Human-in-the-Loop Integration with Domain-Knowledge Graphs for Explainable Federated Deep Learning

Andreas Holzinger[1,2(✉)], Anna Saranti[1,2], Anne-Christin Hauschild[3], Jacqueline Beinecke[3], Dominik Heider[4], Richard Roettger[5], Heimo Mueller[2], Jan Baumbach[6], and Bastian Pfeifer[2]

[1] University of Natural Resources and Life Sciences, Vienna, Austria
andreas.holzinger@human-centered.ai
[2] Medical University of Graz, Graz, Austria
[3] University of Göttingen, Göttingen, Germany
[4] University of Marburg, Marburg, Germany
[5] University of Southern Denmark, Odense, Denmark
[6] University of Hamburg, Hamburg, Germany

Abstract. We explore the integration of domain knowledge graphs into Deep Learning for improved interpretability and explainability using Graph Neural Networks (GNNs). Specifically, a protein-protein interaction (PPI) network is masked over a deep neural network for classification, with patient-specific multi-modal genomic features enriched into the PPI graph's nodes. Subnetworks that are relevant to the classification (referred to as "disease subnetworks") are detected using explainable AI. Federated learning is enabled by dividing the knowledge graph into relevant subnetworks, constructing an ensemble classifier, and allowing domain experts to analyze and manipulate detected subnetworks using a developed user interface. Furthermore, the human-in-the-loop principle can be applied with the incorporation of experts, interacting through a sophisticated User Interface (UI) driven by Explainable Artificial Intelligence (xAI) methods, changing the datasets to create counterfactual explanations. The adapted datasets could influence the local model's characteristics and thereby create a federated version that distils their diverse knowledge in a centralized scenario. This work demonstrates the feasibility of the presented strategies, which were originally envisaged in 2021 and most of it has now been materialized into actionable items. In this paper, we report on some lessons learned during this project.

Keywords: Artificial Intelligence · Explainable AI · Machine Learning · Human-in-the-Loop · Graph Neural Networks · Federated Learning · Counterfactual Explanations

A. Holzinger et al. (Eds.): CD-MAKE 2023, LNCS 14065, pp. 45–64, 2023.
https://doi.org/10.1007/978-3-031-40837-3_4

List of Abbreviations

AI	Artificial Intelligence
CLARUS	interaCtive expLainable plAtform for gRaph neUral networkS
DNA	Deoxyribo-Nucleic Acid
FC	FeatureCloud (EU Project)
GDPR	General Data Protection Regulation
GNN	Graph Neural Network
GNN-LRP	GNN Layer-wise Relevance Propagation
GPU	Graphics Processing Unit
HITL	Human-in-the-Loop
IG	Integrated Gradients
i.i.d.	Independent and identically distributed
LRP	Layerwise Relevance Propagation
MI	Mutual Information
ML	Machine Learning
mRNA	messenger Ribo-Nucleic Acid
OOD	Out-Of-Distribution
PGM	Probabilistic Graphical Model Explainer
UI	User Interface
xAI	explainable Artificial Intelligence

1 Introduction and Motivation

The European Project "FeatureCloud (FC)" (Grant Agreement 826078) created a novel Artificial Intelligence (AI) platform which is based on the idea of federated, decentralised learning where only model parameters are communicated. The FC AI App-store https://featurecloud.ai/ is the first platform worldwide to enable federated learning of diverse AI models in a privacy-preserving way [41]. The types of AI models used are quite diverse, including linear regression, clustering, random forests, deep learning, etc. The fundamental idea is that every software developer or data scientist can federate their AI model provided that the model fulfils some minimum requirements (see: https://featurecloud.eu). Dockerization [43] supports seamlessly the transferability of the federated solution into different machines independent from hardware requirements as much as possible.

Whilst federated decentralized learning enables communication of model parameters, integration with more advanced machine learning concepts, such as deep learning and domain-specific knowledge, can increase its performance and efficiency. Using deep neural networks and enriching them with domain-specific graphs such as protein-protein interaction (PPI) networks can also drastically improve the feature extraction process. The next phase, of course, is then about combining decentralization and the power of Deep Learning. The feature-rich, detailed, and robust parameters, when communicated in a federated learning framework, can lead to highly effective and reliable machine learning applications. The decentralized nature of such a framework not only increases learning

efficiency but also strengthens the trustworthiness of the results by combining masked learning with domain knowledge.

In our work [48], we masked deep neural network learning with a protein-protein interaction (PPI) network. In the context of this paper, "masking" refers to incorporating a domain-knowledge graph (specifically, a PPI network) into a deep neural network for classification. This means that the nodes and edges of the PPI network are added to the input layer of the neural network and are used to enrich the features of the data being processed by the neural network. Features are key for learning, understanding and explaining and consolidated features are more accurate and robust, which helps to make practical machine learning applications more trustworthy [47]. It is a general problem that even the most powerful learning methods suffer from the fact that it is difficult to retrace, interpret and thus explain why a certain result was obtained, and that they lack robustness. Even the smallest perturbations in the input data can dramatically affect the output, leading to completely different results. This is of great importance in virtually all critical domains where we suffer from poor data quality, i.e., where we do not have available the i.i.d. data we would need for ideal learning. However, in medicine, biology, and all life-critical domains, it is about being able to trust the results and retrace them when needed [17,18].

In our next step the classification has been made explainable, i.e. those subnetworks are detected that were relevant for the classification ("disease subnetworks") - subgraphs are called "local spheres" in [20] and [40]. In order to guarantee a representative baseline comparison to the above methodology, the subnetwork detection was realised by means of a random forest [45]. Here, too, the learning process is masked by a knowledge graph. Random forests are particularly relevant in medicine due to their good interpretability. In the work [46] we enabled federated learning with the methods mentioned above. Here, the knowledge graph is divided into relevant subnetworks using explainable AI, based on which an ensemble classifier is constructed. This ensemble classifier can be efficiently learned in a federated way. In addition, a user interface was developed [2] that allows a domain expert to analyse and manipulate the detected subnetworks, delete and add nodes, and finally reintegrate them into the federated ensemble classifier. This paper is organized as follows: In Sect. 2 we provide some background and related work, in Sect. 3 we provide an overview of our implementations, and in Sect. 4 we give a frank description of what we have learned, and in Sect. 5 we conclude and provide some future outlook.

