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

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

[1] Medical University of Graz, Austria
andreas.holzinger@medunigraz.at
[2] University of Natural Resources and Life Sciences Vienna, Austria
[3] University of Göttingen, Germany
[4] University of Marburg, Germany
[5] University of Southern Denmark, Denmark
[6] University of 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 influence the local model's characteristics and thereby create a federated version that distills 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

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 [54]. 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 [57] supports seamlessly the transferability of the federated solution into different machines independent from hardware requirements as much as possible.

In our work [63], we masked deep neural network learning with a protein-protein interaction (PPI) network. In the context of this work, "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 [61]. 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 have dramatic effects on 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 [26], [27].

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 [29] and [52]. In order to guarantee a representative baseline comparison to the above methodology, the subnetwork detection was realised by means of a random forest [59]. 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 [60] 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 [4] that allows a domain expert to analyse and manipulate the detected subnetworks (delete and add nodes) and finally reintegrate them into the ensemble classifier. This paper is organized as follows: In section 2 we provide some background and related work, in section 3 we provide an overview on our implementations, and in section 4 we give a frank description of what we have learned, and in section 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 [29]: (i) Explainable AI on GNNs, (ii) Federated Learning and Multi-Modality, (iii) Knowledge Graphs, and (iv) Human-AI interaction. Consequently, we have aligned all of these topics on the application of precision medicine.

2.1 (i) 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. [75]. GNN's are very successful and enable efficient integration of domain-knowledge graphs to make Deep Learning interpretable and explainable [29]. 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 [45].

As is generally the case with neural networks, also Graph Neural Networks 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* [76], for example, provides *local* explanations for predictions of any graph-based model. This can be used for both node classification and graph classification. *PGExplainer* [47] is a parameterized modification of GNNexplainer. Unlike GNNexplainer, it provides model-level explanations that we find useful for graph classification tasks. *GNN-LRP* [66] is derived from higher-order Taylor expansions based on layerwise relevance propagation (LRP) [42]. It explains the prediction by extracting paths from the input to the output of the GNN model that make 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 [11]. The presented work with a method called CF-Explainer [46] is particularly interesting. Here, explanatory factors can be revealed using counterfactuals.

GCExplainer [50] 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

[32], [25] 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 [49]. 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 (ii) Federated Learning and Multi-Modality

Federated learning (FL) is a 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 [52]. 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 the 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 [5]; 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 [33]. 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 [28], [1], [15]. This brings us directly to graphs and particularly knowledge graphs.

2.3 (iii) 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 macroscopic level (e.g.

social networks) and various methods from network science [14] and computational topology [24] 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 [74].

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 [21]. KG's are very useful for explainability and explainable AI methods based on counterfactual queries to the trained GNN models are very promising. A major advantage of counterfactual generation is that it can be viewed as a post-hoc method and thus can act independently of any classifier [68]. Here, counterfactuals can be modeled as a graph in which features (e.g., genes) are defined as nodes and edges refer to combinations of features (called "counterfactual paths"). First, the counterfactual graph is generated in a purely data-driven manner. Given a test set that includes a sufficient number of patients, this algorithm then traverses the feature space and subsequently swaps the features. Specifically, for each patient, the feature values are swapped with the values of the nearest neighbor, which is labeled with a different class. In this procedure, the feature values are swapped until the patient's outcome class changes. The path leading to this change is reflected in the counterfactual graph as a subnetwork or, more precisely, a walk through the counterfactual graph. The feature values used in this walk are stored in a node feature vector. The distance between counterfactuals (weighted edges) is defined in a similar way as in [51].

2.4 (iv) Human-in-the-loop

Human-in-the-Loop [25] 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 the human's 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 modeling process, so it is important to carefully design and evaluate the interaction between the human and the model.

The aforementioned explainable AI methods can facilitate the discovery of disease-causing regions within the networks thereby contribute to uncover a subset of *candidate features* organized in disease relevant network modules. Such methods can be used for validation of their applicability to the biomedical domain, fostering better decision making through interacting with explanations via the human-in-the-loop approach, and there are numerous successful examples [73], [9], [67], [3], [7].

Explainable AI methods should be based on counterfactual queries to the trained GNN models. A major advantage of generating counterfactuals is that it can be viewed as a post-hoc process that can act independently of any classifier. This, of course, requires modeling the resulting counterfactuals as a graph in which features (e.g., genes) are defined as nodes and edges refer to combinations of features. We call such an approach "counterfactual paths". First, the counterfactual graph is generated in a purely data-driven manner. Given a test set that includes a sufficient number of patients, an algorithm then traverses the feature space and subsequently swaps the features. Specifically, for each patient, the feature values are swapped with the values of the nearest neighbor, which is labeled with a different class. In this process, the feature values are exchanged until the patient's outcome class changes. The path leading to this change is reflected in the counterfactual graph as a subnetwork or, more precisely, a walk through the counterfactual graph. The feature values used in this walk are stored in a node feature vector. If necessary, counterfactuals can be generated using amplification techniques. For example, the distance between counterfactuals (weighted edges) can be defined in a manner similar to [51].

The above methodology is applicable as a *model-agnostic* tool for structureless vector spaces, and is therefore particularly suitable for embedded vector spaces. However, for an input structured by graphs, a counterfactual sampling method exchanges feature values on trajectories of the input graph. In this way, the functional relationships between features are preserved and the algorithm can be used as an explainable AI method [38].

