Graph Learning in Machine-Readable Plant Topology Data

Jonas Oeing*, Kevin Brandt, Michael Wiedau, Gregor Tolksdorf, Wolfgang Welscher, and Norbert Kockmann

DOI: 10.1002/cite.202200223

Supporting Information available online

Digitalization shows that data and its exchange are indispensable for a versatile and sustainable process industry. There must be a shift from a document-oriented to a data-oriented process industry. Standards for the harmonization of data structures play an essential role in this change. In engineering, DEXPI (Data Exchange in the Process Industry) is already a well-developed, machine-readable data standard for describing piping and instrumentation diagrams (P&ID). In this publication, industry, software vendors, and research institutions have joined forces to demonstrate the current developments and potentials of machine-readable P&IDs in the DEXPI format combined with artificial intelligence. The aim is to use graph neural networks to learn patterns in machine-readable P&ID data, which results in the efficient engineering and development of new P&IDs.

Keywords: Artificial intelligence, Data management, DEXPI, Graph neural networks, Piping & instrumentation diagram, Process industry

Received: December 15, 2022; *revised:* March 13, 2023; *accepted:* April 24, 2023

1 Introduction

Machine-readable plant topologies can lead to different improvements in existing work processes in the process industry and even new use cases. The following gives a brief overview and examples of improvement clusters for using plant topology to automate engineering processes, maintenance, and process optimization.

1.1 Automation of Engineering Processes

In former decades, engineering processes of chemical plants were mainly driven by paperwork. Data sheets, piping and instrumentation diagrams (P&IDs), and other related documents played and still play an important role. A big challenge was keeping track of consistency between all documents, particularly concerning procurement processes. The main tasks were creating lists of apparatus and machines, lists of piping components, counting nozzles, and many more activities. With modern computer-aided engineering (CAE) systems, including machine-readable P&IDs, extracting such lists, counting elements, etc., is highly automated. A higher degree of digitalization led to a higher degree of automation in engineering processes and the need for more automation in the chemical plant itself. This results in more and more sensors and digital instruments in the plants, which makes the engineering process even more voluminous, extensive, and complex. This growing degree of digitalization of future plants leads to an even higher need for integrated engineering processes, where machine-readable plant topologies play a crucial role.

[1]Jonas Oeing https://orcid.org/0000-0002-8410-7651 (jonas.oeing@tu-dortmund.de), [1]Kevin Brandt, [2]Michael Wiedau https://orcid.org/0000-0001-7153-7292, [2]Dr. Gregor Tolksdorf, [3]Wolfgang Welscher, [1]Prof. Dr.-Ing. Norbert Kockmann

[1]Department of Biochemical and Chemical Engineering, Laboratory of Equipment Design, TU Dortmund University, Emil-Figge-Straße 68, , 44227 Dortmund, Germany.

[2]X-Visual Technologies GmbH, James-Franck-Straße 15, 12489 Berlin, Germany.

[3]Evonik Operations GmbH, Paul-Baumann-Straße 1, 45128 Marl, Germany.

1.2 Improvement of Plant Maintenance, Service and Safety

Similar to the former described engineering process, these technology principles are also important when operating an existing plant. Typically, an existing chemical plant needs digital technologies to improve plant operation and maintenance processes. Machine-readable plant topologies can automatically generate lists for maintenance procedures and their specific tasks. For digital work permits of engineering and maintenance contractors, the usage of machine-readable P&IDs makes sense to give the contractors a preview of the location where the specific task has to be performed (e.g., for the repair of a specific pump). When it comes to getting an overview of the performance of a currently running plant, Fig. 1 gives an impression of how machine-readable plant topology can be used inside an Industrial Internet of Things (IIoT) platform to connect plant topology and process data for various future applications in the operation of a chemical plant.

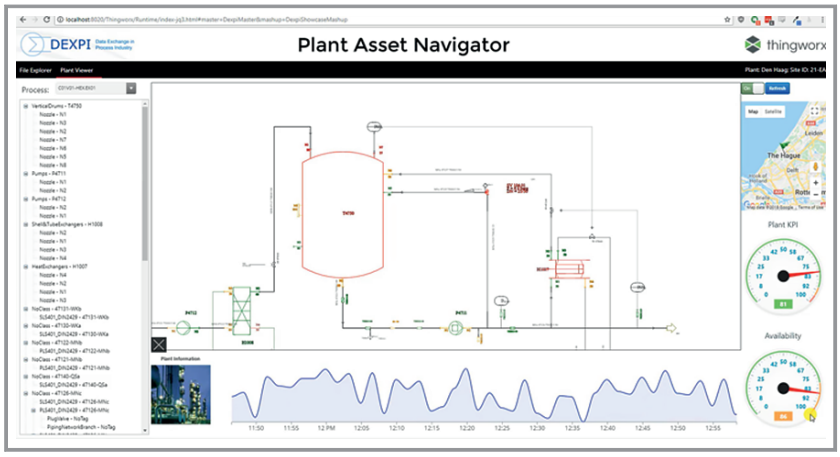

Figure 1. Web-based usage of plant topology of a storage vessel in an Industrial Internet of Things (IIoT) platform. Here: usage of a DEXPI plant topology in PTC ThingWorx.

Another important approach is using digital descriptions of assets in the context of safety considerations. Lee et al. [1] show that the digital twin serves as an essential basis for optimized process control, and those process safety accidents can be reduced. The authors conclude that for process safety considerations, approaches must be developed that use data that unifies different aspects: frameworks for describing structures, plant descriptions, and employees' work processes [2].

1.3 Process Optimization

Using plant data and the connection with the plant topology can also improve the optimization of chemical processes. In the last years/decades, when people started to optimize plants with computer-based optimization methodologies, the topology information of the plant was not available due to the earlier described reasons. Now with this information available, it can additionally be used to optimize chemical processes. For example, the knowledge about the (geometric) location of a measurement tag assists in rapidly repairing the related sensor. The repeated failure of a device represents another exemplary use case: Can measurement data anomalies from neighboring measuring points assist the diagnosis? This knowledge about neighboring can only be derived from machine-readable plant topologies.

1.4 Quality Improvements in Machine-Readable Plant Data

Graph neural networks (GNN) are used to learn patterns in machine-readable P&IDs and do consistency checks in P&IDs from their structure information to improve the quality of machine-readable plant topologies. In this way, the deep learning (DL) models store the complexity of process plant topologies. Therefore, in this work, it is assumed that by supporting the drawing process of P&IDs through AI, time and costs can be saved as errors are minimized. The consistency check is intended to examine three main issues in particular:

- Are components corresponding to the drawn P&ID present at positions that make sense from a process engineering point of view, or are essential components missing?
- Are components completely linked with each other?
- Do additional links, e.g., signal lines or pipelines, such as recycles, have to be added to the P&ID?

