is challenging due to the black-box structure and complex computational process. Therefore, there are a few studies provide explanations globally. ExplainExplore [71] explains feature contributions globally using PCP, where the selected instance to be explained is highlighted, to identify model behavior on similar instances. Histogram is a popular approach to present global feature contributions in understanding the model behavior for group patterns. For example, SUBPLEX [87], explAIner [85], Lauritsen et al. [88], and MELODY [86] use histograms to show the distributions of feature contributions.

Histograms and PCPs can be insufficient to explain the overall model behavior by only displaying feature contributions. To overcome this issue, researchers presented matrix-based visualizations including more details regarding the model behaviors. For example, RuleMatrix [67] visualizes the content of an ordered rule list in a matrix-based design with sorted feature contributions. The design allows users to follow the amount of data satisfying a certain rule with feature contribution. Model Diagnostics [84] also proposed a matrix-based design to explain model decisions in global level. They represented each row as a data group explained by a set of features. In each column, they presented an ordered feature set that contributes to the prediction along with the statistics to provide insights about the accuracy of the decision. With Model Diagnostics, users can drill down from data groups to individual items to inspect explanations from global level to local level. iForest [68] explains random forest globally through decision path flows that reveal decision paths at the individual layer level in detail, as seen in Fig. 19. The decision path view in Fig. 19 enables users to understand feature importance by observing the order of feature appearance in different user-selected decision paths. The width of the curves between features indicates the number of decision paths.

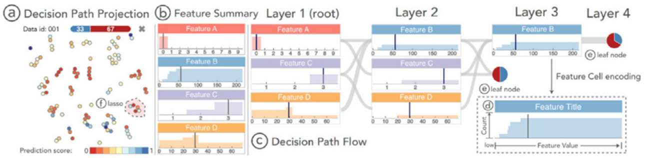


Fig. 19. Decision path view of iForest [68].

Table 6 summarizes the visual approaches used in vXAI papers. According to Table 6, the most popular visual approaches to represent data are displaying actual data and scatterplots. Most of the vXAI tools support local explanations in interpreting the model behavior through bar charts [71,85–87], breakdown plots [75,79,83] and heatmaps [72,81,85]. Global explanations are provided using parallel coordinate plots [71], matrices [67,84,86], and histograms [87].

6. Discussion, opportunities and future work

In the field of VA, we found there are very few research studies focusing on adopting XAI methods to explain ML. Therefore, we developed several research questions to address research needs and directions in this area. This section presents the current trends, research challenges, and opportunities for future work in vXAI, through some predetermined research questions by this survey.

1. How can VA systems be utilized to support the interpretation of NNs through XAI techniques?

Scalability in data representation: ML models learn from data, explore patterns, and make predictions by using the history data. The quality of data, distribution of classes and relevancy affect the performance and understanding of ML models. On the other hand, data visualization helps to analyze and understand the data, and see visible and hidden patterns. Hence, representing data in a VA system enhances the interpretation of NNs and provide more sense-making for predictions to the users. The most straightforward way is to represent actual data in tables [67,68], lists [86], or as images [72,85]. Users can select an instance or subgroups in the raw data to be explained by XAI and monitor how NN make predictions through an interactive VA. However, the main challenge is to have visually scalable data representations. Since most of the real-world applications have high-dimensional datasets, visualizing raw data in a VA system may cause visual clutter and scalability issues.

Therefore, many studies [1,60,64,72,73,87] present data and classes as color-coded scatterplot enhanced by dimensionality reduction techniques to provide scalable data visualization. However, an increase in the dimension of data, classes, and feature space will still cause overlapping data points and scalability issues because the system will not visualize and deal with them. An ideal visual analytics framework should illustrate data in a scalable manner along with the classes and allow more interactions such as searching, filtering, adding, or removing instances to

Table 6
The summary of vXAI papers based on visual approaches.

Study	Data representation			Local explanations				Global explanations			
	Actual data	SP	SD	BC	BDP	Heatmaps	O	PCP	H	Matrix	O
ExplainExplore [71]				✔				✔			
SUBPLEX [87]		✔		✔					✔		
MELODY [86]	✔		✔	✔			✔		✔	✔	
explAIner [85]	✔			✔		✔			✔		
RuleMatrix [67]	✔		✔	✔						✔	
DeepVID [72]	✔	✔				✔					
Krause et al. [84]				✔						✔	
iForest [68]	✔	✔		✔			✔				✔
Li et al. [73]		✔		✔							
Botari et al. [75]		✔			✔						
Baptista et al. [79]		✔			✔						
So [83]					✔						
Lamy et al. [80]		✔					✔				
Cho et al. [90]							✔			✔	
Lauritsen et al. [88]		✔					✔		✔		
J. Li et al. [81]		✔				✔					
Kim et al. [78]	✔					✔					