2 Background and Related Work

There is nothing more practical than a good theory (Kurt Lewin, (1890–1947)). In our work we pursued four central topics from the paper [20]: (i) Explainable AI on GNNs, (ii) Federated Learning, (iii) Knowledge Graphs, and (iv) Human-AI interaction. Consequently, we have aligned all of these topics on the application of precision medicine.

2.1 Explainable AI on Graph Neural Networks

Graph Neural Networks (GNNs) extend neural network architectures to operate on graph-based data by defining learnable functions that extract features and patterns from the graph structure to perform tasks such as node classification, graph classification, link prediction, etc. [58]. GNNs are very successful and enable efficient integration of domain-knowledge graphs to make Deep Learning interpretable and explainable [20]. Federated solutions thereof seem to occur naturally in several applications such as distributed sensors for traffic surveillance, a collaboration of hospitals for efficient solutions of complex medical tasks, distributed social media applications and so on. In the era of big data both the size of the graph datasets as well as the GNN architectures grows, making efficient and privacy-preserving information exchange and computation a challenge. What is more, since the communicating parties, whether they are servers or clients can be represented by a graph themselves, it is shown that GNN architectures can support federation in turn [33].

As is generally the case with neural networks, also GNN results are not easy to retrace and interpret. To address this shortcoming, intensive work is currently being done worldwide on GNN methods that can be explained. Examples include GNNexplainer, PGExplainer, and GNN-LRP. *GNNExplainer* [59], for example, provides *local* explanations for predictions of any graph-based model. This can be used for both node classification and graph classification. *PGExplainer* [35] is a parameterized modification of GNNexplainer. Unlike GNNexplainer, it provides model-level explanations that we find useful for graph classification tasks. *GNN-LRP* [51] is derived from higher-order Taylor expansions based on layer-wise relevance propagation (LRP) [30]. It explains the prediction by extracting paths from the input to the output of the GNN model that makes the largest contribution to the prediction. These paths correspond to *walks* on the input graph. GNN-LRP was developed for node-level explanations and has been modified to work for graph classification in a special arrangement [5]. The presented work with a method called CF-Explainer [34] is particularly interesting. Here, explanatory factors can be revealed using counterfactuals.

GCExplainer [38] stands in the forefront as the first GNN explainer that detects the *learned concepts* of a GNN. The main idea is to perform clustering after the last aggregation layer and to assume that each of the clusters corresponds to a human-recognizable concept. Users have the opportunity to parameterize the explanation process through the number of clusters and the neighbourhood size of the explained component. This approach incorporates the human-in-the-loop [16,23] and at the same time has been shown to achieve good concept purity and completeness. Furthermore, it is the basis of current work that makes GNNs explainable per design by first learning the concepts, then on that basis doing a concept-based prediction [37]. Such explainable AI methods can facilitate the discovery of disease-causing regions in networks, helping to uncover a subset of *candidate features* organized in disease-relevant network modules.

This is exactly where the human-in-the-loop concept helps, as interaction with explanations and the incorporation of conceptual knowledge can further improve the learning algorithm.

2.2 Federated Learning

Federated learning (FL) is an ML approach in which the training data is decentralized and distributed across multiple devices or locations, and the model training process is performed locally on each device or location [40]. The updates to the model are then aggregated centrally, resulting in a global model that incorporates the knowledge learned from each device or location. FL is of course useful in scenarios where the data is sensitive, private, or subject to regulatory constraints, such as medical records or financial transactions. Instead of centralizing the data and running the model training process on a single server or cloud platform, federated learning allows the data to remain on individual devices or locations, and only the model updates are transmitted for aggregation. This preserves the privacy and security of the data and reduces the risk of data breaches or leaks. FL should not be mixed up with purely decentralized learning, where local models do not automatically contribute to each other apart from manually sampling the models and updating the hyperparameters [3]; and also not with collaborative learning in various forms, where the goal is to share information about internal model building between the involved parties in a peer-to-peer manner but keep the local training data confidential. A variant could also train on decentralized features that purportedly model the same underlying instances [24]. It has been known for some time that features for one modality are learned better when multiple modalities are present at the time of feature learning. In multimodal learning, information is from multiple sources. Often, several different modalities contribute to a result. We are motivated by [1,9,19]. This brings us directly to graphs and particularly knowledge graphs.

Federation itself has evolved to be a broad topic; although the main principles are firm, different implementations realize the same goals. What is similar in all instantiations is that there is data isolation to some degree and that the information being exchanged should be minimal and privacy-preserved (i.e. encrypted). Furthermore, the i.i.d. scenario is rather the exception than the norm; several frameworks need to simulate it before the actual deployment [44]. Nonetheless, collaboration has proven to be fruitful in most cases, since no one dataset contains all representative information about a task and ML solutions lack the ability of systematic generalization and out-of-distribution (OOD) prediction even when trained with rich and diverse datasets.

In the more concrete case of Federated GNN, there are mainly three possibilities [14], as also shown in Fig. 1. In the graph-level FL, each client has its graph dataset and potentially also a GNN. In the subgraph-level FL, each of the clients has one part of the graph and in the node-level FL nodes of one graph are distributed among clients.

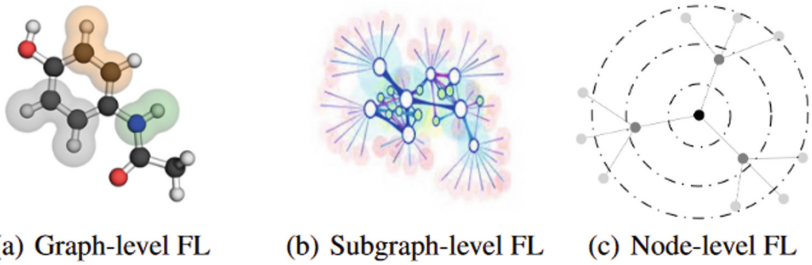

Fig. 1. Three settings of GNN federation [14].

This is following the principles of Horizontal FL (HFL) and Vertical FL (VFL). In the first case, the features of the graphs of all clients are quite similar, but their sample characteristics (data distribution) differ substantially. The opposite occurs in the second case. Both of them are viable scenarios of FL and need to be addressed either with centralized or decentralized FL. In the federated centralized strategy, it is typical that there are several synchronous or asynchronous events containing parts of the dataset, and one server is responsible for the federation (which is also called aggregation). In the federated decentralized case, many clients exchange information with each other; this is more robust as far as privacy attacks are concerned but has substantial communication and organizational overhead.