Fig. 2 shows that, especially in an early planning phase, there are many degrees of freedom and at the same time low modification costs. For this reason, AI-supported process development can bring significant added value, especially in the early engineering phases of conceptual design and basic planning activities.

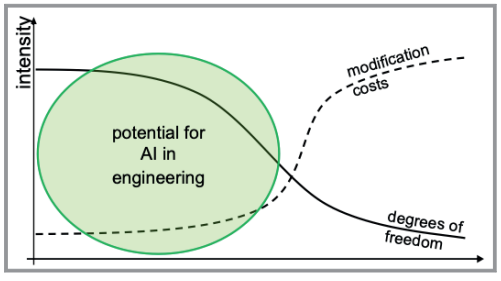

Figure 2. Potential of AI in early engineering phases regarding modification costs and number of degrees of freedom.

1.5 Global Challenges

Overall, various use cases exist in the industry for machine-readable plant topologies. As this is such an important technology, the accuracy and correctness of the plant topology are crucial for these use cases. Concerning global challenges in climate change, circular economy, global pandemics, etc., fast and flexible adoption of production processes is strongly demanded and leads to faster changes in process plant development [3]. At the same time, high efficiency and accuracy in process flow diagrams are necessary to accelerate the process of building process plants. Furthermore, the demographic change in our society leads to fewer experienced process engineers and raises the high demand for digitalization in chemical plant development, construction, building, and maintenance. Due to this global development, there is a high demand to improve the quality of process flow diagrams.

2 State of the Art

2.1 Standards for Machine-Readable Plant Data

When discussing plant data, we mean data around plant items: the apparatuses, machines, pipes, piping components, sensors and actuators, and their designed connections. Looking at the entire asset lifecycle (or plant lifecycle), this structural information and master data neither belong to the process structure consisting of functional unit operations and streams nor to actual measurements or maintenance procedures at physical equipment. Instead, we are discussing the asset lifecycle's functional design and asset specification aspects, see [4]. In this phase, basic engineering takes place, transforming the functional requirements coming from the process to a design of the plant that will be able to realize this process.

The topology of a plant (i.e., the network structure of the connections between plant items) is mainly determined by its included apparatuses and machines (as nodes) on the one hand and piping elements connecting the former (as edges) on the other hand. The topology of a process plant for continuous, stationary processes is of higher importance than the topology of a batch process, with its focus on dynamic behavior taking place in a limited number of vessels according to process recipes consisting of several sequential steps (simplified view for the sake of brevity). For conventional continuous processes, the plant topology can be assumed fixed from the moment of build-up and commissioning. This topology encodes more information about the production process than in batch processes. It can be expected to derive more valuable information from it for operation, performance analysis, and optimization. The most crucial point is that the topology must be available in a machine-interpretable way to make use of it systematically and automatically. In this contribution, we want to put focus on the topology part of plant data, letting aside any graphical aspects of a P&ID and reducing the engineering data aspect to the identification (i.e., name) and classification (i.e., type of technical object) of the plant items to have the minimum viable amount of information to give a meaning to the nodes of the topology's graph representation.

We refer to our previous publication for a general overview of the availability of (digital) standards for the process industry supporting machine learning (ML), reasoning, or AI in general [5]. We will now have a closer look at available and upcoming digital standards, especially for data regarding the topology of process plants and the classification of plant items from a process-domain-focused perspective (opposed to a purely abstract computer science or math-based point of view). To highlight some key challenges experienced in applying ML in chemical engineering, see [6].

Starting with the topology aspect, the IEC/ISO 81346 part 1 defines a mechanism for decomposing a whole facility or a single object into meaningful parts. Based on this methodology, we can first separate the purely functional process parts from the parts of a facility that represent the means (plant items) to realize the functional requirements of the process. The decomposition itself does not give the plant topology, but it helps to name the plant items that are the members of the plant topology. The second standard in the process industry of relevance for the topology is the DEXPI P&ID specification [7]. As this specification provides a way to digitally describe and transport P&IDs, it naturally includes a concept to connect the different technical objects systematically and machine-interpretably. This network of technical objects can be modeled as a graph, and respective graph methods can be applied. A paper from 2021 gives examples of the possibilities of navigation through process plant graphs when having the topological information available, e.g., in a machine-readable Data Exchange in the Process Industry (DEXPI) file [8].

The second aspect, the classification of plant items, is addressed by the ISO 15926 standards, mainly by parts 2 (data model) and 4 (reference data library). The reference data library (RDL) of ISO 15926 [9] is the basis for the CFIHOS RDL and the DEXPI RDL. In addition, the IEC CDD (common data dictionary) operates an RDL for process automation (IEC 61987 [10]) and general items/electric components (IEC 61360 [11]) that are of relevance to the objects that may appear on P&IDs. When plant data is given, the technical objects should be classified by terms provided in one of these international RDLs, to give the topology nodes a reproducible, widely understood meaning. An upcoming, future standard of interest for the application of AI in the process industry is part of the ISO 15926 ecosystem. So far, a technical report exists for a top-level industrial ontology that could be further developed into an official release as part 14 of ISO 15926 [12].

Using these standards for machine-readable plant data is valuable for AI's symbolic and statistical branches. The

symbolic branch of AI uses, e.g., semantic technologies and knowledge graphs to generate new knowledge by semantic reasoning. Standards like ISO 15926-14 are being designed to enable exactly this semantic reasoning (using OWL 2 Direct Semantics) for the process industry's use cases, proving to be crucial for managing the complexity of domains and disciplines [12]. The statistical branch of AI (machine learning that is creating knowledge based on (lots of) data), can make use of these standards by having a tool-independent format and structure (i.e., ontology) to represent the results of the model-building via ML. A recent synthesis of some of the standards mentioned above into a framework for zero-defect engineering is given by [13].

The topology of a process plant, as depicted on a P&ID, is similar but not necessarily identical to the topology of the underlying process that can be seen on a process flow diagram (PFD). We will have a closer look at this aspect in the following subsection, where we will describe the way DEXPI represents topology in more detail.

2.2 DEXPI Plant Topology

The topology of a process plant, as modeled in a P&ID according to DEXPI, is mainly represented by a series of equipment (with nozzles) and piping components. Nozzles, as well as piping components, are owners of so-called PipingNodes. PipingNodes are abstract elements that serve as "ports" where PipingConnections can be plugged in, directly connecting two PipingNodes. In the conceptual view of a graph, the PipingConnections serve as edges, the PipingComponents (valves, fittings, etc.) and equipment (with their nozzles) serve as nodes.