SP: Scatterplot, SD: Sankey Diagram, BC: Bar Chart, BDP: Breakdown Plot, H: Histogram, O: Other, PCP: Parallel Coordinate Plot.

support the interpretation and understanding of NNs through instance- and subset-level explanations. Additionally, clustering and bundling techniques can be also utilized to reduce visual clutter and scalability issues in data representation via VA. Visualizing data in interactive VA tools still has opportunities to be improved such as direct access to data, reducing scalability and visual clutter, increasing interactions, and data manipulation in future works. VA scientists can create new visual representations to deal with the large-scale data representations and components of NNs to support domain experts and ML scientists.

Feature understanding: Feature selection and extraction are significant factors that affect the performance and decisions of NN models. Determining important features and removing irrelevant and redundant ones from datasets improve model performance. The current studies such as [42–44] support various feature selection and extraction methods integrated into interactive VA tools to improve the model performance. Most of the XAI methods explain NNs based on feature weights and contributions to the predictions. The current studies visualize feature weights over the layers of NNs [1,8,9,88] and highlight feature relevance for the final decision [72,85,86]. They also allow limited interactions and manipulations on feature space to understand the model behavior and explanations on predictions. For example, Model Diagnostics [84] removes features one by one until there is any change in the prediction results to examine the effect of the features on the predictions. To enhance interpretation of NNs, in future work, VA systems should support automated feature selection/extraction preserving class balance along with additional and broader interactions to improve model accuracy and feature-based explainability. Including the user into the VA loop interactively will enhance the understanding of the effects of features on the predictions. Users can observe how feature selection/extraction can make changes in classification results/weights/activations on neurons with increased and broader interactions. Therefore, users can gain more insights how feature space affect the model decisions and performance, and understand how the model will behave for future predictions with selected feature space.

Performance analysis: Data and feature quality, model hyperparameters such as activation functions, learning rate, batch size, optimizers, and the number of nodes at each layer affect the performance of NN models. Conducting performance analysis on NN architectures increase the interpretation of NNs by analyzing and exploring misclassified instances through an interactive VA tool. Traditional performance summary tools such as confusion matrices can be insufficient explaining misclassified instances in multiclass classifiers. Therefore, many studies [1,48,49,53] have developed interactive customized performance analysis tools to reveal the reasoning behind the model predictions and interpreting NNs. Since these tools are developed either for a specific model or data type, there is no generalization and consensus on performance analysis for multiclass classifiers. Future works have opportunities to develop generalizable performance analysis tools for multiclass classifiers and interactive hyperparameter search to compare different NN configurations. ML scientists can support VA scientists by providing scalable computational methods in visual based neural architecture search. Therefore, domain experts can understand, compare, and select the best NN configurations by examining them visually for their domain data. This also reduces the human effort in the trial-and-error process of selecting the best configurations of NNs for domain experts and ML scientists.

Architecture understanding: Researchers have focused on visualizing architecture of NN models using network-based approaches. For example, ActiVis [1] and CNNVis [5] formulated NN as directed acyclic graphs to interpret deep learning models and results. CNNExplainer [58] uses heatmaps to represent nodes and shows internal operations to make a prediction for a selected instance. VA research has been well established to interpret NN by supporting the understanding of the internal working mechanism and diagnosing the model errors through interactive interfaces. However, there is still a lack of visualizing real-time online training process that helps interactive steering and model development and refinement process with user interactions. A real-time training process system should allow saving and reviewing interactions done by users during the training. It will enhance hyperparameter exploration and searching to build better models. However, developing an interactive real-time training process system has limitations such as time, computational resources, and scalability. Collaboration between VA and ML/AI communities would help to resolve this challenge.

Explainable AI and ML is a new research field and still evolving, so there is no standardized way of explaining each ML model and no consensus on the applicability of XAI in visual analytics for specific types of data. Researchers should focus on integrating XAI methods in training process by visualizing explanations over the architecture in real-time to make black-box models more transparent in future works. Thus, it will be possible to explain globally how NN models made their decision as a whole. VA scientists and ML scientists can work collaboratively to provide scalable, optimized, and user-friendly visual analytics tools both visually and computationally in explaining NNs using XAI.

2. What visualization techniques have been used to support the explainability of XAI techniques?

Local explanations: Local explanations focus on explaining model decisions based on a selected instance or subgroups. Most of the vXAI papers provided local explanations by displaying values of feature importance that contribute to the predictions. The most common way for local explanation is to use surrogate models [28], which helps to simplify the complex models by using interpretable models such as linear models and tree-based models.