2.3 Knowledge Graphs

Knowledge graphs (KG) are a type of database that represents knowledge in a structured, interconnected format, using a graph-based data model. It typically consists of a set of nodes (also called entities) that represent concepts or things, and a set of edges (also called relationships or properties) that connect the nodes and represent the connections or interactions between them. Many phenomena from nature can be represented in graph structures, whether at the molecular level (e.g. protein-protein interaction) or at the macroscopic level (e.g. social networks) and various methods from network science [7] and computational topology [15] can be applied. Some of the most successful application areas of machine learning and knowledge extraction in recent years can be seen as learning with graph representations [57].

In a knowledge graph, each node and edge can have additional attributes or metadata associated with it, providing additional information or context about the node or edge. This metadata can include labels, descriptions, categories, or other semantic information. Knowledge graphs are often used to represent information from diverse sources and domains in a multi-modal manner. They can be used to represent both factual knowledge (such as the properties of objects or events) and conceptual knowledge (such as the relationships between abstract

concepts). Knowledge graphs are also used as a foundation for various applications, such as natural language processing, semantic search, recommendation systems, and data integration. They enable efficient querying and reasoning about complex, heterogeneous data, as well as support the development of intelligent agents that can reason and learn from the knowledge represented in the graph [12]. KG's are very useful for explainability and explainable AI methods based on counterfactual queries to the trained GNN models are very promising [39,53].

2.4 Human-in-the-Loop

Human-in-the-Loop [16] refers to the process of involving a human expert interactively in the machine learning (ML) process to provide feedback, guidance, or even corrections to the model. The human is an integral part of the ML pipeline, interacting with the model/algorithm to improve its performance and ensuring that it aligns with the desired goals and values. This approach is useful in scenarios where the data is complex, ambiguous, or subject to change, and where the model's performance can benefit from human expertise or even from the experts' subjective judgment. This is because sometimes - of course not always - the human expert has domain knowledge, experience and contextual understanding, in German "Hausverstand" - what the best AI algorithms are lacking today. An additional benefit is that the human-in-the-loop approach can also improve the transparency, interpretability, and fairness of machine learning models, as it allows for human oversight and intervention in cases where the model produces biased or undesirable results. However, the human-in-the-loop approach, on the other hand, has drawbacks as it can be time-consuming, expensive, and potentially introduce bias or subjectivity into the modelling process, so it is important to carefully design and evaluate the interaction between the human and the model.

3 Methods, Solutions and Implementations

3.1 Disease Subnetwork Detection

In a publication about GNNSubNet [48], we presented a novel method for disease subnetwork detection using protein-protein interaction (PPI) networks and explainable graph neural networks (GNN). Our method leveraged the PPI knowledge to enable more reliable and biologically meaningful learning trajectories compared to classical deep learning approaches. The nodes of the induced PPI network are enriched by biological features from various modalities, such as gene expression and DNA methylation (see Fig. 2). We applied our proposed method to patients with kidney cancer and demonstrated its ability to detect disease subnetworks. The developed methodology is implemented within

our GNN-SubNet Python package, freely available on GitHub (https://github.com/pievos101/GNN-SubNet). In addition, we enhance ensemble learning based on the detected networks. This makes the classifier more robust, but also more interpretable [46]. Ensemble-learning with GNNs is implemented within our Ensemble-GNN Python package (https://github.com/pievos101/Ensemble-GNN). In further updates of the package additional GNN-based explainers such as GNN-LRP and PGM-Explainer to further increase the interpretability of the detected subnetworks will be implemented.

Moreover, as a reliable baseline, in terms of classification performance and overlay interpretability, we have developed the software package DFNET (https://github.com/pievos101/DFNET) [45], which implements a network-guided random forest to derive an ensemble classifier based on any induced knowledge-graph. However, in a federated case, a local random forest would need to share the exact split values of its nodes [13]. This is of much concern and was one of the reasons why we further developed federated solutions based on deep GNNs. The shared parameter among clients in that deep learning setting is more secure with regard to privacy concerns.

3.2 Explainability

The classification of Part 1 has been made explainable, i.e. those subnetworks detected that were relevant for the classification ("disease subnetworks") - subgraphs aka "local spheres". For this purpose we have developed a modified version of the GNNexplainer [59] to compute global explanations. This is realized by sampling patient-specific input graphs while optimizing a single-node mask (see Fig. 2). From these values, edge weights are calculated and assigned to the edges of the PPI network. Finally, a weighted community detection algorithm infers the relevant subnetworks.

PPI networks generally provide crucial insights into cellular functions and processes, and alterations in these interactions often lead to diseases. Consequently, such networks are important in understanding complex diseases like cancer, which typically involve changes in the interaction patterns of proteins. Explainability can here help to understand disease mechanisms, e.g. to reveal the underlying mechanisms of diseases. By understanding which interactions contribute to the prediction and how, researchers can potentially uncover new biological insights. For example, a model might predict a certain protein as being critical to a disease because of its numerous interactions with other proteins. This could lead to further biological investigations into the role of that protein in the disease. This can help in creating personalized treatment strategies. For instance, if certain protein interactions are critical in the disease progression of a particular patient, treatments can be tailored to target these specific interactions. Identifying which features (e.g., specific proteins or interactions) are most important in the model's predictions. For example, a model might reveal that a specific protein or a set of proteins plays a significant role in a particular disease, informing further biological research.

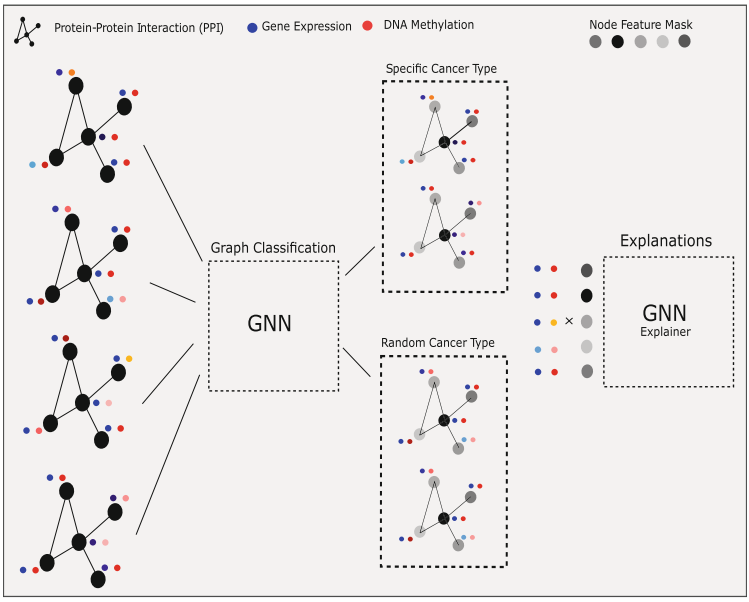

Fig. 2. Illustration of patient classification into a cancer-specific and randomized cancer group using explainable Graph Neural Networks (taken from [48]). Each patient is represented by the topology of a protein-protein interaction network (PPI). Nodes are enriched by multi-omic features from gene expression and DNA Methylation (coloured circles). The topology of each graph is the same for all patients, but the node feature values vary, reflecting the cancer-specific molecular patterns of each patient. (Color figure online)