It is known that communities (strongly connected features) within the counterfactual graph are similar in context. As a result, strongly disconnected communities indicate substantially different classes of counterfactuals. This provides a global view of the counterfactual landscape, facilitating human interaction with the counterfactual graph and ultimately enabling adaptation of the multimodal machine learning model. It is important to highlight semantically similar subgraphs so that human experts can visually explore the counterfactual graph in an efficient manner. Communities that have neither semantic content nor lead

to causal understanding can thus be revised by a human who has the necessary domain knowledge and experience. To do this, the human could define, adopt, or modify paths in the loop (see figure below). The path modifications can thereby be integrated as knowledge-based constraints in the stochastic procedures used to train high-end learning models. External resources such as the ClinVar [41] or the STRING [72] database can be used to further enrich these constraints in the process. This can be repeated until the human experts are satisfied with the *"what-if"* answers of the learning models. Other forms of developing textual descriptions of concepts and explanations include Inductive Logic Programming [8],[12], [18], and interactive symbolic AI methods [70] such as DeepProbLog [53].

The biomedical experts can not only interact with the counterfactual graph, but also explore easy-to-understand decision trees derived from the counterfactual graph. To do this, the counterfactual graph itself could be transformed into a Decision Forest (DF) classifier that includes multiple trees. The features reflected by the counterfactual communities can in turn be used to create independent Decision Trees (DT) that form a novel ensemble classifier [62]. The algorithm to generate the counterfactual paths is implemented within our *cpath* software library (`https://github.com/pievos101/cpath`). A corresponding scientific paper is in progress.

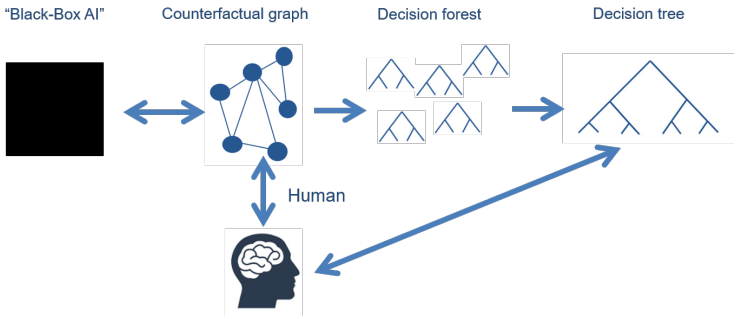

Fig. 1. A possible interaction: In our approach the "human-in-the-loop" has the opportunity to study the counterfactual graph and the derived decision trees or subgraphs of the ensemble classifier; thereby the domain expert will be able to adopt the modifications to the ensemble classifier which approximates the black-box model (feedback-loop).

3 Methods, Solutions and Implementations

3.1 (1) Disease Subnetwork Detection

As first step in [63], 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. 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 [60]. 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 we have developed the software package DFNET (`https://github.com/pievos101/DFNET`) [59], which implements a network-guided random forest which to this end can also be used to derive an ensemble classifier from the above described counterfactual graph (Section 2.4).

3.2 (2): Explainability

The classification of Part 1 has been made explainable, i.e. those subnetworks are 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 [76] to compute global explanations. This is realized by sampling patient-specific input graphs while optimizing a single node mask. 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.

Furthermore, *model-agnostic* counterfactual explanations and their associated counterfactual paths can be generated using our *cpath* software library (`https://github.com/pievos101/cpath`).

3.3 (3): Knowledge Graph

GNNs provide a crucial benefit of enabling the integration of knowledge graphs [37]. This implies that both ontologies and PPI networks can be effectively incorporated into the algorithmic pipeline, as highlighted in much previous research [69], [40], [44], [36]. This also enables to integrate 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 [31], [32], [35]. It is worth noting that the inclusion of a domain expert 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 [27]. 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 (4): Explainable Federated Deep Learning

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. Nonetheless, collaboration has proven to be fruitful in most cases, since no one dataset contains all representative information about a task and AI solutions lack the ability of systematic generalization and out-of-distribution (OOD) prediction unless trained with rich and diverse datasets.

In the more concrete case of Federated GNN, there are mainly three possibilities [23], as also shown in figure 2. 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.

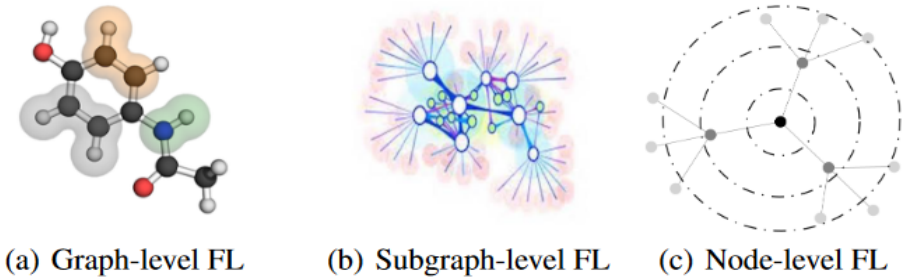

Fig. 2. Three settings of GNN federation [23].

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 centralized strategy, it is typical that there are several synchronous or asynchonous events containing parts of the dataset (as shown in figure 2), and one server is responsible for the federation (which is also called aggregation). In the 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.

Regardless of the client-server topology, there is an inter-client graph that can either be known a priori or can be discovered through self-attention mechanisms (this can also be helpful in the case where new clients enter the client-server topology). There are several ways to implement federation for all four combinations of FL possibilities, which in some centralized cases even needs alternating local and global optimization. Each of them has different convergence guarantees (if any) and individual countermeasures for the expected privacy attacks.

What is more, the client-server topology or the inter-client graph (in the decentralized case), have also a graph structure. Although the basic aggregation procedure (both in centralized and decentralized versions) is the averaging weighted by the number of samples in each client [55], there are other types of more sophisticated federation. Inspired by the ideas of learned aggregation functions of GNNs itself, more sophisticated handling of weights and biases were invented, as seen in figure 3.