They are used to classify samples that are generated by perturbing neighbors around a selected instance to explain the behavior of complex NN models. Surrogate models provide feature-wise contributions through either coefficient of linear models or local increments for tree-based models. These contributions are presented using bar charts (*for example, SUBPLEX [87], iForest [68]*), breakdown plots (*for example, Botari et al. [75], Baptista et al. [79]*), heatmaps (*for example, DeepVID [72], iNNvestigate [89]*), partial dependency plots (*for example, iForest [68]*) and highlighted data representations (*for example, explAIner [85], DeepVID [72], iNNvestigate [89]*).

The most common issue in vXAI is visual scalability. When the feature space is high dimensional, most of the VA have performance issues since they could not visualize all features that help to explain the predictions locally. Due to the simplicity of application, many vXAI works adopt local explainers focused on instance explanations rather than the entire model behavior. This leads us to another challenge in the VA systems: the bias in representative examples to be explained. Because local explanations are conducted around an instance selected by the user and its perturbed neighbors, this selection may cause bias in explaining similar instances. Future work should focus on presenting better and generalized ways to represent high impact data and features to resolve the scalability and bias issues. Additionally, developing common approaches for different domains to present explainability for XAI methods in supporting interpretation of NNs is important.

Global explanations: Global explanations reveal the reasoning behind the model predictions and its working mechanism. When the local explanations are insufficient to understand the model behavior, providing global explanations is essential. Parallel coordinate plots (*for example ExplainExplore [71]*), histograms (*for example SUBPLEX [87]*), and matrices (*for example Krause et al. [84] and MELODY [86]*) are commonly seen for the global explanations. Parallel coordinate plots help to see overall feature distribution so that users can see whether there are patterns for predictions. Similarly, histograms show the distribution and frequency of all features and their contributions to the overall predictions. vXAI is an evolving research field and still has opportunities to explain the working mechanisms of NNs during the training by displaying feature contributions obtained by XAI over the architectures.

Local explanations help to understand how a prediction has been made for an individual instance, and global explanations provide insight on general behaviors of a model for future predictions. Although there are differences in the applications of local and global explainers, the current stage of vXAI tools tends to combine explanations of NNs from both global and local perspectives.

3. Is there a common visual approach to support the illustration of XAI methods for different types of data/models?

vXAI is a new and evolving research field that aims to explain NN models and their decisions visually using XAI methods. Due to its novelty, there are no common visual approach to support the illustration of XAI methods for different types of data/models. There is also no standardized way to present local and global explanations.

Nevertheless, we discovered some similarities in visual approach among vXAI papers. For example, data are often represented through scatterplots or Sankey diagrams; however, there is no consensus on visualizations and general approach in representing different types of data in different domains. Actual data are often presented as tables, images or lists forms. While local explanations are often visualized using bar charts, global explanations are presented with parallel coordinate plots. Researchers usually tend to develop custom graphs according to their data domain and application area. Moreover, the same XAI method could adopt different visual approaches in model and instance explanations. For example, while ExplainExplore [71] presents the sampling region of LIME method via HyperSlice plots, DeepVID [72] displays this sampling region in a parallel coordinate. Similarly, explanations obtained by a rule-based explanation method are displayed as a matrix in RuleMatrix [67]; however, ModelSpeX [69] presented these explanations as logical statements.

Another difference seen among vXAI papers is the choice of NN models. Most of the papers focus on explaining CNN and RNN models among NNs. The state-of-the-art performance of these models in image, text, and time-series data in sensitive domains captivates researchers. Therefore, they focus on explaining these models and their decisions to increase trust and transparency. However, there are many other NN models including deep generative models, such as autoencoder and generative adversarial networks, that need to be explained. To present a generic visual platform in explaining NN models irrespective of their features and categories, more research should be conducted.

7. Conclusion

To gain the trustworthy on the decisions of the black-box models, XAI research has been growing rapidly. Since then, many XAI methods have been proposed to provide understandable results of AI to humans. This survey summarized the current state, challenges, and future directions of developing better visual analytics for XAI methods in interpreting neural networks. We have reviewed the interpretability of VA with and without involving XAI methods in both model usage and visual approach.

The main goal of the survey was to address how visual analytics can support better interpretations of neural networks through XAI methods. Therefore, we stated differences among the application of XAI methods in interpreting NNs through VA. We also identified the common visual approaches, such as bar charts, parallel coordinate plots, and heatmaps, to support global and local explanations of XAI methods. Since the vXAI field is still evolving, there are challenges related to scalability, performance analysis, the bias in representative examples, and consensus on common visual approach of the XAI methods. These issues meanwhile highlight future research directions of the fields.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kahng M, Andrews PY, Kalro A, Chau DH. ActiVis: Visual exploration of industry-scale deep neural network models. IEEE Trans Vis Comput Graph 2018;24:88–97. http://dx.doi.org/10.1109/TVCG.2017.2744718.