Furthermore, *model-agnostic* counterfactual explanations and their associated counterfactual paths can be generated using our *cpath* software library (https://github.com/pievos101/cpath). The implemented methodology provides counterfactual explanations by identifying alternative paths that could have led to different predictions. The proposed method is particularly suited to generate explanations based on counterfactual paths on knowledge graphs. By exploring hypothetical changes to the input data on the knowledge graph, we can systematically validate the behaviour of the model and investigate the features, or combination of features, that are most important for the model's predictions. Our approach provides a more intuitive and interpretable explanation of the model's behaviour than traditional feature importance methods and can help to identify and mitigate biases in the model. A scientific paper about *cpath* is in progress.

3.3 Knowledge Graph

GNNs provide a crucial benefit of enabling the integration of knowledge graphs [27]. This implies that both ontologies and Protein-Protein Interaction (PPI)

networks can be effectively incorporated into the algorithmic pipeline, as highlighted in much previous research [26,29,32,54]. This also enables to integrate of human experience, conceptual knowledge, and contextual understanding into machine learning architectures, which is a notable advantage. This "human-in-the-loop" or "expert-in-the-loop" approach can, in some cases, lead to more robust, reliable, and interpretable results [22,23,25].

PPIs reflect the physical or functional connections between proteins in a cell or organism. These networks can be represented as graphs, where proteins are nodes and their interactions are edges. PPIs can be retrieved from the STRING database [56]. STRING provides a comprehensive collection of known and predicted protein interactions, allowing users to explore and analyze protein networks to gain insights into cellular processes and functional relationships.

It is worth noting that the inclusion of domain knowledge does not guarantee success in every instance. However, the incorporation of such expertise can contribute to the attainment of the most critical goals of the AI community, namely, the development of robust, explainable and trustworthy solutions [18]. These objectives are essential in ensuring the practical and ethical applications of AI in various fields and are meanwhile mandatory e.g. in the European Union.

3.4 Federated Ensemble Learning with GNNs

In recent work [46] we enabled federated learning with the methods mentioned above. Here, the knowledge graph is divided into relevant subnetworks using explainable AI, based on which an ensemble classifier is constructed. This ensemble classifier can be efficiently learned in a federated way.

The main idea of the ensemble federation is depicted in Fig. 3. Each client contains several graphs and each of those graphs represents a patient. The values of the nodes and edges are different in general (as depicted by the different colours of the nodes in the upper part of Fig. 3), but the structure of the graphs is the same. Those graphs can be classified by a GNN and the GNN-SubNet method [48] can compute a set of relevant subgraphs for this classification. GNN-SubNet concentrates on providing the relevant structure or topology only; therefore the subgraphs are depicted with white in the middle of Fig. 3. The concrete values of the nodes and edges are transferred in a third step though from the original graphs (upper part of Fig. 3) to the concrete subgraphs that have the topology of the relevant subgraphs and values overtaken from the original graph (lower part of Fig. 3). By creating a new dataset for each discovered relevant subgraph where its structure is repeated and the values are taken from the original graph of all the patients in the client, a separate GNN is trained. The predictions of all those GNNs are input to a majority vote procedure that - in its non-federated version - has an acceptable local performance.

The federation is depicted in Fig. 4 and follows a decentralized strategy. The clients use local GNNs of their peers in the inter-client network, that were created with similar logic but were trained with graphs having different topologies - since the relevant subgraphs for each client are expected to vary in general. There is no exchange of the discovered relevant topologies of each client, only

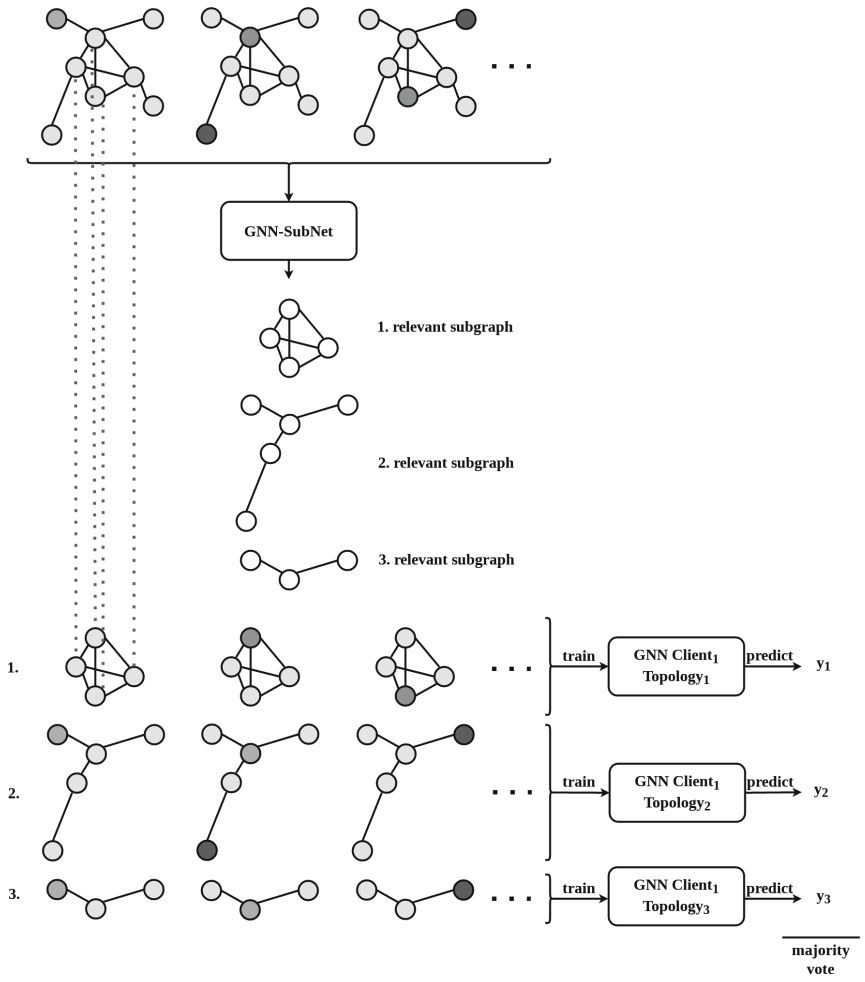

Fig. 3. The use of GNN-SubNet in one client, containing a set of graphs for classification. This method extracts a list of relevant subgraph structures (topologies) and uses them by filling the corresponding values of nodes and edges from the original graphs. The newly created datasets are used to train local GNNs and make predictions which are aggregated by majority voting.