In a currently ongoing project that started in 2022, the DEXPI group is developing an extension of the existing P&ID specification that aims at describing PFD and block flow diagram (BFD) as well. These two types of diagrams (PFD, BFD) describe the purely functional process requirements, i.e., process steps (also called unit operations, depending on the level of detail) and streams (of material, energy, information). This is in contrast to what is shown on a P&ID that contains the means of performing these process functions (e.g., centrifugal pump (P&ID) performs pumping (PFD), mixer (P&ID) performs mixing (PFD), etc.). Of course, these elements of the different worlds (process vs. plant) are related but not the same; see [13] for more detailed elaboration.

In order to represent the new elements necessary to describe PFDs and BFDs, the extended DEXPI specification will include a new package for classes of processes and streams. Having all concepts in one specification theoretically allows for "two-layer" models, combining the information of PFD and P&ID in one knowledge graph. Regarding the topology, new types of connections need to be introduced, e.g., a ProcessStreamNode as a port for unit operations in analogy to the PipingNode for piping components and nozzles. The only allowed relation for two elements of these different worlds would be a relation of type "realization"; it is, e.g., not allowed to connect a unit operation with a piping component via PipingNode or ProcessNode.

2.3 Graph Learning on Machine-Readable P&ID Data

In the last years, the application of AI increased in the field of machine-readable plant topologies. At least since the introduction of the digital twin [14] and the potential it offers by connecting AI, most stakeholders in the process industry agree that this can save costs, time, and engineering effort [15]. However, even before that, there were initial approaches that dealt with the analysis of topologies in the process industry. In particular, the topology analysis using the POOM approach developed by H. Gabbar can be mentioned. The plant topology is divided into several areas, which can be mapped to plant operation levels described by ISA-S88 using an intelligent algorithm [16]. Other approaches already use artificial intelligence algorithms to predict the topology of plants. However, these mostly follow a sequential approach [17–19]. Thus, Vogel et al. and Hirtreiter et al. use strings in the SFILES 2.0 [20] notation to describe the topology of PFD, including control structure [17, 18] as strings. The PFDs in the SFILES 2.0 notation are further processed using NLP (natural language processing) transformer models. These allow either the prediction of follow-up process units or additional control structures. As a preliminary work, Oeing et al. [19] show, in addition to the sequential processing of plant topologies, especially the modeling of plant topologies as graphs. In this work, standardized DEXPI P&IDs were also already used as a data basis. First feasibility analyses showed the potential to process P&ID data interpreted as graphs using P&ID to classify components [19].

Besides the process industry, especially the manufacturing industry is interested in using graph learning in graph-structured data, which, in contrast, does not focus on P&IDs. For example, Mortlock et al. show a futuristic approach to how graph learning methods can influence complex manufacturing decisions and enable more autonomy in the future. The approach uses a query-based graph learning framework to facilitate cognition in digital twins [21].

3 Graph Learning in P&IDs using GNNs

The hypothesis investigated in this paper assumes that graph learning can detect anomalies in P&ID, which consequently results in quality improvement of the drawing process of P&IDs. For this reason, GNNs can learn the topology of process plants. Comparing a machine-readable P&ID with the information learned by the GNN made

predictions as to whether the P&ID shows inconsistencies regarding its structure. In the following, the methods used, their implementation and results are explained in more detail.

3.1 Methods

In recent years, graph learning algorithms have become increasingly important in addition to the classical AI algorithms in image processing. These allow the convolution of graph-based data and can learn the relationship of interconnected data in this way. Preliminary work has confirmed the initial feasibility of using P&ID data in GNNs [19]. Within the scope of this work, quantitative results are produced based on this to ensure the application of consistency checks. In addition, the results are implemented in P&ID software from the vendor X-Visual Technologies GmbH, Berlin, Germany.

3.1.1 Graph Neural Networks

Graphs G are defined as a set of nodes V and edges E: $G = \{V,E\}$. Nodes interpret different objects, and edges represent the relations between these objects. Defined are the edges as an ordered pair of nodes: $(u,v) \in E$ with starting point $u \in V$ and endpoint $v \in V$ [22]. P&IDs are also mathematically equivalent to a graph in which the components describe the nodes, and the pipelines or signal lines represent the edges, which makes it possible to apply GNNs to P&IDs [19].

GNNs are artificial networks that can learn graph structures using neighborhood aggregation. A mathematical description of the idea of GNNs is represented by Eq. (1) and (2). The information that a node gets from its neighbors is aggregated with an aggregation function and processed with an update function to the new state of the node. The update function comprises a weighting matrix and a non-linear activation function [23, 24]. The aggregation function's result is also called message mN(u).

The computation of the state h (also embedding) of a node u is iterative, so the state in iteration k+1 depends only on the own state of iteration $k\left(h_u^{(k)}\right)$ and the states of the neighbors in iteration $k\left(h_v^{(k)}\right)$.

$$h_u^{(k+1)} = UPDATE^{(k)}\left(h_u^{(k)},\ AGGREGATE^{(k)}\left(\left\{h_v^{(k)} \forall v \in N(u)\right\}\right)\right) \tag{1}$$

$$h_u^{(k+1)} = UPDATE^{(k)}\left(h_u^{(k)},\ m_{N(u)}^{(k)}\right) \tag{2}$$

The iterations defined in this way can also be interpreted as layers. The aggregation and update functions are also marked with the upper index k since they are the same for all nodes within a layer but can differ in the weightings or entirely in the different layers. As a state of the nodes in layer 0, the properties (features) X of the nodes are usually used. For example, the eigenvector centrality or other metrics regarding the node's structure can be used if the nodes do not have any features. With each additional layer, the nodes also reach information from a step further away. Fig. 3 shows the *neighborhood aggregation* for an example graph for a GNN with two layers: To compute the state of node 5 in layer 2, information is aggregated from nodes 2 and 3 in layer 1, and for the states, in layer one, the respective properties of the predecessors in layer 0 are aggregated. The so-called computation graph of the GNN shown in Fig. 3 on the right represents a tree structure created by unfolding the neighborhood of the target node (here, node 5) [23, 24]. The aggregation and update functions used in this work are shown in detail in the Supporting Information.

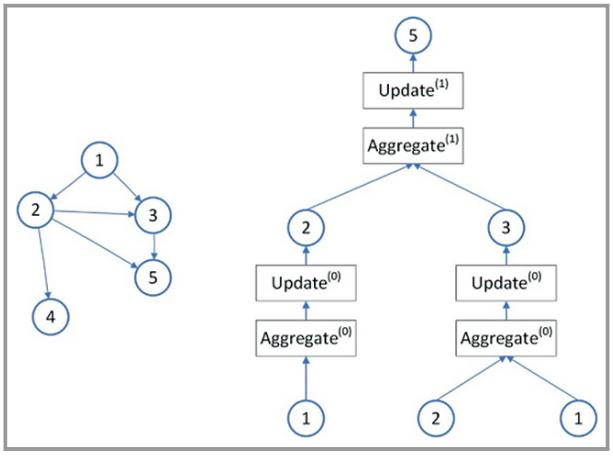

Figure 3. Message passing example of a GNN with two layers and the target node 5. Left: input graph, right: computational graph.