[2] Chatzimparmpas A, Martins RM, Jusufi I, Kucher K, Rossi F, Kerren A. The state of the art in enhancing trust in machine learning models with the use of visualizations. Comput Graph Forum 2020;39(3):713–56. http://dx.doi.org/10.1111/cgf.14034.

[3] Azodi CB, Tang J, Shiu S. Opening the black box: interpretable machine learning for geneticists. Trends Genet 2020;36(6):442–55. http://dx.doi.org/10.1016/j.tig.2020.03.005.

[4] Daglarli E. Explainable artificial intelligence (XAI) approaches and deep meta-learning models. In: Aceves-Fernandez Marco Antonio, editor. Advances and applications in deep learning. London, UK: IntechOpen; 2020, [Chapter 5]. http://dx.doi.org/10.5772/intechopen.92172.

[5] Liu M, Shi J, Li Z, Li C, Zhu J, Liu S. Towards better analysis of deep convolutional neural networks. IEEE Trans Vis Comput Graph 2017;23(1):91–100. http://dx.doi.org/10.1109/TVCG.2016.2598831.

[6] Gunning D, Aha D. DARPA's explainable artificial intelligence (XAI) program. AI Mag 2019;40(2):44–58. http://dx.doi.org/10.1609/aimag.v40i2.2850.

[7] Miller T. Explanation in artificial intelligence: Insights from the social sciences. Artif Intell 2019;267:1–38. http://dx.doi.org/10.1016/J.ARTINT.2018.07.007.

[8] Strobelt H, Gehrmann S, Pfister H, Rush AM. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. IEEE Trans Vis Comput Graph 2018;24(1):667–76. http://dx.doi.org/10.1109/TVCG.2017.2744158.

[9] Chung S, Suh S, Park C, Kang K, Choo J, Kwon BC. ReVACNN: Real-Time visual analytics for convolutional neural network. In: ACM SIGKDD workshop on interactive data exploration and analytics. 2016. p. 30–6.

[10] Ming Y, Xu P, Cheng F, Qu H, Ren L. ProtoSteer: Steering deep sequence model with prototypes. IEEE Trans Vis Comput Graph 2020;26(1):238–48. http://dx.doi.org/10.1109/TVCG.2019.2934267.

[11] Xu F, Uszkoreit H, Du Y, Fan W, Zhao D, Zhu J. Explainable AI: A brief survey on history, research areas, approaches and challenges. In: J. Tang, Kan MY, Zhao D, Li S, H. Zan, editors. Natural language processing and chinese computing. 2019, http://dx.doi.org/10.1007/978-3-030-32236-6_51.

[12] Emmert-Streib F, Yli-Harja O, Dehmer M. Explainable artificial intelligence and machine learning: A reality rooted perspective. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery; 2020, p. 1–13. http://dx.doi.org/10.1002/widm.1368.

[13] Liang Y, Li S, Yan C, Li M, Jiang C. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. Neurocomputing 2021;419:168–82. http://dx.doi.org/10.1016/j.neucom.2020.08.011.

[14] Das A, Rad P. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. 2020, p. 1–24, arXiv:2006.11371.

[15] Garcia R, Telea A, Silva BB, Tørresen J, Comba J. A task-and-technique centered survey on visual analytics for deep learning model engineering. Comput Graph 2018;77:30–49. http://dx.doi.org/10.1016/j.cag.2018.09.018.

[16] Chatzimparmpas A, Martins RM, Jusufi I, Kerren A. A survey of surveys on the use of visualization for interpreting machine learning models. Inf Vis 2020;19(3):207–33. http://dx.doi.org/10.1177/1473871620904671.

[17] Hohman F, Kahng M, Pienta RS, Chau DH. Visual analytics in deep learning: An interrogative survey for the next frontiers. IEEE Trans Vis Comput Graph 2019;25:2674–93. http://dx.doi.org/10.1109/TVCG.2018.2843369.

[18] Das S, Agarwal N, Venugopal D, Sheldon F, Shiva S. Taxonomy and survey of interpretable machine learning method. In: IEEE symposium series on computational intelligence (SSCI). 2020, p. 670–7. http://dx.doi.org/10.1109/SSCI47803.2020.9308404.

[19] Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 2018;6:52138–60. http://dx.doi.org/10.1109/ACCESS.2018.2870052.

[20] Choo J, Liu S. Visual analytics for explainable deep learning. IEEE Comput Graph Appl 2018;38(4):84–92. http://dx.doi.org/10.1109/MCG.2018.042731661.