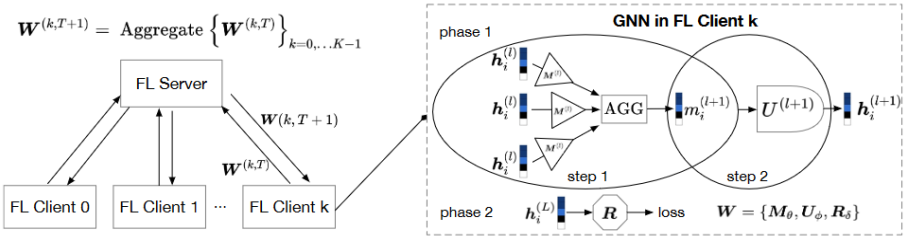

Fig. 3. FedGraphGNN learned aggregation procedure for federation, as presented in [23].

The whole aggregation functionality can be solved by a GNN that takes as input the topology along with weights, gradients or even embeddings (provided they've been sent encrypted) as node and edge features, and returns new parameters in each federation round. Whether the topology is known beforehand or is changing, GNN-assisted FL is an emerging area of research [45]. In many real-world application cases, the assumption is that clients that are "close" have similar data, thereby their local GNNs will probably also have similar parameters. In this case, the result of the trained GNN is practically the federation function, which goes beyond FedAVg [55] and FedOpt[2] possibilities.

Ensemble In the work [3] we enable 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.

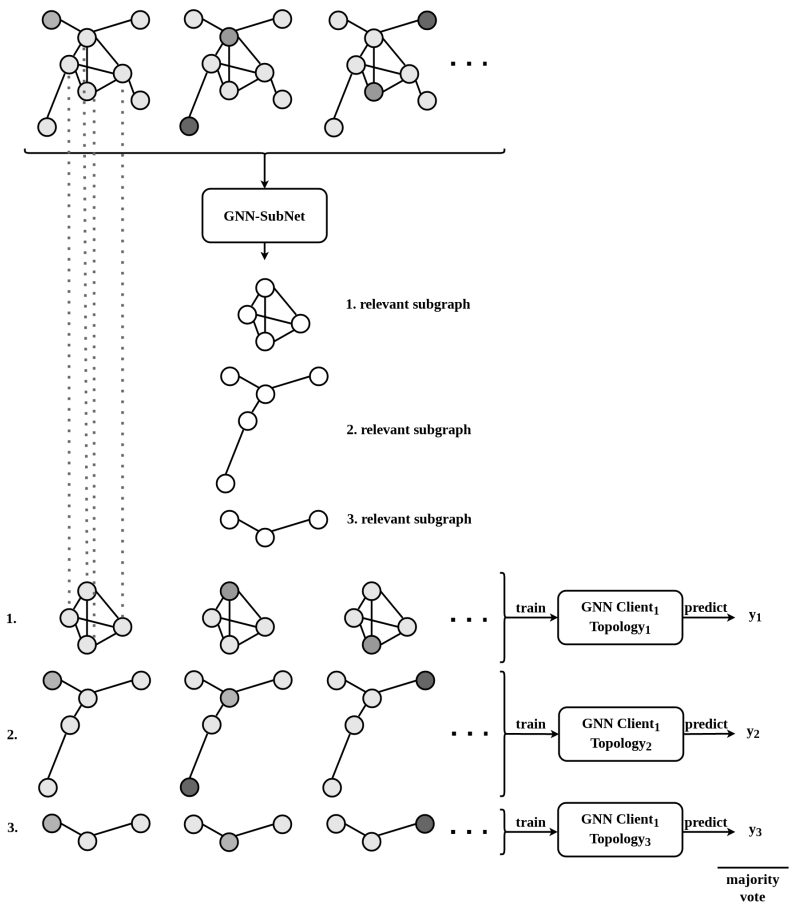

Fig. 4. The use of GNN-SubNet in one client, containing a set of graphs for classification. This method extracts a list of relevant subgraph structures (topologies) and uses them by filling the corresponding values of nodes and edges from the original graphs. The newly created datasets are used to train local GNNs and make predictions which are aggregated by majority voting.

The main idea of the ensemble federation is depicted in figure 4. 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 figure 4), but the structure of the graphs is the same. Those graphs can be classified by a GNN and the GNN-SubNet method [63] 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 figure 4. The concrete values of the nodes and edges are transferred in a third step though from the original graphs (upper part of figure 4) to the concrete subgraphs that have the topology

of the relevant subgraphs and values overtaken from the original graph (lower part of figure 4). 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 figure 5 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 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 [60]. 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`).

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. 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.

Centralized FL with xAI and HITL The main goal of the federation is to create an AI model that distills knowledge from diverse datasets, that share some pre-defined commonalities either in the feature space (Horizontal FL) or in the data space (Vertical FL). One characteristic example in the medical domain is the existence of medical records for different patients in different hospitals. Provided that the data are gathered to diagnose the same disease, one can fairly assume that all hospitals gather similar information. Nevertheless, one cannot completely exclude a situation where a hospital or a doctor decides to gather more or different information than the others. This creates a situation where the number of input features of the AI model is not the same among all hospitals. Furthermore, the number of samples (corresponding to one case or one patient) of each hospital is different; particularly for the cases where a small dataset is gathered one would like to have a more powerful AI model, which is not possible unless one has enough data.