3.1.2 Jumping Knowledge Networks

The number of layers in a GNN has a significant impact: If the direct environment of the nodes is to be considered, a few layers are sufficient. However, if information from more distant nodes is also deemed, more layers must be used. In the case of using many layers, the problem may arise that a lot of information from more distant neighborhoods is aggregated, and the data from the immediate neighborhood of the node is lost. On the other hand, if only a few layers are used, the information from the more distant nodes will be missing. For example, with two layers, a pump only receives data from the directly adjacent pipe tees and valves, and classification of whether a pump is useful or not at this point in the P&ID cannot be made as a result. One way to avoid these two problems is to introduce jump connections. Here the states of the nodes are forwarded to the next layer on the one hand, but on the other hand, they are stored and used again at the end. This idea comes from Xu et al. [25], who refer to this type of GNNs as jumping knowledge

networks (JKN). The structure of a JKN is shown in Fig. 4. There are several approaches to how the information can end up being used. For example, max-pooling can filter out the most relevant information for each node from each layer's states. This way, different layers can be considered important for each node respectively. Another possibility is to concatenate all states of a node and multiply them by a weighting matrix. This also leads to the fact that different layers can have a decisive influence on the output state of a node. Since the same weighting matrix is used for all nodes, the adjustment to individual nodes is not relatively as high, but the layers that have the most significant influence can be weighted the most.

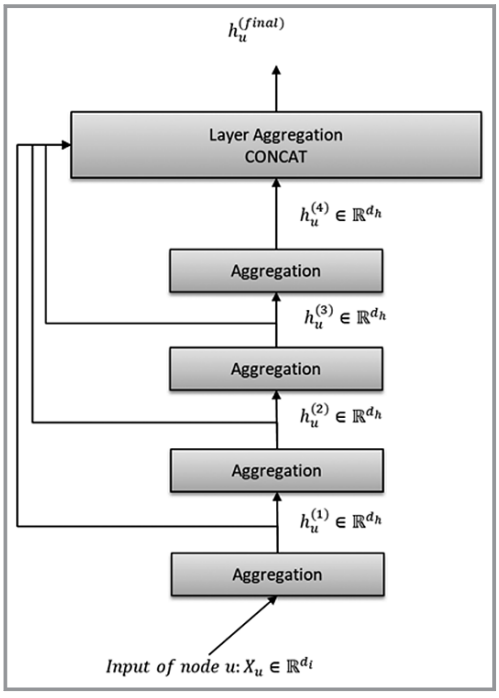

Figure 4. Structure of a 4-layer jumping knowledge network.

3.2 Dataset

The dataset used in this work is a set of machine-readable DEXPI P&IDs. In total, 25 P&IDs were used, mainly consisting of desorption-absorption modules, distillation plants, and modular laboratory plants. A large part of these P&IDs is available as an open-source dataset provided by the KEEN project [26]. To maintain good accessibility of the data, the DEXPI P&IDs were converted into a GraphML format [19] using a DEXPI2GraphML-converter [27, 28], which is available as a python and windows application. The dataset contains 2504 components and 2842 connections (piping or signal lines), which are increased to 5647 connections for the case of an undirected representation. For improved training, the component classes that exist in the P&IDs are subdivided into the following classes based on their technical process function (see Tab. 1).

Table 1. Categories of P&ID equipment regarding its technical functionality including their number in the data set.

Category	Number in the data set
Valves	725
Ball valves	74
Control valves	36
Check valves	47
Safety valves	68
Pumps	60
Heat exchangers	82
Separation equipment	30
Temperature sensors	171
Pressure sensors	115
Flow sensors	60
Other sensors	71
Vessels	56
Pipe tees	616
Piping components	67
Inlet/outlet	226

3.3 Implementation

In this work, GNNs are used to detect anomalies in P&IDs. A workflow is developed, which allows the AI-based consistency check to be applied directly to a drawn P&ID. Based on this, the developed workflow will also be implemented in the P&ID software of the software vendor X-Visual Technologies GmbH, Berlin, Germany.

3.3.1 Consistency Check – Workflow

In the following, the workflow (see Fig. 5) for consistency checking of P&IDs with the help of GNNs is explained in more detail. Therefore, a node classification is performed by a GNN. The basis is the data set presented in Sect. 3.2, of which 70 % of the components are used as training data. The P&ID graphs extract the features required for node classification (node class, node subclass). Subsequently, the randomly initialized models are trained with the training data (loss calculation and backpropagation). The training continues until the accuracy of the test data starts to drop, as this is an indicator for overfitting or 500 epochs are reached. The trained model will subsequently be used for consistency checking (validation). An unseen P&ID is used as a GraphML file. From this file, all components are separately passed into the model. The GNN aggregates the neighborhood for each component and compares it with the learned neighborhood. Based on this comparison, a

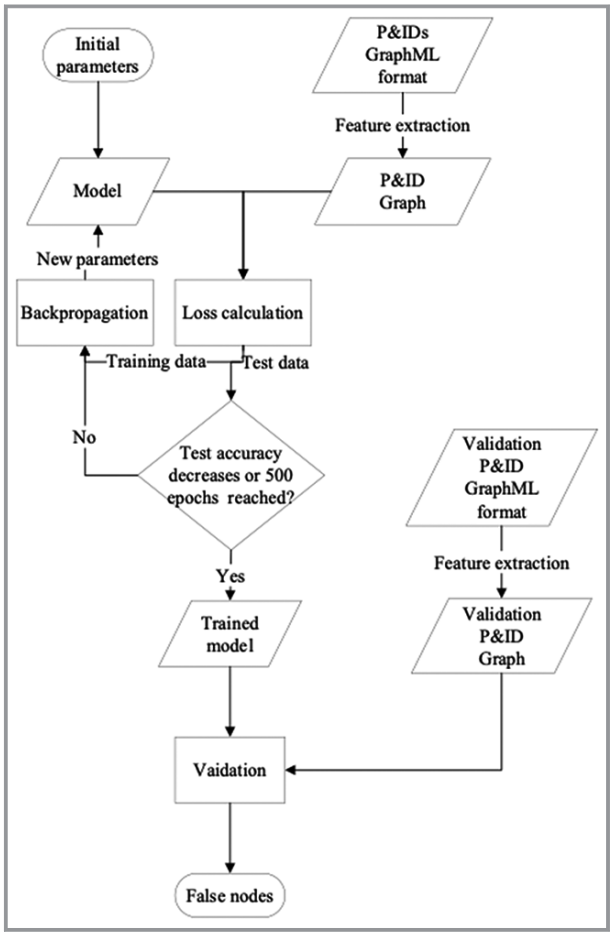

Figure 5. Workflow for AI-based consistency checks using GNNs.

decision is made whether a component is located at a meaningful position. If anomalies occur, the model returns the components involved in this anomaly (false nodes).