[21] Ripley BD. Pattern recognition and neural networks. Cambridge University Press; 2007, http://dx.doi.org/10.1017/CBO9780511812651.

[22] Publication-ready NN architecture schematics. 2016, http://alexlenail.me/NN-SVG. [Accessed 20 2020].

[23] Rai A. Explainable AI: from black box to glass box. J Acad Mark Sci 2020;48(1):137–41. http://dx.doi.org/10.1007/s11747-019-00710-5.

[24] Rodríguez N, Pisoni G. Accessible cultural heritage through explainable artificial intelligence. In: 28th ACM conference on user modeling, adaptation and personalization. 2020, p. 317–24. http://dx.doi.org/10.1145/3386392.3399276.

[25] Moradi M, Samwald M. Post-hoc explanation of black-box classifiers using confident itemsets. Expert Syst Appl 2021;165. http://dx.doi.org/10.1016/j.eswa.2020.113941.

[26] Arrieta A, Diaz-Rodriguez N, Ser J, Bennetot A, Tabik S, Barbado A, Garcia S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities, and challenges toward responsible AI. Inf Fusion 2019;58:82–115. http://dx.doi.org/10.1016/j.inffus.2019.12.012.

[27] Schoenborn JM, Althoff K. Recent trends in XAI: A broad overview on current approaches. Methodol Interact ICCBR Workshops. 2019;2567:51–60.

[28] Ribeiro MT, Singh S, Guestrin C. 'Why should i trust you?' Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference of knowledge discovery and data mining. 2016, p. 1135–44. http://dx.doi.org/10.1145/2939672.2939778.

[29] Selvaraju RR, Das A, Vedantam R, Cogswell M, Parikh D, Batra D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int J Comput Vis 2020;128(2):336–59. http://dx.doi.org/10.1007/s11263-019-01228-7.

[30] Bach S, Binder A, Montavon G, Klauschen F, Müller K, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One 2015;10(7):1–46. http://dx.doi.org/10.1371/journal.pone.0130140.

[31] Dragoni M, Donadello I, Eccher C. Explainable AI meets persuasiveness: Translating reasoning results into behavioral change advice. Artif Intell Med 2020;105. http://dx.doi.org/10.1016/j.artmed.2020.101840.

[32] Letham B, Rudin C, McCormick T, Madigan D. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. Ann Appl Stat 2015;9(3):1350–71. http://dx.doi.org/10.1214/15-AOAS848.

[33] Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining. 2015, p. 721–1730. http://dx.doi.org/10.1145/2783258.2788613.

[34] Tan S, Caruana R, Hooker G, Lou Y. Distill-and-Compare: auditing black-box models using transparent model distillation. In: Proceedings of the 2018 AAAI/ACM conference on AI, ethics, and society. 2018, p. 303–10. http://dx.doi.org/10.1145/3278721.3278725.

[35] Lundberg SM, Lee S. A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems (NIPS'17). Hook, NY, USA: Curran Associates Inc. Red; 2017, p. 4768–77. http://dx.doi.org/10.5555/3294996.

[36] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: Proceedings of the 34th international conference on machine learning, in proceedings of machine learning research, vol. 70. 2017, p. 3145–53, Retrieved from http://proceedings.mlr.press/v70/shrikumar17a.html.

[37] Breiman L. Manual on setting up, using, and understanding random forests v3. Tech Rep 2002;4(1):29, https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf. [Accessed 15 2020].

[38] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: Proceedings of the 34th international conference on machine learning, vol. 70. 2017, p. 3319–28. http://dx.doi.org/10.5555/3305890.3306024.

[39] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: 2nd International Conference on Learning Representations. 2014, p. 1–8, URL arXiv:1312.6034.

[40] Ribeiro MT, Singh S, Guestrin C. Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1.. 2018, p. 1527–35, Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/11491.

[41] Zhou B, Khosla A, Lapedriza, A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: IEEE conference on computer vision and pattern recognition (CVPR). 2016, p. 2921–9. http://dx.doi.org/10.1109/CVPR.2016.319.

[42] Zhao J, Karimzadeh M, Masjedi A, Wang T, Zhang X, Crawford M, Ebert D. FeatureExplorer: Interactive feature selection and exploration of regression models for hyperspectral images. In: IEEE visualization conference (VIS). 2019, p. 161–5. http://dx.doi.org/10.1109/VISUAL.2019.8933619.

[43] Brooks M, Amershi S, Lee B, Drucker S, Kapoor A, Simard P. FeatureInsight: Visual support for error-driven feature ideation in text classification. In: IEEE conference on visual analytics science and technology (VAST). 2015, p. 105–12. http://dx.doi.org/10.1109/VAST.2015.7347637.