Due to privacy constraints, it is not possible to share information between the hospitals in general. Due to the General Data Protection Regulation (GDPR), personal data cannot be shared. Nonetheless, it is less risky to send information about the AI models, such as weights, gradients, or embeddings although it has been shown that one can extract valuable information even from those, in the case of a successful attack [19], [22], [10]. Since there are ways to counteract those attacks [34], [77], [16], this research work concentrates on the simulation of the

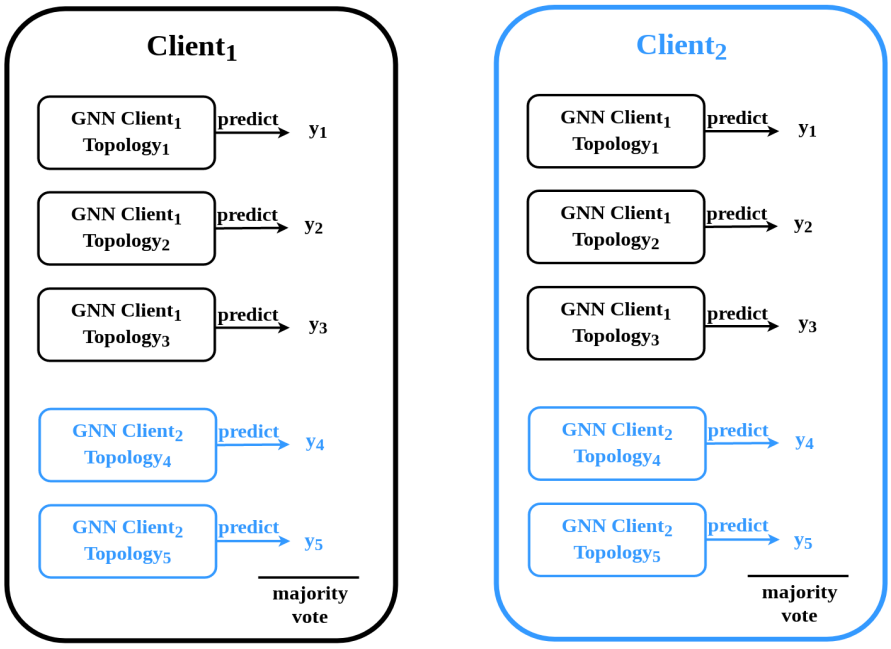

Fig. 5. Depiction of the federated learning of Ensemble-GNN. The late fusion of exchanged GNN's predictions through voting is the way federation is driven by the result of the xAI method.

information exchange containing Graph Neural Network weights, gradients and embeddings, and the aspects of privacy are handled by colleagues in the Feature Cloud Project - as they are taken over for the other AI models in the FC Cloud store as well.

The basic model of centralized federation applied in the Counterfactuals platform is presented in figure 6. It is composed of several clients, each of them operating independently from each other, and one central server. Each of the clients contains its dataset and its own GNN model, although there must be some pre-agreement between them as far as the similarities and differences of the datasets are concerned. It is expected that the size of the graph and the number of the features of each dataset can also vary; for this task which encompasses graph classification, different graph sizes are not a problem - as long as the type of nodes and edges is the same, meaning that the features have to be equal (up to their values). The platform does not support heterogeneous graphs yet; this is a prerequisite for federating graph datasets of different types of nodes and edges. For the basic federation scheme to work, all clients and the server have to have the same GNN architecture.

As also seen in figure 6, it is expected that in the first round, all clients train their own local GNN, each of them with their dataset. The weights of the first GNN are described by $\mathbf{W_1}$, the ones of the second $\mathbf{W_2}$ and so on. At some

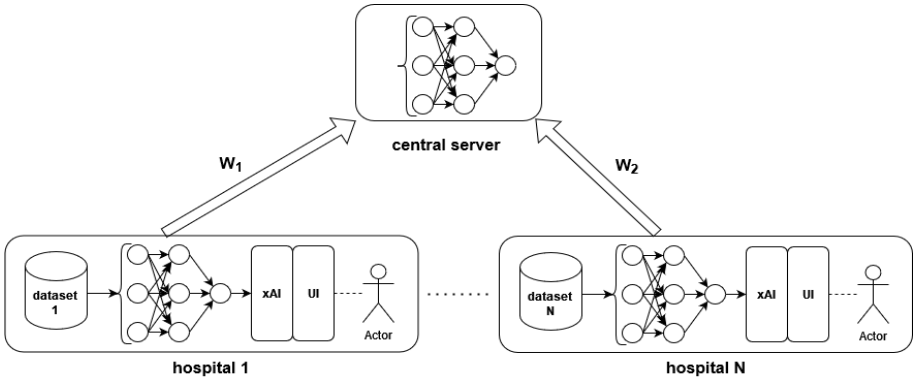

Fig. 6. Federated Learning Overview: Each hospital has its own dataset with different characteristics, but also some similarities with the others.

particular point which can either be a) periodic or b) asynchronous, the weights of one or more of those GNNs are sent to the server. There, they will be averaged, as described in the research work [55], and the resulting weights **W** are going to be sent back to each of the clients. The GNN of each client **can** use the weights **W** and replace its weights with it, or not.

Typically after the adoption of **W** weights one detects lower performance in each of the individual client GNNs, which is expected since those weights are not tailored to the distribution of the local dataset. Nevertheless, the adoption of those weights prepares the GNN for adequate generalization in cases where similar patients/diseases (whatever the graph represents) as the ones presented to other clients, occur in the future - even if there are not as many as the ones the other models from other clients have been trained to. One example scenario would be as follows: a patient coming to hospital 1 that has similar characteristics with several patients of the other hospitals will highly likely not have a good diagnosis with the first GNN, if he/she is different from the patients already registered in hospital 1. Even re-training with this one new data sample in the dataset will not be enough to influence the weights of the first GNN towards a direction where this (more or less) outlier will need for producing a correct prediction in the output. What is more, there are cases reported where the central GNN model weights **W** were proven to be better - as far as performance goes - than the local GNN model [23]. The reasons for that are currently unknown, they could be probably uncovered through Explainable AI (xAI) methods and they are a very interesting direction for future work.