3.4 Results

In the following, different aggregation and update functions are used for node classification of P&ID equipment, and the results will be analyzed in detail. Subsequently, an investigation of additional model parameter and their influence on the performance of the node classification occurs. The metric used is the prediction accuracy for the test data. This indicates how many components within the test data set to be examined are assigned to a correct class by the respective model (see Eq. (3)).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3}$$

Fig. 6 shows the results in the form of radar charts. The test accuracies for different aggregation functions (*sum, mean, GCN, max-pooling, min-pooling, MLP*) are plotted against the number of layers (*2, 3, 4, 5, 6, 4-layer JKN, 6-layer JKN*). The diagrams describe different update functions (*ReLU, SeLU, tanh*). Each layer consists of 35 hidden neurons and has a self-loop, which allows aggregating information of the node of interest into the neighborhood. Based on a rule of thumb regarding Hagan et al., the number of hidden layers is calculated. The aim is to limit the number of free parameters in the model to adjust it to the degrees of freedom in the data set [29]. To get more information, the P&IDs have been processed as undirected graphs.

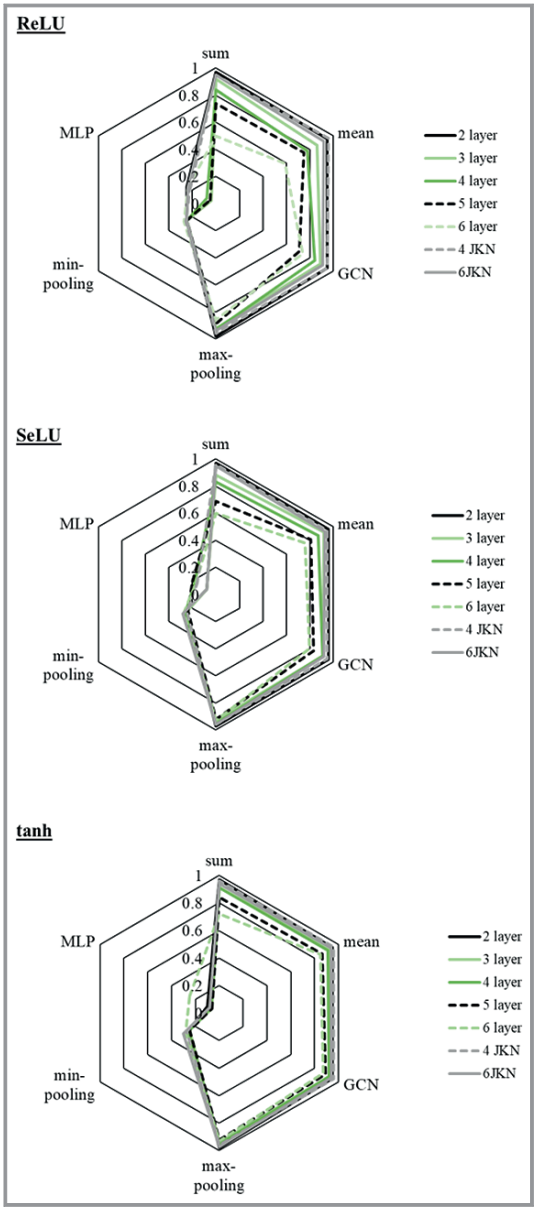

Figure 6. Prediction accuracy of the test data using different aggregation and update functions for different layer sizes including 4-layer and 6-layer JKNs.

The results for all update functions show the same trend. Thus, aggregation by *min-pooling* or *MLP* cannot aggregate

the neighborhood's behavior to the extent that the model can learn the structure of P&IDs. On the other hand, all model combinations can learn information in P&IDs. Thus, the *max-pooling* aggregator reliably provides good test accuracies of over 90 % for all update functions. In general, *tanh* offers the best results. For example, *tanh* shows higher accuracies (>72 %) for the sum, *mean, GCN*, and *max-pooling* aggregations, especially for GNN with a higher number of layers (*6 layers*), than ReLU (>60 %) or SeLU (>50 %). For *max-pooling* as well as for *mean* and *GCN*, all test accuracies are above 90 %. For *sum* aggregation, these drop off with an increasing number of layers. This is due to a possible „over-smoothing", where the large neighborhoods become too similar during summation, making the correct classification of components difficult. Further confirmation of this assumption is that the results for *2-layer* and *4-layers* are consistently high and generally decrease for increasing numbers of layers (e.g., *6-layer, sum, SeLU* with 60 %). Thus, the structure can be learned well for aggregating small neighborhoods, but these models cannot detect patterns of larger dimensions. For this reason, JKNs were also examined in this work. As expected, these give high results (92 % to 96 %) for all three update functions.

To evaluate the results, the confusion matrix for a 6-layer JKN with the tanh update function, *max-pooling* aggregation, and 35 hidden neurons per layer is shown in Fig. 7 for the aggregation of undirected graphs. Here, the results of the test data are displayed. The columns show the predicted P&ID components, while the rows represent the respective real P&ID components. Consequently, the matrix's main diagonal provides the number of correctly classified components. Almost all components are correctly classified. However, it is particularly noticeable that five pipe tees are falsely detected as valves. Since three-way valves and pipe tees have similar equipment types in their neighborhoods, this is not surprising and acceptable. In addition, just seven of 616 pipe tees are incorrectly detected, which is a very small error. The confusion matrix thus confirms the high prediction accuracy even with an uneven distribution of classes in P&IDs.

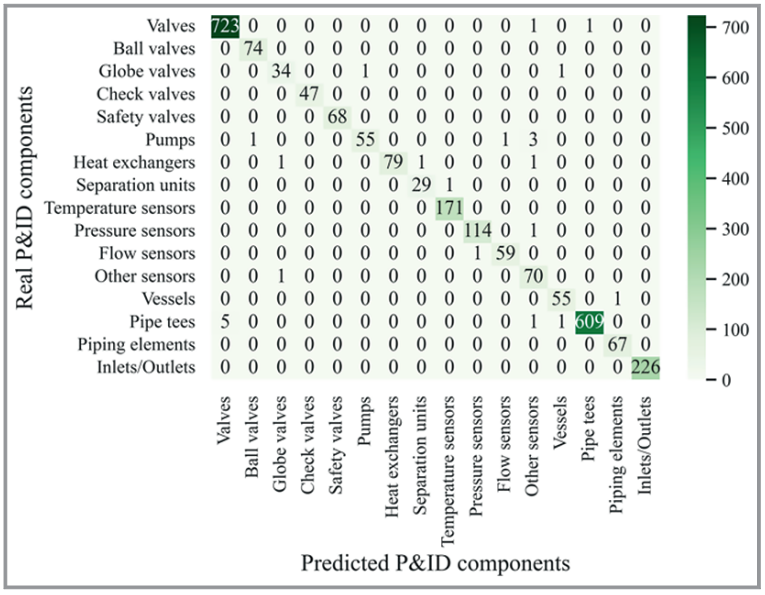

Figure 7. Confusion matrix of a 4-layer JKN with tanh update function and max-pooling aggregation for processing an undirected P&ID graph with 35 hidden neurons.