the GNN parameters are transferred - which is as far as privacy is concerned less revealing. The majority vote over all those GNNs provided a better performance over each client's test set, but not over a test set that was isolated from all clients, as shown in [46]. The described methodology is implemented within our Python package Ensemble-GNN, freely available on GitHub (https://github.com/pievos101/Ensemble-GNN). A Feature Cloud app implementation is also available (https://github.com/pievos101/fc-ensemble-gnn).

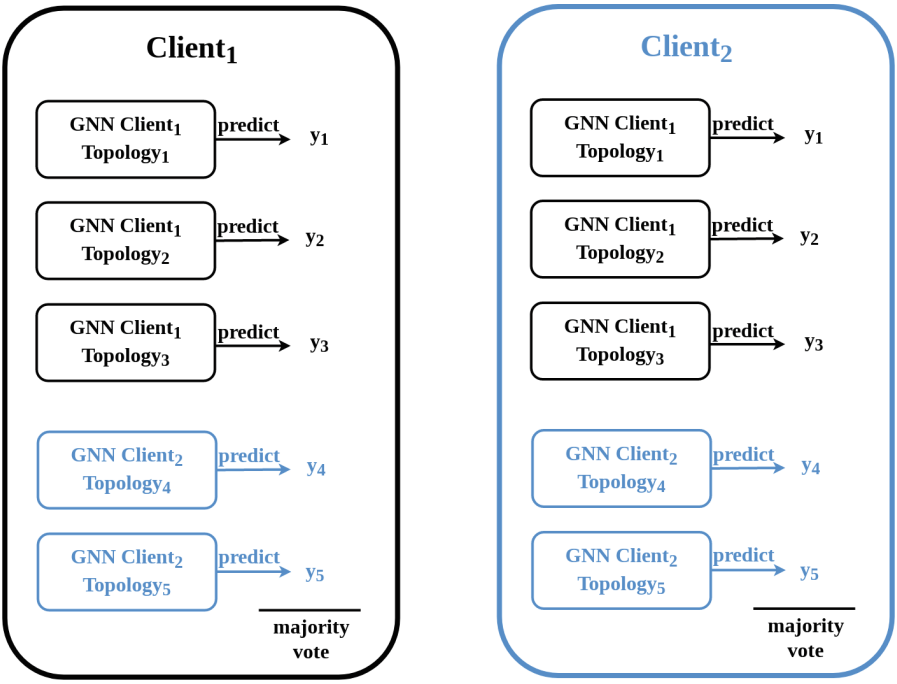

Fig. 4. Depiction of the federated learning of Ensemble-GNN. The late fusion of exchanged GNN's predictions through voting is the way the federation is driven by the result of the employed xAI method in [48].

The scenario of non-i.i.d. data has to be simulated in future work, by including imbalanced distribution of data and potentially explicitly defining different feature distributions in the clients [44]. Lastly, the discovered relevant topologies can also be subject to changes driven by human users through a UI, changing the local GNNs, and by that the whole federation process.

3.5 interaCtive expLainable plAtform for gRaph neUral networkS (CLARUS)

The CLARUS UI platform [2] is accessible under http://rshiny.gwdg.de/apps/clarus/. The goal of the UI platform is to provide any human user interactive access to prepared datasets, GNNs and several xAI methods. All necessary information about the platform usage, datasets, features and performance metrics are provided through the platform. An overview of the typical sequence of steps that a user takes is presented in Fig. 5.

For the user to be able to make informed actions [49] with the use of diverse xAI methods (GNNExplainer [59], GCExplainer [38]), all nodes and edges are presented by sorted relevance values. The colouring scheme depends on the properties of the xAI method itself; the saliency method [55], Integrated Gradients

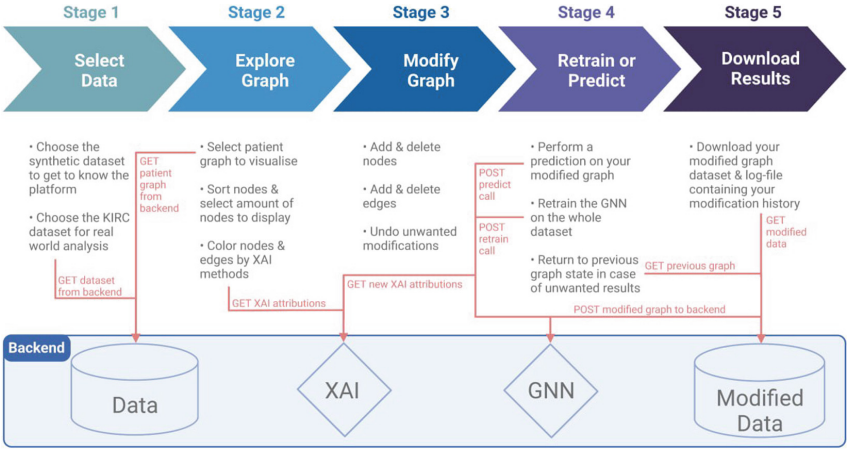

Fig. 5. The sequence of user action steps in the CLARUS platform. First, the user selects one of the prepared datasets and immediately after he/she has the opportunity to explore any graph visually by zooming and by inspecting the nodes and edges feature values. The backend has already trained a GNN with the training dataset after a stratified split of the data and presents performance results (individual and global), xAI relevance values as well as additional information that can be useful such as the degree of each node. With the help of this information, and additional acquired domain knowledge, the human user decides to take action(s) and either add or delete nodes, edges and features thereof. To see how those actions affected the task prediction of the current GNN a new prediction can be triggered. In cases where the changes are substantial a retrain from scratch can be also made, deleting by that all old information in the current GNN. This process can be repeated as many times as desired until the user conceives the decision-making process to an acceptable extent through the generated counterfactual explanations. A download of all data and model details at a particular time point, together with a unique timestamp is possible on demand.