[44] Krause J, Perer A, Bertini E. INFUSE: Interactive feature selection for predictive modeling of high dimensional data. IEEE Trans Vis Comput Graphics 2014;20(12):1614–23. http://dx.doi.org/10.1109/TVCG.2014.2346482.

[45] Ali M, Jones MW, Xie X, Williams M. TimeCluster: Dimension reduction applied to temporal data for visual analytics. The Visual Computer. 2019;35:1013–26. http://dx.doi.org/10.1007/s00371-019-01673-y.

[46] Hohman F, Wongsuphasawat K, Kery MB, Patel K. Understanding and visualizing data iteration in machine learning. In: Proceedings of the CHI conference on human factors in computing systems. 2020, p. 1–13. http://dx.doi.org/10.1145/3313831.3376177.

[47] May T, Bannach A, Davey J, Ruppert T, Kohlhammer J. Guiding feature subset selection with an interactive visualization. In: IEEE conference on visual analytics science and technology (VAST). 2011, p. 111–20. http://dx.doi.org/10.1109/VAST.2011.6102448.

[48] Zeng H, Haleem H, Plantaz X, Cao N, Qu H. CNNComparator: Comparative analytics of convolutional neural networks. 2017, URL arXiv:1710.05285.

[49] Park C, Lee J, Han H, Lee K. ComDia+: An interactive visual analytics system for comparing, diagnosing, and improving multiclass classifiers. In: IEEE Pacific visualization symposium (PacificVis). 2019, p. 313–7. http://dx.doi.org/10.1109/PacificVis.2019.00044.

[50] Steed C, Goodall J, Chae J, Trofimov A. CrossVis: A visual analytics system for exploring heterogeneous multivariate data with applications to materials and climate sciences. Graph Vis Comput 2020;3:200013. http://dx.doi.org/10.1016/j.gvc.2020.200013.

[51] Murugesan S, Malik S, Du F, Koh E, Lai T. DeepCompare: Visual and interactive comparison of deep learning model performance. IEEE Comput Graph Appl 2019;39(5):47–59. http://dx.doi.org/10.1109/MCG.2019.2919033.

[52] Pühringer M, Hinterreiter A, Streit M. InstanceFlow: Visualizing the evolution of classifier confusion on the instance level. In: IEEE Visualization Conference (VIS). 2020, p. 291–5. http://dx.doi.org/10.1109/VIS47514.2020.00065.

[53] Ren D, Amershi S, Lee B, Suh J, Williams J. Squares: Supporting interactive performance analysis for multiclass classifiers. IEEE Trans Vis Comput Graph 2017;23(1):61–70. http://dx.doi.org/10.1109/TVCG.2016.2598828.

[54] Alsallakh B, Hanbury A, Hauser H, Miksch S, Rauber A. Visual methods for analyzing probabilistic classification data. IEEE Trans Vis Comput Graph 2014;20:1703–12. http://dx.doi.org/10.1109/TVCG.2014.2346660.

[55] Cashman D, Perer A, Chang R, Strobelt H. Ablate, variate, and contemplate: Visual analytics for discovering neural architectures. IEEE Trans Vis Comput Graph 2020;26(1):863–73. http://dx.doi.org/10.1109/TVCG.2019.2934261.

[56] Shen J, Shen H. An Information-theoretic visual analysis framework for convolutional neural networks. 2020, URL arXiv:2005.02186.

[57] Liu M, Shi J, Cao K, Zhu J, Liu S. Analyzing the training processes of deep generative models. IEEE Trans Vis Comput Graph 2018;24(1):77–87. http://dx.doi.org/10.1109/TVCG.2017.2744938.

[58] Wang ZJ, Turko R, Shaikh O, Park H, Das N, Hohman F, Kahng M, Chau DH. CNNExplainer: Learning convolutional neural networks with interactive visualization. IEEE Trans Vis Comput Graph 2020;27:1396–406. http://dx.doi.org/10.1109/TVCG.2020.3030418.

[59] Li G, Wang J, Shen HW, Chen K, Shan G, Lu Z. CNNPruner: Pruning convolutional neural networks with visual analytics. IEEE Trans Vis Comput Graph 2021;27:1364–73. http://dx.doi.org/10.1109/TVCG.2020.3030461.

[60] Pezzotti N, Höllt T, Gemert JV, Lelieveldt B, Eisemann E, Vilanova A. DeepEyes: Progressive visual analytics for designing deep neural networks. IEEE Trans Vis Comput Graph 2018;24(1):98–108. http://dx.doi.org/10.1109/TVCG.2017.2744358.

[61] Liu D, Cui W, Jin K, Guo Y, Qu H. DeepTracker: Visualizing the training process of convolutional neural networks. ACM Trans Intell Syst Technol 2018;10(1):1–25. http://dx.doi.org/10.1145/3200489.