In the following, we investigate how other parameters affect the accuracy of the modeling. Based on the results from Fig. 6, *tanh* is used as the update and *max-pooling* as the aggregation, as these achieve the best results, up to 97 % prediction accuracy. Furthermore, different layer numbers and JKNs are investigated in more detail.

Number of Hidden Neurons

In this section, the length of the state vectors is examined for a node in the hidden layers (hidden neurons). In total, 16, 35, 50, and 100 neurons were investigated. The results are shown in Fig. 8a. No significant differences can be detected for all hidden neurons examined. In general, using as few hidden neurons as possible is recommended to prevent overfitting. Since 16 neurons provide slightly worse results, 35 neurons are recommended.

Directed vs. Undirected Graphs

The influence of the directed or undirected edges on the test accuracy of the GNNs is investigated in Fig. 8b. It shows that undirected graphs provide better test accuracy for all iteration depths except six layers. Especially for a low number of layers, the difference is large. This can be easily explained since more information is available when using undirected graphs since both the preceding and the nodes from the succeeding neighborhood are considered.

Node Degree

The influence of additional node features on the respective test accuracies is investigated to identify the most sensitive equipment features. Thus, it can be assumed that features of the graph structure significantly influence the training, which is why it can be helpful to consider information about these structures as input for the models. First, the degree of the nodes is used for this purpose. It is assumed that the GNN can improve the learning of structural information, such as the number of inputs and outputs of certain component classes, with the help of the degree. The results are shown in Fig. 8c. The degree of the nodes has no significant influence on the performance of the GNNs. Thus, the test accuracies of the JKNs and the 4-layer GNN remain unchanged at 95 % or higher, while those of the other GNNs show slightly worse results

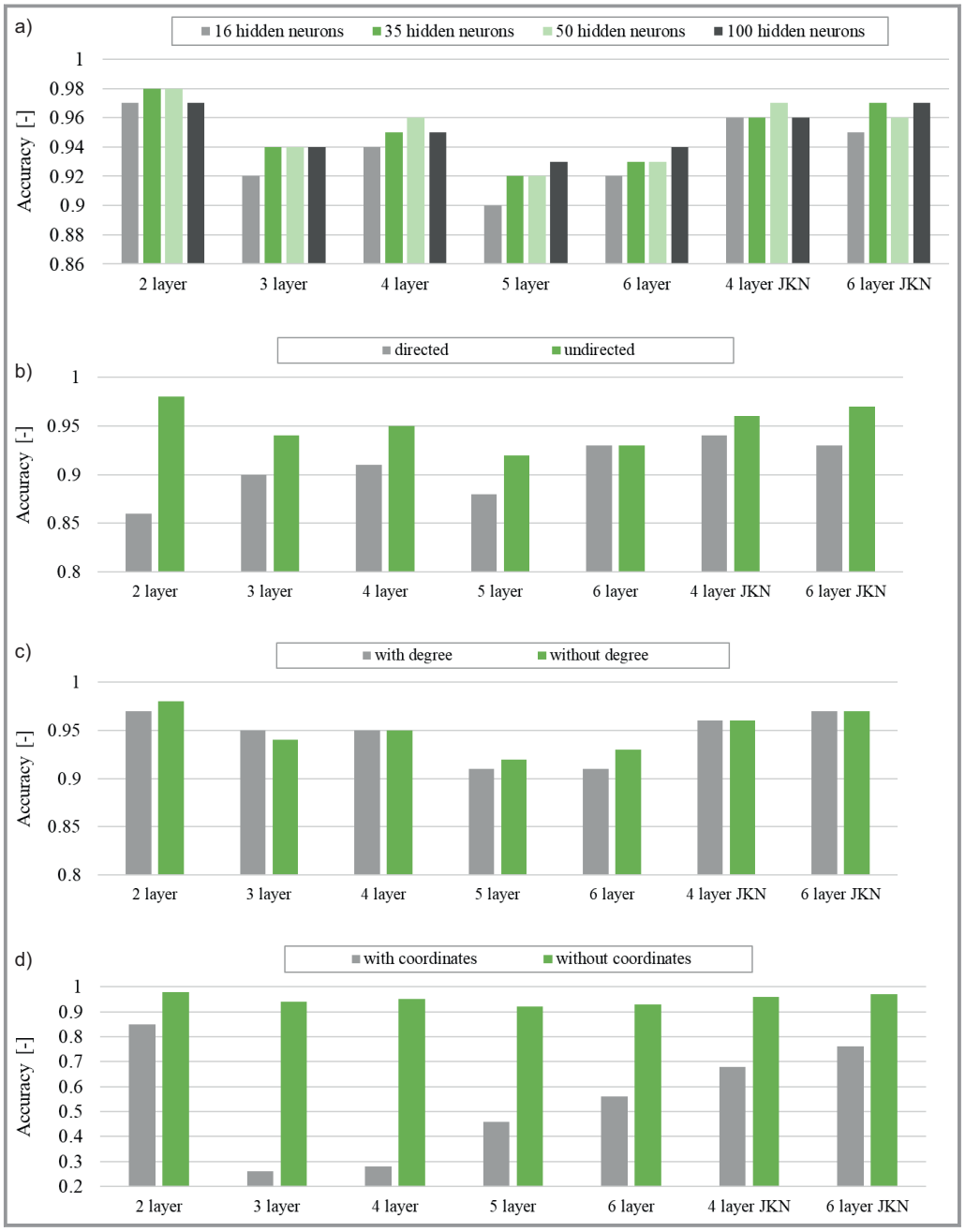

Figure 8. Prediction accuracy of GNN and JKN models with tanh update function depending on various parameters (graph direction, hidden neurons, node degree, and coordinates). a) Max-pooling aggregation for processing an undirected P&ID graph for different numbers of hidden layers; b) max-pooling aggregation and 35 hidden neurons for processing directed and undirected P&ID graphs; c) max-pooling aggregation and 35 hidden neurons for processing undirected P&ID graphs using node degree as an additional feature in aggregation; d) max-pooling aggregation and 35 hidden neurons for processing undirected P&ID graphs using coordinates of equipment as an additional feature in aggregation.

due to the different use of the degree except for the 3-layer GNN.