(IG) method, and the GNNExplainer return only positive relevance values, but methods like GNN-LRP (Layer-wise Relevance Propagation) return both positive and negative values. Those two groups of relevance value ranges have discrete colourings for a better understanding of the concept of negative relevance as one denoting element in the data sample that "speak against" a class and even in a correct classification is responsible for making the confidence value smaller. Beyond that, for each sample it has to be clear if it is correctly classified or misclassified; even the exact prediction performance is present. This is because the reliability of explanations in the misclassification case is questionable and it is a subfield of xAI research itself. Therefore, several classification metrics are accessible: the confusion matrix, sensitivity, specificity and in the future Mutual Information (MI) [4,11,36]. After each retrain and prediction, those metrics are re-computed and in general they have changed values. A detailed description of the pre-selected datasets, their preprocessing, various interaction scenarios and abilities of the platform can be found in [2].

With the use of adequately designed UI tests on this platform it is possible to show the effect of counterfactual questions and corresponding user actions on user understanding of the model. The completeness of the already used xAI methods is enhanced by the actions triggered by users in combination with the already present domain knowledge, but also from the juxtaposition of their results since they all differ to a certain extent. The user is motivated and inspired to make informed actions, imagine what their effect would be and compare the actual result with his/her preconceived notions about why the model solves the task sufficiently well (or not) in a dialectical manner. The path to increasing causability [42] with the use of specially designed interfaces [21] is at the forefront for the causal understanding of AI models in the future. The described user interface CLARUS [2] allows a domain expert to analyse and manipulate the detected subnetworks, which to this end could be reintegrated into the federated ensemble classifier.

4 Lessons Learned

What was not done and why?
The implementation of other xAI methods than GNNExplainer for the detection of disease subnetworks. This is particularly relevant for ensemble-based GNN architectures. Each GNN xAI method might create different ensemble members, which to this end could be studied in terms of performance and interpretability (e.g. GO enrichment of the detected PPI subnetworks).

What problems occurred?
Other xAI methods were more difficult to integrate. Some explainers only compute relevances for edges or nodes, others like GNN-LRP [50] assign relevance values only on walks; that means that a node or edge belonging to more than one walk (which is usually the case) has not one clearly defined relevance value. Data scientists may be tempted to average all edge relevances to infer the relevance of the node or the opposite, but this is not representative of the xAI method. Furthermore, GNN-LRP provides both negative and positive relevances, which means that not only the colour map has to be distinct from the methods that provide only positive relevance, but that the relevance of the paths needs an individual visualization strategy that allows overlapping and user selection. Other methods like the GCExplainer [38] compute a representative set of subgraphs that is relevant for each relevant concept w.r.t. the accomplished task. Although this is a valuable approach which has some similarities with the detection of relevant disease subnetworks, since it does not directly return numerical relevance values for the individual components of the graphs, cannot be straightforwardly integrated into the UI framework.

What was difficult?
What was particularly difficult for both data scientists and users is the discovery of differences between the xAI methods results; this consists of the so-called "disagreement problem" [28]. Data scientists provide several xAI methods to

shed light on different aspects of the design-making process of the model, but if the results of those methods deviate from each other, this disagreement is not easy to interpret and understand. Furthermore, counter-intuitive phenomena were observed; it is assumed for example, that if a user deletes components of a graph according to decreasing (positive) relevance order, then the performance of the model will not only decrease monotonically but also that the newly computed relevance order after a new triggered prediction will remain the same. In many cases this was not experienced, making the users question the reliability of the xAI methods. Related to that, the value range of the colour map was an issue, since the minimum and maximum value of relevance change in general after a prediction is initiated.

What did we learn?
The fact that each graph has the same topology (PPI network) hinders stable and robust graph classification, especially in cases where the input graph is large. We could observe that GNNs on smaller graphs perform generally better [46]. Further, we have learned that in the herein-studied cases of the same topology graphs, using Laplacian layers might be more efficient in terms of performance. Therefore, we also included the ChebNet approach [6] as an option for GNN-SubNet and Ensemble-GNN. However, GNNs are generic models and applicable to many other related tasks. Also, we might model each patient with different graph topologies. In that case, the ChebNet approach is not applicable.

We have further learned that the quality and validity of the knowledge graph are crucial. Knowledge graphs must be further improved in order to obtain reliable and domain-specific meaningful results. Also, it has been shown that most methods for disease module discovery learn from the PPI node degrees and mostly fail to exploit the biological knowledge encoded in the edges of the PPI networks [31]. Although we believe that our proposed methodology is not biased to that described case, further investigations are needed to understand and quantify the bias induced by the network structure.

What open work remains for the future?
Heterogeneous Graphs (including text and images or different types of nodes and edges) were not included. After preliminary tests, we know that they need more resources and xAI methods need to be thoroughly tested before deployment. So far we have multi-model genomic data in tabular form, structured by a PPI network.

Until now the GNN architecture is pre-defined for every dataset and it is somehow intertwined with the characteristics of this dataset - and most of all its size. In the case where the user changes increases or decreases the size of the dataset and/or changes its characteristics substantially, the platform cannot guarantee similar performance since the GNN's architecture is not adapted. To automatically find the adequate GNN architecture is a topic of Automated Machine Learning (Auto-ML), and its incorporation in this platform will come with additional time costs which will, in turn, influence the waiting time of the users in favour of performance and better xAI results.

The existence of the aforementioned "disagreement problem" [28] drives future work in the direction of not only integrating more xAI methods but also considering the computation and presentation of several xAI quality metrics thereof to the users. Fidelity, sensitivity, clusterability, robustness and others [8,52] provide additional guidelines for the reliability of each method in cases where the top relevant features or their ordering is inconsistent. In the end, upon deployment, several UI evaluation tests have to be made to explore the extent of biased preference of xAI relevance results. In the end, the plurality of xAI methods does not necessarily consist of a problem but might be the means for a sophisticated, holistic and dialectic approach for shedding light on different aspects of the decision-making process of GNNs.

The main reason federation is used, is for the central model to learn something from the different local models, trained with their datasets. Comparing the performance of the local models with the central model: what are the differences there?

It does make a considerable difference whether we test the federated global model on an independent global test data set, or on multiple client-specific test data sets (see [46]). It still needs to be investigated which scenario is most relevant and why these differ so much in terms of the performance of the global model.