One fundamental difference of the approach with the use of the xAI Counterfactuals platform is that the model parameters change only after retraining; this is decided by a human user and is does not occur because some new patient or new disease information has entered or was removed from the local dataset. Those two processes are both occur asynchronously. Two strategies can be developed: either the weights are gathered periodically (even if for some local clients

they haven't changed at all - one can "catch" it with a request) or each time a client retrains, after the retrain is finished, the weights can be sent to the server and a new average can be computed.

The scenario of different hospitals needs to be simulated; to achieve this, the dataset that we already work with (PPI) needs to be split in a way that simulates non-identically independent (non-i.i.d.) distributed data [58]. That means that the balance of each of the client datasets needs to differ. Since the task is a binary classification, it must be ensured that there are local datasets containing f.e. 70% of their data belonging to class 0 and 30% of their data belonging to class 1, but also other ones having f.e. 60% of their data belonging to class 1 and 40% of their data belonging to class 0. The logic of creating this imbalance and the number of clients should be configurable. This is the most fundamental type of non-i.i.d. simulation for the data distribution; in a future step there is the need to synthetically generate non-i.i.d. topology data [45].

3.5 Interactive User Interface

In addition, a user interface was developed that allows a domain expert to analyse and manipulate the detected subnetworks (delete and add nodes) and finally reintegrate them into the ensemble classifier.

The incorporation of the human-in-the-loop can be also made even more directly with the use of the GCEXplainer. This can be central component of an encompassing framework for Graph Concept Interpretation [39] which measures quantitatively the interpretation-concept alignment using the number of samples for which the interpretation function is true. Good performance in several quality metrics such as concept completeness, predictability, coherence, and purity creates the basis for characterizing the interpretations of this method as overall "relevant".

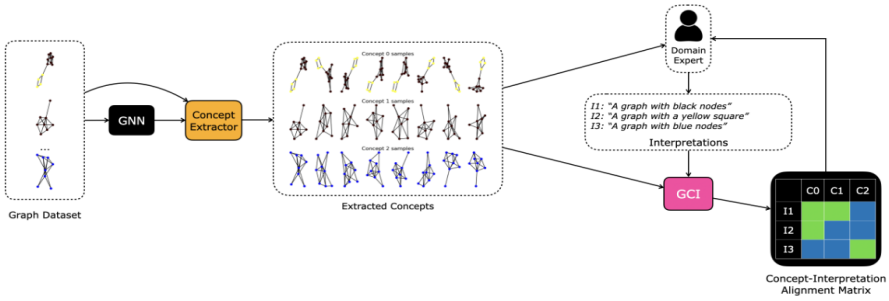

Fig. 7. Implementation of the Human-in-the-Loop principle with an actionable interaction between the human and the AI solution (here a local GNN) through xAI and a User-Interface [39].

interaCtive expLainable plAtform for gRaph neUral networkS (CLARUS) The CLARUS UI platform 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 figure 8.

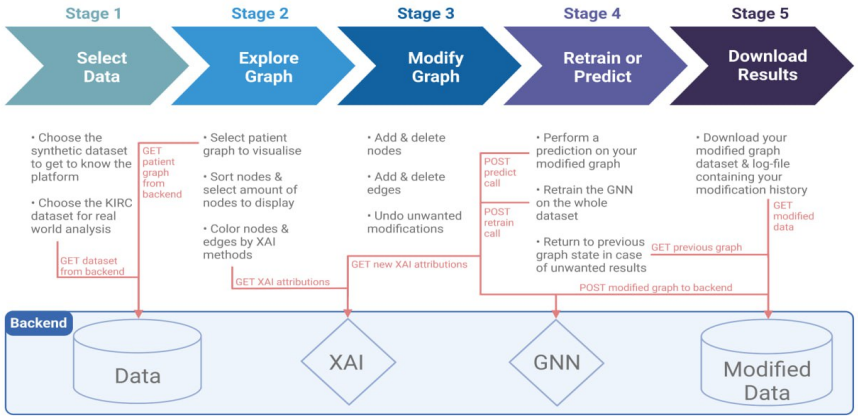

Fig. 8. The sequence of user action steps in the CLARUS platform. First, the user selects one of the prepared datasets and immediately after he/she has the opportunity to explore any graph visually by zooming and by inspecting the nodes and edges feature values. The backend has already trained a GNN 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.

For the user to be able to make informed actions [64] with the use of diverse xAI methods (GNNExplainer [76], GCExplainer [50]), 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 [71], Integrated Gradients (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) [6], [20], [48]. 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 [4].

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 [56] with the use of specially designed interfaces [30] is at the forefront for the causal understanding of AI models in the future.

4 Lessons learned

What was not done and why?
The implementation of other explainer than GNNexplainer for the detection of disease subnetworks. This is particularly relevant for the ensemble-based GNN architectures. Each GNN explainer 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).

GNN-LRP [65] implements Layer-wise Relevance Propagation on GNNs and assigns relevance values on walks; that means that a node or edge belonging to more than walks (which is usually the case) has not one relevance; taking a simple mean is not representative of the explanation method. This method 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.

What problems occurred?
Other explanations than GNNexplainer were more difficult to implement than

expected. Some explainers only compute relevances for edges or nodes. The 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.

What was the most pressing problem?
Computational resource sometimes was a bottleneck. It is important to have Graphics Processing Unit (GPU), since without it the tasks cannot be solved in a timely manner.

What was easy?
Using the Pytorch Geometric software `https://pytorch-geometric.readthedocs.io/en/latest/` and its built-in explainability package Captum `https://captum.ai/` was fairly straightforward. Furthermore, the tutorials help the data scientists to understand the theory thoroughly.