Coordinates of equipment

In addition to the degree, the coordinates of the components were also taken into account based on their position in the drawing (see Fig. 8d). The idea is that it is possible to incorporate other relationships into the training via the drawing position. The results show that using coordinates gives significantly worse outcomes than without location. For 3, 4, 5, and 6 layers, the GNN can no longer learn patterns in the data due to the coordinates. This indicates that

the position is not related to the typical structure of a chemical plant displayed in the P&ID. In addition, the max-pooling aggregator reacts sensitively to strong variations within the coordinates. For this reason, training using the coordinates is not possible.

In summary, previous results show that nearly all GNN of the investigated models, except for MLP or min-pooling aggregation, can perform consistency checks in machine-readable plant topologies. Accuracies of up to 98 % could be achieved with a 2-layer GNN and max-pooling aggregation. However, this network is unsuitable for learning large-scale patterns due to its low iteration depth of $k = 2$. For this reason, it is recommended to use a JKN (up to 96 % test accuracy) for an AI-based consistency check in P&IDs, which aggregates all iteration depths equally and thus avoids over-smoothing.

3.5 Application in PlantEngineer (P&ID software)

The model trained in Sect. 2.3 is integrated into the PlantEngineer P&ID software in cooperation with the software vendor X-Visual Technologies GmbH. The implementation is visualized in Fig. 9 on the left-hand side. The software interconnection is carried out via a docker container that contains the previously trained GNN model. The docker container accesses a GraphML P&ID, which is written into an Exchange folder from the PlantEngineer software on demand and reports the results of the AI-based consistency check into a .json-file that is used as a communication file. The consistency check results are visualized in PlantEngineer, with inconsistencies of components or mismatched links highlighted in red.

A corresponding example is shown in Fig. 9 on the right-hand side with an example of a storage vessel. The main advantages of containerization are good accessibility and the encapsulation of the model. For this reason, the user does not need any knowledge of AI or python installations. At the same time, the model can be easily replaced by swapping the container. Thus, future models can be adapted to the P&ID model depending on the requirements (pharmaceutical, petrochemical, or fine chemical plants, etc.). At the same time, a cloud solution was deliberately avoided to ensure the data security of the P&ID data through local processing.

4 Conclusion and Outlook

This paper highlighted the importance of machine-readable plant topologies and showed their potential for future process development. Thus, the current developments and opportunities for software vendors, process industry and research are outlined and a comprehensive approach for modeling machine-readable P&IDs is given. It was validated that graph learning methods, such as GNNs, can learn patterns in DEXPI-standardized P&IDs and that these patterns can be used as consistency checks to improve the drawing of new P&IDs (e.g., greenfield plants). However, the approach can be applied to all graph-structured, machine-readable plant topologies. JKNs could predict anomalies in P&IDs with up to 98 % accuracy during testing. Combined with the software plant engineer from vendor X-Visual Technologies GmbH it was possible to develop a prototype that enables real-time detection of inconsistencies (e.g., missing equipment, faulty connections). This reduces the development time in detail engineering due to a decrease in error rate and a resulting time saving. The implementation intentionally used local processing and deployment of the model via docker containerization to ensure data privacy.

In the future, it should be discussed whether using a cloud-based solution can further improve performance, as this would allow faster and more effective training of the model as the model has access to a more extensive database. In this way, cloud-based training could lead to robust models without a user-specific bias. Furthermore, in the future,

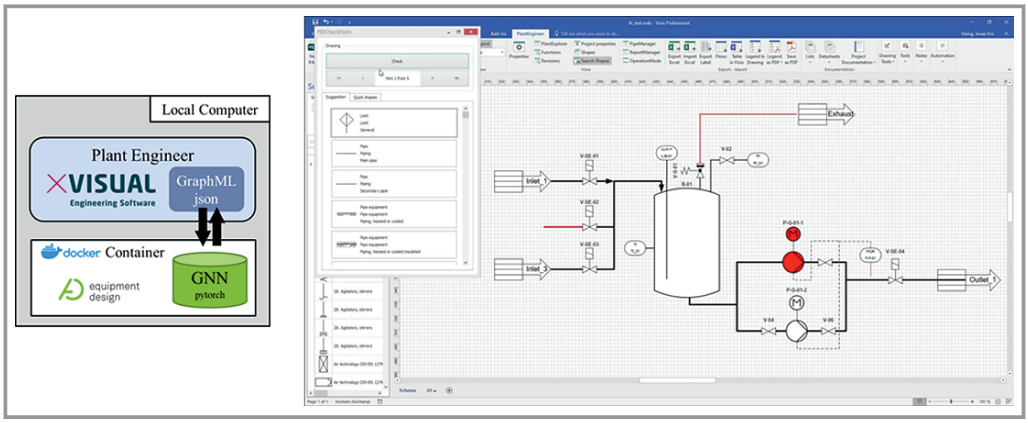

Figure 9. Implementation of the GNN in the P&ID software PlantEngineer from the software vendor X-Visual Technologies GmbH (left: implementation via docker container, right: visualization of the GNN results in the P&ID software PlantEngineer).

it makes sense to use other information, such as equipment data (e.g., material, design specifications, etc.), as additional input features and to investigate their impact on the AI-based consistency checking of P&IDs. However, this will require that detailed and fully machine-readable P&IDs of industrial process plants be available as training data. These must be available in large quantities and must have a high variance to ensure sufficiently robust modeling using GNNs.

Supporting Information

Supporting Information for this article can be found under DOI: https://doi.org/10.1002/cite.202200223. This section includes additional references to primary literature relevant for this research [30–34].

Acknowledgment

The research work has been developed within the KEEN project (support code: 01MK20014S). The authors thank the BMWK (Federal Ministry of Economic Affairs and Climate Action) for funding this project. Open access funding enabled and organized by Projekt DEAL.

Symbols used

d	[–]	degree of a node
E	[–]	Amount of edges
G	[–]	Graph
h	[–]	Embedding of a node
k	[–]	Iteration step
m	[–]	Message neighborhood aggregation
MLP	[–]	Multilayer perceptron
$N(u)$	[–]	Neighborhood of node u
W	[–]	Weighting matrix
X	[–]	Input features
α	[–]	Gating vector
σ	[–]	Non-linear activation function

Abbreviations

AI	Artificial Intelligence
BFD	Block Flow Diagram
CAE	Computer Aided Engineering
DEXPI	Data Exchange in the Process Industry
DL	Deep Learning
GCN	Graph Convolutional Network
GNN	Graph Neural Network
IIoT	Industrial Internet of Things
JKN	Jumping Knowledge Network
ML	Machine Learning
MLP	Multilayer Perceptron
P&ID	Piping and Instrumentation Diagram
PFD	Process Flow Diagram
RDL	Reference Data Library
ReLU	Rectified Linear Unit