5 Conclusion and Future Outlook

In this work, we have demonstrated how to make federated deep learning more interpretable and accessible to the domain expert. First, we have incorporated domain knowledge into the deep learning process using Graph Neural Networks and Protein-Protein Interaction (PPI) networks. Second, we have decomposed the PPI knowledge graph into more interpretable smaller subnetworks using explainable AI. Based on these subnetworks an ensemble classifier is constructed which can be learned in a federated manner. The shared parameters of this deep learning ensemble are more secure compared to e.g. the shared split values of decision trees in a federated random forest. Finally, the ensemble member (subnetworks) can be analysed by a domain expert through an interactive UI.

Future work can be done from various directions. Until now, xAI methods that were used (GNNExplainer, PGExplainer, GCexplainer) return relevant values of nodes, edges and features thereof. Apart from the fact that some fundamental principles of them need to be explained to the users (f.e. that the GNN-LRP assigns relevance to walks and not directly to nodes and edges), the interpretation of those numerical values is a task that the user's mental model needs to undertake. In contrast to that, explanations in the form of rules, provide a completely different user experience and understanding. It would be interesting to research how Logical Rules (e.g. with Prolog) guide the selection of subnetworks [10], similarly or differently with the numerical relevance values.

Furthermore, a framework that asks the domain expert about their preconceived notions as far as what parts of the input data should be important, before

seeing xAI results is worthwhile studying. The comparison of users' reactions after confronting relevant values vs. uninfluenced opinions derived from their knowledge before any interaction could uncover interesting effects of human-AI interaction.

Acknowledgements. The authors declare that there are no conflict of interests. This work does not raise any ethical issues. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 826078 (Feature Cloud). This publication reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains. Parts of this work have been funded by the Austrian Science Fund (FWF), Project: P-32554 (explainable Artificial Intelligence). This paper has been made open access CC-BY, freely accessible to the international research community. We are grateful for the valuable reviewer comments.

References

1. Acosta, J.N., Falcone, G.J., Rajpurkar, P., Topol, E.J.: Multimodal biomedical AI. Nat. Med. **28**(9), 1773–1784 (2022)
2. Beinecke, J., et al.: CLARUS: an interactive explainable AI platform for manual counterfactuals in graph neural networks. bioRxiv (2022). https://doi.org/10.1101/2022.11.21.517358
3. Bellavista, P., Foschini, L., Mora, A.: Decentralised learning in federated deployment environments: a system-level survey. ACM Comput. Surv. (CSUR) **54**(1), 1–38 (2021)
4. Bishop, C.M., Nasrabadi, N.M.: Pattern Recognition and Machine Learning, vol. 4. Springer, New York (2006)
5. Chereda, H., et al.: Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. Genome Med. **13**(1), 1–16 (2021). https://doi.org/10.1186/s13073-021-00845-7
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. arXiv:1606.09375 [cs, stat] (2016)
7. Dehmer, M., Emmert-Streib, F., Shi, Y.: Quantitative graph theory: a new branch of graph theory and network science. Inf. Sci. **418**, 575–580 (2017). https://doi.org/10.1016/j.ins.2017.08.009
8. Doumard, E., Aligon, J., Escriva, E., Excoffier, J.B., Monsarrat, P., Soulé-Dupuy, C.: A quantitative approach for the comparison of additive local explanation methods. Inf. Syst. **114**, 102162 (2023)
9. Ektefaie, Y., Dasoulas, G., Noori, A., Farhat, M., Zitnik, M.: Multimodal learning with graphs. Nat. Mach. Intell. **5**(4), 340–350 (2023)
10. Finzel, B., Saranti, A., Angerschmid, A., Tafler, D., Pfeifer, B., Holzinger, A.: Generating explanations for conceptual validation of graph neural networks. KI-Künstl. Intell. **36**, 271–285 (2022). https://doi.org/10.1007/s13218-022-00781-7
11. Géron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media (2019)
12. Hamilton, W., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding logical queries on knowledge graphs. In: Advances in Neural Information Processing Systems, vol. 31 (2018)

13. Hauschild, A.C., et al.: Federated Random Forests can improve local performance of predictive models for various healthcare applications. Bioinformatics **38**(8), 2278–2286 (2022). https://doi.org/10.1093/bioinformatics/btac065
14. He, C., et al.: FedGraphNN: a federated learning benchmark system for graph neural networks. In: ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML) (2021)
15. Holzinger, A.: On topological data mining. In: Holzinger, A., Jurisica, I. (eds.) Interactive Knowledge Discovery and Data Mining in Biomedical Informatics. LNCS, vol. 8401, pp. 331–356. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43968-5_19
16. Holzinger, A.: Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inform. **3**(2), 119–131 (2016). https://doi.org/10.1007/s40708-016-0042-6
17. Holzinger, A.: The next frontier: AI we can really trust. In: Kamp, M., et al. (eds.) ECML PKDD 2021. CCIS, vol. 1524, pp. 427–440. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-93736-2_33
18. Holzinger, A., et al.: Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence. Inf. Fusion **79**(3), 263–278 (2022). https://doi.org/10.1016/j.inffus.2021.10.007
19. Holzinger, A., Haibe-Kains, B., Jurisica, I.: Why imaging data alone is not enough: AI-based integration of imaging, omics, and clinical data. Eur. J. Nucl. Med. Mol. Imaging **46**(13), 2722–2730 (2019). https://doi.org/10.1007/s00259-019-04382-9
20. Holzinger, A., Malle, B., Saranti, A., Pfeifer, B.: Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI. Inf. Fusion **71**(7), 28–37 (2021). https://doi.org/10.1016/j.inffus.2021.01.008
21. Holzinger, A., Müller, H.: Toward human-AI interfaces to support explainability and causability in medical AI. IEEE Comput. **54**(10), 78–86 (2021). https://doi.org/10.1109/MC.2021.3092610
22. Holzinger, A., Plass, M., Holzinger, K., Crisan, G.C., Pintea, C.M., Palade, V.: Towards interactive machine learning (iML): applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In: Buccafurri, F., Holzinger, A., Kieseberg, P., Tjoa, A., Weippl, E. (eds.) CD-ARES 2016. LNCS, vol. 9817, pp. 81–95. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-45507-5_6
23. Holzinger, A., et al.: Interactive machine learning: experimental evidence for the human in the algorithmic loop. Appl. Intell. **49**(7), 2401–2414 (2019). https://doi.org/10.1007/s10489-018-1361-5
24. Hu, Y., Niu, D., Yang, J., Zhou, S.: Stochastic distributed optimization for machine learning from decentralized features, pp. 1–10. arXiv:1812.06415 (2018)
25. Hudec, M., Minarikova, E., Mesiar, R., Saranti, A., Holzinger, A.: Classification by ordinal sums of conjunctive and disjunctive functions for explainable AI and interpretable machine learning solutions. Knowl. Based Syst. **220**, 106916 (2021). https://doi.org/10.1016/j.knosys.2021.106916
26. Jeanquartier, F., Jean-Quartier, C., Holzinger, A.: Integrated web visualizations for protein-protein interaction databases. BMC Bioinform. **16**(1), 195 (2015). https://doi.org/10.1186/s12859-015-0615-z
27. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans. Neural Netw. Learn. Syst. **33**(2), 494–514 (2022). https://doi.org/10.1109/TNNLS.2021.3070843
28. Krishna, S., et al.: The disagreement problem in explainable machine learning: a practitioner's perspective. arXiv preprint arXiv:2202.01602 (2022)