[62] Dang T, Van H, Nguyen HN, Pham V, Hewett R. DeepVix: Explaining long short-term memory network with high dimensional time series data. In: Proceedings of the 11th international conference on advances in information technology. 2020, http://dx.doi.org/10.1145/3406601.3406643.

[63] Zhang J, Wang Y, Molino P, Li L, Ebert DS. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. IEEE Trans Vis Comput Graph 2019;25(1):364–73. http://dx.doi.org/10.1109/TVCG.2018.2864499.

[64] Kwon BC, Choi M, Kim J, Choi E, Kim YB, Kwon S, Sun J, Choo J. RetainVis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. IEEE Trans Vis Comput Graph 2018;25(1):299–309. http://dx.doi.org/10.1109/TVCG.2018.2865027.

[65] Hohman F, Park H, Robinson C, Chau DHPolo. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. IEEE Trans Vis Comput Graph 2020;26(1):1096–106. http://dx.doi.org/10.1109/TVCG.2019.2934659.

[66] Rathore A, Chalapathi N, Palande S, Wang B. TopoAct: Visually exploring the shape of activations in deep learning. Comput Graph Forum 2021;40:1–16. http://dx.doi.org/10.1111/cgf.14195.

[67] Ming Y, Qu H, Bertini E. RuleMatrix: Visualizing and understanding classifiers with rules. IEEE Trans Vis Comput Graph 2019;25(1):342–52. http://dx.doi.org/10.1109/TVCG.2018.2864812.

[68] Zhao X, Wu Y, Lee D, Cui W. IForest: Interpreting random forests via visual analytics. IEEE Trans Vis Comput Graph 2019;25(1):407–16. http://dx.doi.org/10.1109/TVCG.2018.2864475.

[69] Schlegel U, Cakmak E, Keim DA. ModelSpeX: Model specification using explainable artificial intelligence methods. International workshop on machine learning in visualization for big data 2020;1:2–6. http://dx.doi.org/10.2312/mlvis.20201100.

[70] Lamy J, Tsopra R. Visual explanation of simple neural networks using interactive rainbow boxes. In: 23rd international conference information visualisation (IV). 2019, p. 50–5. http://dx.doi.org/10.1109/IV.2019.00018.

[71] Collaris D, Wijk JV. ExplainExplore: Visual exploration of machine learning explanations. In: IEEE Pacific Visualization Symposium (PacificVis). 2020, p. 26–35. http://dx.doi.org/10.1109/PacificVis48177.2020.7090.

[72] Wang J, Gou L, Zhang W, Yang H, Shen H. Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation. IEEE Trans Vis Comput Graph 2019;25(6):2168–80. http://dx.doi.org/10.1109/TVCG.2019.2903943.

[73] Li Y, Fujiwara T, Choi YK, Kim KK, Ma K. A visual analytics system for multi-model comparison on clinical data predictions. Vis Inf 2020;4(2):122–31. http://dx.doi.org/10.1016/j.visinf.2020.04.005.

[74] Yang F, Huang Z, Scholtz J, Arendt D. How do visual explanations foster end users' appropriate trust in machine learning? In: Proceedings of the 25th international conference on intelligent user interfaces. 2020, p. 189–201. http://dx.doi.org/10.1145/3377325.3377480.

[75] Botari T, Izbicki R, Carvalho ACPLF de. Local interpretation methods to machine learning using the domain of the feature space. In: Joint European conference on machine learning and knowledge discovery in databases. 2020, p. 241–52. http://dx.doi.org/10.1007/978-3-030-43823-4_21.

[76] Alvarez-Melis D, Jaakkola TS. Towards robust interpretability with self-explaining neural networks. In: Proceedings of the 32nd international conference on neural information processing systems (NIPS'18). NY, USA: Curran Associates Inc. Red Hook; 2018, p. 7786–95. http://dx.doi.org/10.5555/3327757.

[77] Meske C, Bunde E. Transparency and trust in human-ai-interaction: The role of model-agnostic explanations in computer vision-based decision support. In: Proceedings of the international conference of AI in HCI. 2020. 2020, p. 54–69. http://dx.doi.org/10.1007/978-3-030-50334-5_4.

[78] Kim B, Park J, Suh J. Transparency and accountability in AI decision support: Explaining and visualizing convolutional neural networks for text information. Decis Support Syst 2020;134:113302. http://dx.doi.org/10.1016/j.dss.2020.113302.

[79] Baptista M, Mishra M, Henriques E, Prendinger H. Using explainable artificial intelligence to interpret remaining useful life. 2020, http://dx.doi.org/10.13140/RG.2.2.27721.36963.