References

[1] J. Lee, I. Cameron, M. Hassall, *Process Saf. Environ. Prot.* **2019**, *132*, 325–339. DOI: https://doi.org/10.1016/j.psep.2019.10.021
[2] J. Lee, I. Cameron, M. Hassall, *Digital Chem. Eng.* **2022**, *3*, 100017. DOI: https://doi.org/10.1016/j.dche.2022.100017
[3] *Prozessindustrie wird flexibler, Teil 1: Warum wird flexible Produktion gebraucht?*, CHEManager, February **2018**.
[4] M. Wiedau, L. von Wedel, H. Temmen, R. Welke, N. Papakonstantinou, *Chem. Ing. Tech.* **2019**, *91 (3)*, 240–255. DOI: https://doi.org/10.1002/cite.201800112
[5] M. Wiedau, G. Tolksdorf, J. Oeing, N. Kockmann, *Chem. Ing. Tech.* **2021**, *93 (12)*, 2105–2115. DOI: https://doi.org/10.1002/cite.202100203
[6] A. M. Schweidtmann, E. Esche, A. Fischer, M. Kloft, J. Repke, S. Sager, A. Mitsos, *Chem. Ing. Tech.* **2021**, *93 (12)*, 2029–2039. DOI: https://doi.org/10.1002/cite.202100083
[7] M. Theißen, M. Wiedau, *DEXPI – P&ID Specification*, **2021**. https://dexpi.org/specifications/
[8] A. Berlet, J. Rückert, H. Koziolek, R. Drath, M. Barth, *atp magazin* **2021**, *63 (10)*, 76–83. DOI: https://doi.org/10.17560/atp.v63i10.2569
[9] ISO 15926-4, *Industrial Automation Systems and Integration – Integration of Lifecycle Data for Process Plants Including Oil and Gas Production Facilities - Part 4: Initial Reference Data*, International Organization for Standardization, Geneva **2013**.
[10] IEC 61987, *Industrial Process Measurement and Control – Data Structures and Elements in Process Equipment Catalogues*, International Electrotechnical Commission, Geneva **2016**.
[11] IEC 61360-4, *Standard Data Element Types with Associated Classification Scheme: Part 4: IEC Common Data Dictionary (IEC CDD)*, International Electrotechnical Commission, Geneva **2017**.
[12] ISO 15926-14:2020(E), *Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 14: Industrial top-level ontology*, Working draft, **2020**.
[13] D. B. Cameron, A. Waaler, E. Fjøsna, M. Hole, F. Psarommatis, *Front. Manuf. Technol.* **2022**, *2*. DOI: https://doi.org/10.3389/fmtec.2022.945717
[14] A. Bamberg, L. Urbas, S. Bröcker, N. Kockmann, M. Bortz, *Chem. Ing. Tech.* **2020**, *92 (3)*, 192–198. DOI: https://doi.org/10.1002/cite.201900168
[15] *Working Paper: Artificial Intelligence in Industrie 4.0*, Bundesministerium für Wirtschaft und Energie, Berlin **2019**.
[16] H. A. Gabbar, *Ind. Manage. Data Syst.* **2007**, *107 (2)*, 229–250. DOI: https://doi.org/10.1108/02635570710723822
[17] G. Vogel, L. Schulze Balhorn, A. M. Schweidtmann, *Comput. Chem. Eng.* **2023**, *171*, 108162. DOI: https://doi.org/10.1016/j.compchemeng.2023.108162
[18] E. Hirtreiter, L. S. Balhorn, A. M. Schweidtmann, *Towards automatic generation of Piping and Instrumentation Diagrams (P&IDs) with Artificial Intelligence*, arXiv:2211.05583, **2022**. DOI: https://doi.org/10.48550/arXiv.2211.05583
[19] J. Oeing, W. Welscher, N. Krink, L. Jansen, F. Henke, N. Kockmann, *Digital Chem. Eng.* **2022**, *4*, 100038. DOI: https://doi.org/10.1016/j.dche.2022.100038

[20] G. Vogel, L. S. Balhorn, E. Hirtreiter, A. M. Schweidtmann, *SFILES 2.0: An extended text-based flowsheet representation*, arXiv:2208.00778, **2022**. DOI: https://doi.org/10.48550/arXiv.2208.00778
[21] T. Mortlock, D. Muthirayan, S.-Y. Yu, P. P. Khargonekar, M. Abdullah Al Faruque, *IEEE Trans. Emerg. Top. Comput.* **2022**, *10 (1)*, 34–45. DOI: https://doi.org/10.1109/TETC.2021.3132251
[22] V. Turau, C. Weyer, *Algorithmische Graphentheorie*, De Gruyter, Berlin **2015**.
[23] W. L. Hamilton, *Graph Representation Learning*, Springer, Cham **2020**. DOI: https://doi.org/10.2200/S01045ED1-V01Y202009AIM046
[24] W. L. Hamilton, R. Ying, J. Leskovec, Inductive Representation Learning on Large Graphs, in *Proc. of the 31st International Conference on Neural Information Processing Systems*, Association for Computing Machinery, New York **2017**, 1025–1035.
[25] K. Xu, C. Li, Y. Tian, T. Sonobe, K. I. Kawarabayashi, S. Jegelka, *Proc. Mach. Learn. Res.* **2018**, *80*, 5453–5462.
[26] J. Oeing, *DEXPI – Training data*, V1, Root, **2022**. DOI: https://doi.org/10.57826/KEEN/UKEMUG
[27] J. Oeing, T. Holtermann, *DEXPI2GraphML converter (python)*, **2022**. https://github.com/TUDoAD/DEXPI2graphML
[28] J. Oeing, *DEXPI2GraphML converter (python)*, V1, Root, **2022**. DOI: https://doi.org/10.57826/KEEN/NFOR42
[29] M. T. Hagan, H. B. Demuth, M. H. Beale, O. De Jesús, *Neural Network Design*, 2nd ed., eBook **2014**.
[30] T. N. Kipf, M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, ICLR 2017, **2017**. DOI: https://doi.org/10.48550/arXiv.1609.02907
[31] W. Di, A. Bhardway, *Deep Learning Essentials: Your Hands-on Guide to the Fundamentals of Deep Learning and Neural Network Modeling*, Packt Publishing, Birmingham **2018**.
[32] L. Lu, *Commun. Comput. Phys.* **2020**, *28 (5)*, 1671–1706. DOI: https://doi.org/10.4208/cicp.OA-2020-0165
[33] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in *Proc. of the 31st International Conference on Neural Information Processing Systems*, Association for Computing Machinery, New York **2017**, 972–981.
[34] T. Pham, T. Tran, D. Phung, S. Venkatesh, Column Networks for Collective Classification, in *Proc. of the AAAI Conference on Artificial Intelligence*, AAAI, Palo Alto **2017**. DOI: https://doi.org/10.1609/aaai.v31i1.10851