29. Kulmanov, M., Smaili, F.Z., Gao, X., Hoehndorf, R.: Machine learning with biomedical ontologies. bioRxiv (2020). https://doi.org/10.1101/2020.05.07.082164
30. Lapuschkin, S., Binder, A., Montavon, G., Müller, K.R., Samek, W.: The LRP toolbox for artificial neural networks. J. Mach. Learn. Res. (JMLR) **17**(1), 3938–3942 (2016)
31. Lazareva, O., Baumbach, J., List, M., Blumenthal, D.B.: On the limits of active module identification. Briefings Bioinform. **22**(5), bbab066 (2021)
32. Liu, G., Wong, L., Chua, H.N.: Complex discovery from weighted PPI networks. Bioinformatics **25**(15), 1891–1897 (2009). https://doi.org/10.1093/bioinformatics/btp311
33. Liu, R., Yu, H.: Federated graph neural networks: overview, techniques and challenges. arXiv preprint arXiv:2202.07256 (2022)
34. Lucic, A., ter Hoeve, M., Tolomei, G., de Rijke, M., Silvestri, F.: CF-GNNExplainer: counterfactual explanations for graph neural networks. arXiv:2102.03322 (2021)
35. Luo, D., et al.: Parameterized explainer for graph neural network. In: Advances in Neural Information Processing Systems, vol. 33, pp. 19620–19631 (2020)
36. MacKay, D.J.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
37. Magister, L.C., et al.: Encoding concepts in graph neural networks. arXiv e-prints arXiv:2207.13586 (2022)
38. Magister, L.C., Kazhdan, D., Singh, V., Liò, P.: GCExplainer: human-in-the-loop concept-based explanations for graph neural networks. arXiv preprint arXiv:2107.11889 (2021)
39. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv:1912.03277 (2019)
40. Malle, B., Giuliani, N., Kieseberg, P., Holzinger, A.: The more the merrier - federated learning from local sphere recommendations. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2017. LNCS, vol. 10410, pp. 367–373. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66808-6_24
41. Matschinske, J., et al.: The featurecloud AI store for federated learning in biomedicine and beyond (2021). https://doi.org/10.48550/arXiv.2105.05734. arXiv:2105.05734
42. Müller, H., Holzinger, A., Plass, M., Brcic, L., Stumptner, C., Zatloukal, K.: Explainability and causability for artificial intelligence-supported medical image analysis in the context of the European In Vitro Diagnostic Regulation. New Biotechnol. **70**, 67–72 (2022). https://doi.org/10.1016/j.nbt.2022.05.002
43. Naik, N.: Migrating from virtualization to dockerization in the cloud: simulation and evaluation of distributed systems. In: 2016 IEEE 10th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA), pp. 1–8. IEEE (2016). https://doi.org/10.1109/MESOCA.2016.9
44. Ortega, A., Frossard, P., Kovačević, J., Moura, J.M., Vandergheynst, P.: Graph signal processing: overview, challenges, and applications. Proc. IEEE **106**(5), 808–828 (2018)
45. Pfeifer, B., Baniecki, H., Saranti, A., Biecek, P., Holzinger, A.: Multi-omics disease module detection with an explainable greedy decision forest. Sci. Rep. **12**(1), 1–15 (2022). https://doi.org/10.1038/s41598-022-21417-8
46. Pfeifer, B., et al.: Ensemble-GNN: federated ensemble learning with graph neural networks for disease module discovery and classification. bioRxiv (2023). https://doi.org/10.1101/2023.03.22.533772

47. Pfeifer, B., Holzinger, A., Schimek, M.G.: Robust random forest-based all-relevant feature ranks for trustworthy AI. Stud. Health Technol. Inform. **294**, 137–138 (2022). https://doi.org/10.3233/SHTI220418
48. Pfeifer, B., Saranti, A., Holzinger, A.: GNN-SubNet: disease subnetwork detection with explainable graph neural networks. Bioinformatics **38**(S-2), ii120–ii126 (2022). https://doi.org/10.1093/bioinformatics/btac478
49. Saranti, A., et al.: Actionable explainable AI (AxAI): a practical example with aggregation functions for adaptive classification and textual explanations for interpretable machine learning. Mach. Learn. Knowl. Extract. **4**(4), 924–953 (2022). https://doi.org/10.3390/make4040047
50. Schnake, T., et al.: Higher-order explanations of graph neural networks via relevant walks. arXiv preprint arXiv:2006.03589 (2020)
51. Schnake, T., et al.: XAI for graphs: explaining graph neural network predictions by identifying relevant walks. arXiv:2006.03589 (2020)
52. Schwalbe, G., Finzel, B.: A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. Data Min. Knowl. Discov. 1–59 (2023). https://doi.org/10.1007/s10618-022-00867-8
53. Singh, R., et al.: Directive explanations for actionable explainability in machine learning applications. arXiv:2102.02671 (2021)
54. Staab, S., Studer, R.: Handbook on Ontologies. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-92673-3
55. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning, pp. 3319–3328. PMLR (2017)
56. Szklarczyk, D., et al.: The STRING database in 2021: customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Res. **49**(D1), D605–D612 (2021)
57. Veličković, P.: Everything is connected: graph neural networks. Curr. Opin. Struct. Biol. **79**, 102538 (2023). https://doi.org/10.1016/j.sbi.2023.102538
58. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2021). https://doi.org/10.1109/TNNLS.2020.2978386
59. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: generating explanations for graph neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)