What was difficult?
There is a constant and rapid development on the theoretical foundations of GNNs and their explanations. The associated research field is very competitive. To not get confused or overwhelmed, an in-depth knowledge about the research field is required, from theory to practical solutions and concrete implementations. Otherwise it may lead to miscalculations of the planed project goals.

What was particularly difficult for both data scientists and users is the discovery of differences between the xAI methods results. 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 [60]. Further, we have learned that in the herein studied cases of same topology graphs, using laplacian layers might be more efficient in terms of performance. Therefore, we also included the ChebNet approach [13] 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 is 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 [43]. 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 ressources and xAI methods need to be thouroughly 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 increase or decrease the size of the dataset and/or change 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 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 [60]). It still needs to be investigated which scenario is most relevant and why these two cases differ so much in terms of the performance of the global model.

5 Conclusion and Future Outlook

The plethora of work in FL shows that it is quite multi-faceted. In the proposed research work triggering a retrain may change the parameters of the local GNN model completely and thereby also influence the central model substantially. If

the federation process has "matured" substantially, meaning that the number of clients is sufficient and the overall FL system has been working with a good performance for a long time, one would like that a new client, influenced by the actions of a user is not substantially perturbed immediately. Therefore ideas in the direction of online federation learning have emerged, which resemble already known solutions that combine local and global embeddings for similar purposes [45]. Future work can also go into the direction of recognizing unusual user behaviour as well as the recommendation of counterfactual actions that were proven to provide substantial insights to other users as well.

Until now, xAI methods that were used (GNNExplainer, PGExplainer, GC-explainer) 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 counterfactuals [17], similarly or differently with the numerical relevance values. Furthermore, a framework that asks the users 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.

6 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

7 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. Nature Medicine **28**(9), 1773–1784 (2022)
2. Asad, M., Moustafa, A., Ito, T.: Fedopt: Towards communication efficiency and privacy preservation in federated learning. Applied Sciences **10**(8), 2864 (2020)
3. Baur, T., Heimerl, A., Lingenfelser, F., Wagner, J., Valstar, M.F., Schuller, B., André, E.: explainable cooperative machine learning with nova. KI - Künstliche Intelligenz (2020). https://doi.org/10.1007/s13218-020-00632-3
4. Beinecke, J., Saranti, A., Angerschmid, A., Pfeifer, B., Klemt, V., Holzinger, A., Hauschild, A.C.: Clarus: An interactive explainable ai platform for manual counterfactuals in graph neural networks. bioRxiv p. 2022.11. 21.517358 (2022). https://doi.org/10.1101/2022.11.21.517358
5. Bellavista, P., Foschini, L., Mora, A.: Decentralised learning in federated deployment environments: A system-level survey. ACM Computing Surveys (CSUR) **54**(1), 1–38 (2021)
6. Bishop, C.M., Nasrabadi, N.M.: Pattern recognition and machine learning, vol. 4. Springer (2006)
7. Bodén, A.C., Molin, J., Garvin, S., West, R.A., Lundström, C., Treanor, D.: The human-in-the-loop: an evaluation of pathologists' interaction with artificial intelligence in clinical practice. Histopathology **79**(2), 210–218 (2021). https://doi.org/10.1111/his.14356
8. Bratko, I., Muggleton, S.: Applications of inductive logic programming. Communications of the ACM **38**(11), 65–70 (1995). https://doi.org/10.1145/219717.219771
9. Bruckert, S., Finzel, B., Schmid, U.: The next generation of medical decision support: A roadmap toward transparent expert companions. Frontiers in Artificial Intelligence **3**, 507973 (2020). https://doi.org/10.3389/frai.2020.507973
10. Chen, J., Huang, G., Zheng, H., Yu, S., Jiang, W., Cui, C.: Graph-fraudster: Adversarial attacks on graph neural network-based vertical federated learning. IEEE Transactions on Computational Social Systems (2022)

11. Chereda, H., Bleckmann, A., Menck, K., Perera-Bel, J., Stegmaier, P., Auer, F., Kramer, F., Leha, A., Beißbarth, T.: Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. Genome medicine **13**(1), 1–16 (2021)
12. De Raedt, L., Kersting, K.: Probabilistic inductive logic programming. In: Probabilistic Inductive Logic Programming, pp. 1–27. Springer (2008). https://doi.org/10.1007/978-3-540-78652-8_1
13. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. arXiv:1606.09375 [cs, stat] (Jun 2016), http://arxiv.org/abs/1606.09375
14. Dehmer, M., Emmert-Streib, F., Shi, Y.: Quantitative graph theory: A new branch of graph theory and network science. Information Sciences **418**, 575–580 (2017). https://doi.org/10.1016/j.ins.2017.08.009
15. Ektefaie, Y., Dasoulas, G., Noori, A., Farhat, M., Zitnik, M.: Multimodal learning with graphs. Nature Machine Intelligence pp. 1–11 (2023)
16. Eloul, S., Silavong, F., Kamthe, S., Georgiadis, A., Moran, S.J.: Enhancing privacy against inversion attacks in federated learning by using mixing gradients strategies. arXiv preprint arXiv:2204.12495 (2022)
17. Finzel, B., Saranti, A., Angerschmid, A., Tafler, D., Pfeifer, B., Holzinger, A.: Generating explanations for conceptual validation of graph neural networks. KI-Künstliche Intelligenz pp. 1–15 (2022)
18. Finzel, B., Tafler, D.E., Scheele, S., Schmid, U.: Explanation as a process: user-centric construction of multi-level and multi-modal explanations. In: KI 2021: Advances in Artificial Intelligence: 44th German Conference on AI, Virtual Event, September 27–October 1, 2021, Proceedings 44. pp. 80–94. Springer (2021). https://doi.org/10.1007/978-3-030-87626-5_7
19. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients-how easy is it to break privacy in federated learning? Advances in Neural Information Processing Systems **33**, 16937–16947 (2020)
20. Géron, A.: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media (2019)
21. Hamilton, W., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding logical queries on knowledge graphs. Advances in neural information processing systems **31** (2018)
22. Hatamizadeh, A., Yin, H., Molchanov, P., Myronenko, A., Li, W., Dogra, P., Feng, A., Flores, M.G., Kautz, J., Xu, D., et al.: Do gradient inversion attacks make federated learning unsafe? IEEE Transactions on Medical Imaging (2023)
23. He, C., Balasubramanian, K., Ceyani, E., Yang, C., Xie, H., Sun, L., He, L., Yang, L., Philip, S.Y., Rong, Y., et al.: Fedgraphnn: A federated learning benchmark system for graph neural networks. In: ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML) (2021)
24. Holzinger, A.: On topological data mining. In: Interactive Knowledge Discovery and Data Mining in Biomedical Informatics, Lecture Notes in Computer Science LNCS 8401, pp. 331–356. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43968-5_19
25. Holzinger, A.: Interactive machine learning for health informatics: When do we need the human-in-the-loop? Brain Informatics **3**(2), 119–131 (2016). https://doi.org/10.1007/s40708-016-0042-6
26. Holzinger, A.: The next frontier: Ai we can really trust. In: Kamp, M. (ed.) Proceedings of the ECML PKDD 2021, CCIS 1524, pp. 427–440. Springer Nature (2021). https://doi.org/10.1007/978-3-030-93736-2_33