[80] Lamy J, Sekar B, Guezennec G, Bouaud J, Séroussi B. Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. Artif Intell Med 2019;94:42–53. http://dx.doi.org/10.1016/j.artmed.2019.01.001.

[81] Li J, Chen X, Hovy E, Jurafsky D. Visualizing and understanding neural models in NLP. In: Conference of the north american chapter of the association for computational linguistics: human language technologies. 2016, p. 681–91. http://dx.doi.org/10.18653/v1/N16-1082.

[82] Islam SR, Eberle W, Ghafoor S. Towards quantification of explainability in explainable artificial intelligence methods. 2019, URL ArXiv:1911.10104.

[83] So C. Understanding the prediction mechanism of sentiments by XAI visualization. In: Proceedings of the 4th international conference on natural language processing and information retrieval. 2020, p. 18–20. http://dx.doi.org/10.1145/3443279.3443284.

[84] Krause J, Dasgupta A, Swartz J, Aphinyanaphongs Y, Bertini E. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In: IEEE conference on visual analytics science and technology (VAST). 2017, p. 162–72. http://dx.doi.org/10.1109/VAST.2017.8585720.

[85] Spinner T, Schlegel U, Schäfer H, El-Assady M. ExplAIner: A visual analytics framework for interactive and explainable machine learning. IEEE Trans Vis Comput Graph 2020;26(1):1064–74. http://dx.doi.org/10.1109/TVCG.2019.2934629.

[86] Chan GY, Bertini E, Nonato L, Barr B, Silva CT. Melody: generating and visualizing machine learning model summary to understand data and classifiers together. 2020, URL arXiv:2007.10614.

[87] Chan GY, Yuan J, Overton K, Barr B, Rees K, Nonato L, Bertini E, Silva CT. SUBPLEX: Towards a better understanding of black box model explanations at the subpopulation level. 37, (4). 2020, URL arXiv:2007.10609.

[88] Lauritsen SM, Kristensen M, Olsen MV, Larsen MS, Lauritsen KM, Jørgensen MJ, Lange J, Thiesson B. Explainable artificial intelligence model to predict acute critical illness from electronic health records. Nature Commun 2020;11(1):1–11. http://dx.doi.org/10.1038/s41467-020-17431-x.

[89] Alber M, Lapuschkin S, Seegerer P, Hägele M, Schütt KT, Montavon G, Samek W, Müller K, Dähne S, Kindermans P. INNvestigate neural networks!. J Mach Learn Res 2019;20:1–8, Retrieved from http://jmlr.org/papers/v20/18-540.html.

[90] Cho S, Lee G, Chang W, Choi J. Interpretation of deep temporal representations by selective visualization of internally activated units. 2020, URL ArXiv:2004.12538.

[91] Angelov P, Soares E. Towards explainable deep neural networks (xDNN). Neural Netw 2020;130:185–94. http://dx.doi.org/10.1016/j.neunet.2020.07.010.

[92] Lu Y, Garcia R, Hansen B, Gleicher M, Maciejewski R. The State-of-the-Art in predictive visual analytics. Comput Graph Forum 2017;36(3):539–62. http://dx.doi.org/10.1111/cgf.13210.

[93] Jolliffe IT. Principal component analysis. 2nd ed. 2002, http://dx.doi.org/10.1007/b98835, New York, NY.

[94] McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform manifold approximation and projection. J Open Source Softw 2018;3:861. http://dx.doi.org/10.21105/joss.00861.

[95] Yuan J, Chen C, Yang W, Liu M, Xia J, Liu S. A survey of visual analytics techniques for machine learning. Comput Vis Media 2021;7:3–36. http://dx.doi.org/10.1007/s41095-020-0191-7.

[96] Kwon BC, Eysenbach B, Verma J, Ng K, deFilippi C, Stewart W, Perer A. A Clustervision: Visual supervision of unsupervised clustering. IEEE Trans Vis Comput Graph 2018;24(1):142–51. http://dx.doi.org/10.1109/TVCG.2017.2745085.

[97] Maaten LV, Hinton GE. Visualizing data using t-SNE. J Mach Learn Res 2008;9:2579–605, Retrieved from https://www.jmlr.org/papers/v9/vandermaaten08a.html.

[98] Heer J, Bostock M, Ogievetsky V. A Tour through the Visualization Zoo. Queue 2010;8(5):20–30. http://dx.doi.org/10.1145/1794514.1805128.

[99] Joia P, Coimbra D, Cuminato J, Paulovich F, Nonato L. Local affine multidimensional projection. IEEE Trans Vis Comput Graph 2011;17:2563–71. http://dx.doi.org/10.1109/TVCG.2011.220.