27. Holzinger, A., Dehmer, M., Emmert-Streib, F., Cucchiara, R., Augenstein, I., Del Ser, J., Samek, W., Jurisica, I., Díaz-Rodríguez, N.: Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence. Information Fusion **79**(3), 263–278 (2022). https://doi.org/10.1016/j.inffus.2021.10.007
28. Holzinger, A., Haibe-Kains, B., Jurisica, I.: Why imaging data alone is not enough: Ai-based integration of imaging, omics, and clinical data. European Journal of Nuclear Medicine and Molecular Imaging **46**(13), 2722–2730 (2019). https://doi.org/10.1007/s00259-019-04382-9
29. Holzinger, A., Malle, B., Saranti, A., Pfeifer, B.: Towards multi-modal causability with graph neural networks enabling information fusion for explainable ai. Information Fusion **71**(7), 28–37 (2021). https://doi.org/10.1016/j.inffus.2021.01.008
30. Holzinger, A., Müller, H.: Toward human-AI interfaces to support explainability and causability in medical ai. IEEE COMPUTER **54**(10), 78–86 (2021). https://doi.org/10.1109/MC.2021.3092610
31. 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: Springer Lecture Notes in Computer Science LNCS 9817, pp. 81–95. Springer, Heidelberg, Berlin, New York (2016). https://doi.org/10.1007/978-3-319-45507-56
32. Holzinger, A., Plass, M., Kickmeier-Rust, M., Holzinger, K., Crişan, G.C., Pintea, C.M., Palade, V.: Interactive machine learning: experimental evidence for the human in the algorithmic loop. Applied Intelligence **49**(7), 2401–2414 (2019). https://doi.org/10.1007/s10489-018-1361-5
33. Hu, Y., Niu, D., Yang, J., Zhou, S.: Stochastic distributed optimization for machine learning from decentralized features. arXiv:1812.06415 pp. 1–10 (2018)
34. Huang, Y., Gupta, S., Song, Z., Li, K., Arora, S.: Evaluating gradient inversion attacks and defenses in federated learning. Advances in Neural Information Processing Systems **34**, 7232–7241 (2021)
35. 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. Knowledge Based Systems **220**, 106916 (2021). https://doi.org/10.1016/j.knosys.2021.106916
36. Jeanquartier, F., Jean-Quartier, C., Holzinger, A.: Integrated web visualizations for protein-protein interaction databases. BMC Bioinformatics **16**(1), 195 (2015). https://doi.org/10.1186/s12859-015-0615-z
37. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21 (2021). https://doi.org/10.1109/TNNLS.2021.3070843
38. Karimi, A.H., von Kügelgen, J., Schölkopf, B., Valera, I.: Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. arXiv:2006.06831 (2020)
39. Kazhdan, D., Dimanov, B., Magister, L.C., Barbiero, P., Jamnik, M., Lio, P.: Gci: A (g) raph (c) oncept (i) nterpretation framework. arXiv preprint arXiv:2302.04899 (2023)
40. 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
41. Landrum, M.J., Lee, J.M., Benson, M., Brown, G., Chao, C., Chitipiralla, S., Gu, B., Hart, J., Hoffman, D., Hoover, J.: Clinvar: public archive of interpretations of clinically relevant variants. Nucleic acids research **44**(D1), D862–D868 (2016). https://doi.org/10.1093/nar/gkv1222

42. Lapuschkin, S., Binder, A., Montavon, G., Müller, K.R., Samek, W.: The lrp toolbox for artificial neural networks. The Journal of Machine Learning Research (JMLR) **17**(1), 3938–3942 (2016)
43. Lazareva, O., Baumbach, J., List, M., Blumenthal, D.B.: On the limits of active module identification. Briefings in Bioinformatics **22**(5), bbab066 (2021)
44. 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
45. Liu, R., Yu, H.: Federated graph neural networks: Overview, techniques and challenges. arXiv preprint arXiv:2202.07256 (2022)
46. Lucic, A., ter Hoeve, M., Tolomei, G., de Rijke, M., Silvestri, F.: Cf-gnnexplainer: Counterfactual explanations for graph neural networks. arXiv:2102.03322 (2021)
47. Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., Zhang, X.: Parameterized explainer for graph neural network. Advances in neural information processing systems **33**, 19620–19631 (2020)
48. MacKay, D.J., Mac Kay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)
49. Magister, L.C., Barbiero, P., Kazhdan, D., Siciliano, F., Ciravegna, G., Silvestri, F., Liò, P., Jamnik, M.: Encoding Concepts in Graph Neural Networks. arXiv e-prints pp. arXiv–2207 (2022)
50. 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)
51. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv:1912.03277 (2019)
52. Malle, B., Giuliani, N., Kieseberg, P., Holzinger, A.: The more the merrier - federated learning from local sphere recommendations. In: Machine Learning and Knowledge Extraction, Lecture Notes in Computer Science LNCS 10410, pp. 367–374. Springer (2017). https://doi.org/10.1007/978-3-319-66808-6_24
53. Manhaeve, R., Dumančić, S., Kimmig, A., Demeester, T., De Raedt, L.: Deepproblog: Neural probabilistic logic programming. arXiv:1805.10872 (2018)
54. Matschinske, J., Späth, J., Nasirigerdeh, R., Torkzadehmahani, R., Hartebrodt, A., Orbán, B., Fejér, S., Zolotareva, O., Bakhtiari, M., Bihari, B., Bloice, M., Donner, N.C., Fdhila, W., Frisch, T., Hauschild, A.C., Heider, D., Holzinger, A., Hötzendorfer, W., Hospes, J., Kacprowski, T., Kastelitz, M., List, M., Mayer, R., Moga, M., Müller, H., Pustozerova, A., Röttger, R., Saranti, A., Schmidt, H.H., Tschohl, C., Wenke, N.K., Baumbach, J.: The featurecloud ai store for federated learning in biomedicine and beyond. arXiv:2105.05734 (2021). https://doi.org/10.48550/arXiv.2105.05734
55. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
56. 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 Biotechnology **70**, 67–72 (2022). https://doi.org/10.1016/j.nbt.2022.05.002
57. 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

58. Ortega, A., Frossard, P., Kovačević, J., Moura, J.M., Vandergheynst, P.: Graph signal processing: Overview, challenges, and applications. Proceedings of the IEEE **106**(5), 808–828 (2018)
59. Pfeifer, B., Baniecki, H., Saranti, A., Biecek, P., Holzinger, A.: Multi-omics disease module detection with an explainable greedy decision forest. Scientific reports **12**(1), 1–15 (2022). https://doi.org/10.1038/s41598-022-21417-8
60. Pfeifer, B., Chereda, H., Martin, R., Saranti, A., Angerschmid, A., Clemens, S., Hauschild, A.C., Beissbarth, T., Holzinger, A., Heider, D.: Ensemble-gnn: federated ensemble learning with graph neural networks for disease module discovery and classification. bioRxiv p. 2023.03.22.533772 (2023). https://doi.org/10.1101/2023.03.22.533772
61. Pfeifer, B., Holzinger, A., Schimek, M.G.: Robust random forest-based all-relevant feature ranks for trustworthy ai. Studies in Health Technology and Informatics **294**, 137–138 (2022). https://doi.org/10.3233/SHTI220418
62. Pfeifer, B., Saranti, A., Holzinger, A.: Network module detection from multi-modal node features with a greedy decision forest for actionable explainable ai. arXiv preprint arXiv:2108.11674 (2021)
63. 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
64. Saranti, A., Hudec, M., Minarikova, E., Takac, Z., Großschedl, U., Koch, C., Pfeifer, B., Angerschmid, A., Holzinger, A.: Actionable explainable ai (axai): a practical example with aggregation functions for adaptive classification and textual explanations for interpretable machine learning. Machine Learning and Knowledge Extraction **4**(4), 924–953 (2022). https://doi.org/10.3390/make4040047
65. Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K.R., Montavon, G.: Higher-order explanations of graph neural networks via relevant walks. arXiv preprint arXiv:2006.03589 (2020)
66. Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K.R., Montavon, G.: Xai for graphs: Explaining graph neural network predictions by identifying relevant walks. arXiv:2006.03589 (2020)
67. Schramowski, P., Stammer, W., Teso, S., Brugger, A., Herbert, F., Shao, X., Luigs, H.G., Mahlein, A.K., Kersting, K.: Making deep neural networks right for the right scientific reasons by interacting with their explanations. Nature Machine Intelligence **2**(8), 476–486 (2020). https://doi.org/10.1038/s42256-020-0212-3
68. Singh, R., Dourish, P., Howe, P., Miller, T., Sonenberg, L., Velloso, E., Vetere, F.: Directive explanations for actionable explainability in machine learning applications. arXiv:2102.02671 (2021)
69. Staab, S., Studer, R.: Handbook on ontologies. Springer Science and Business Media, Heidelberg (2010)
70. Stammer, W., Schramowski, P., Kersting, K.: Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3619–3629 (2021)
71. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
72. Szklarczyk, D., Morris, J.H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N.T., Roth, A., Bork, P.: The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic acids research **45**(D1), D362–D368 (2016). https://doi.org/10.1093/nar/gkw937

73. Teso, S., Kersting, K.: Explanatory interactive machine learning. In: AIES19 Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. AAAI (2019). https://doi.org/10.1145/3306618.3314293
74. Veličković, P.: Everything is connected: Graph neural networks. Current Opinion in Structural Biology **79**, 102538 (2023). https://doi.org/10.1016/j.sbi.2023.102538
75. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21 (2020). https://doi.org/10.1109/TNNLS.2020.2978386
76. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32** (2019)
77. Zhang, R., Guo, S., Wang, J., Xie, X., Tao, D.: A survey on gradient inversion: Attacks, defenses and future directions. arXiv preprint arXiv:2206.07284 (2022)