

Design Patterns based on Deep Learning analyzing Distributed Data

Daniel Burkhardt, Nana Agyei-Kena, Patrick Frey, Sven Kurrle, Heiner Lasi¹

¹Ferdinand-Steinbeis-Institute, Stuttgart, Germany

{Daniel.Burkhardt, Nana.Agyei-Kena, Patrick.Frey, Sven.Kurrle, Heiner.Lasi}@steinbeis.de

Abstract. Data silos in many system landscapes complicate the creation of comprehensive information. Distributed Ledger Technology enables trust between assets via a distributed, secure and immutable storage of transactions. Deep Learning realizes intelligence to make decisions, conduct them and analyze their results based on the gathered data. In order to counteract the limitations of current system landscapes, an integrative implication of both Distributed Ledger Technology and Deep Learning is needed. Many considerations arise during the design of information systems integrating the named technologies. Transparency over the training and deployment of various Deep Learning methods on a distributed data landscape needs to be achieved. Using both a literature review and a qualitative research approach, this paper describes the development of design patterns and their selection criteria with different dimensions taken into consideration. The evaluation phase comprises of semi-structured interviews with experts from different disciplines. The result of this paper guides stakeholders in the selection of a suitable technical solution.

Keywords: Deep Learning, Distributed Data, Design Patterns, Distributed Ledger Technology

1 Introduction

Advances in the Internet of Things (IoT), the integration of IT systems and improvements in data capturing have led to data volume increases in different domains. Data is saved either in a structured or unstructured manner increasing the complexity of the storage landscape. Furthermore, non-integrative data silos hinder the creation of a comprehensive user history. In order to support the right decisions, information needs to be at the right place at the right time. Thus, methods to combine the information from the described complex data landscape and analytical tools to evaluate related data sets are required [1].

Nowadays, Machine Learning (ML) has gained attention in the analytic's spectrum. Deep Learning (DL) is a sub-method of ML which uses layered artificial neural networks (ANNs) to virtually mimic human brain structure in order to extract features from raw data [2]. In different domains, DL plays an important role due to the flexibility and performance of neural networks. Nevertheless, the application of ANNs has not been extensively evaluated due to the described complexity in combining the vast data with the appropriate tools [1].

Due to the sensitivity of especially personal data, conducting analysis is highly challenging by covering all distributed data. Distributed Ledger Technology (DLT) provides trust and governance between entities of a distributed data landscape. Due to the high variety of technologies and their speed of development, companies struggle to identify the right variant for their specific problem [3]. The creation of decentralized solutions provides a further potential, but also complicates the selection process. A conceptual layer is needed that abstracts the various implementations and supports the technology application. Thus, the following research question was derived from the previous implications.

RQ: How to select a design pattern based on Deep Learning analyzing distributed data?

This paper aims to build transparency on developments in DL by proposing a model¹ that shows different design patterns. Furthermore, to apply the model in a specific domain or to an existing problem, a list of selection criteria is developed. Practical scenarios of healthcare and expert interviews illustrate the application of the created artefact. The paper is structured as follows. First, the section on the theoretical framework presents the key terms and concepts to which the paper refers. Thereafter, related work is discussed, and the methodology is described. The results section covers the solution in the form of a design pattern model. In addition, selection criteria are provided enabling an informed decision about the most suitable design pattern. Lastly, the concept is evaluated and a conclusion with future research implications is drawn.

2 Theoretical Framework

It is identified that many domains, like healthcare, are missing knowledge of selecting suitable analytical tools, but also suffers from a scarcity of data scientists in order to access the described potential [5]. The designed model of this research work creates analytical transparency that is necessary to approach the named challenge of this paper.

¹ Models are abstractions or representations using constructs (language in which problems and solutions are explained) to represent the reality and provide the solution's feasibility [4].

2.1 Design Patterns

To reduce complexity in the usage of technologies, the method of design patterns is part of the conceptual layer above the implementation layer. In this research work, design patterns are used to describe the deployment of DL on distributed data. Furthermore, the integration of DLT is shown by the examples of selected design patterns.

Design patterns show solutions to a specific problem that have developed over time. They are visualized in an easy, well-structured and understandable form to capture the key aspects of the solution and to help grasp complex fields. As a template, design patterns provide the input for the solution implementation and therefore, grasp the requirements and show an initial direction for a solution. The goal is to create design patterns reusable over different domains in order to guarantee a broad application and robust design. Furthermore, they provide flexibility in presenting alternatives, as well as lead to an efficient solution selection through the capturing of positive past experiences. Presuming a broad usage of design patterns they support the interoperability between systems by implementing the same design [6].

In this paper, the design patterns visualize various components that are relevant for storage, processing and analysis of data. They show the coordination between the involved parties and provide information to the DL architecture and learning process.

2.2 Deep Learning

Learning can be described as the process of associating a specific output with a specific input. The goal is to accurately represent the relation between input and output. A system that has achieved this association for a specific set of inputs can predict outputs with a better-than-chance probability. This is achieved by correctly retaining the input-output relation. It allows the system to predict outputs for inputs it has not yet been trained on. The goal of ML is to automate this process in a computer code. One possible way of achieving this is by imitating the logical function of biological neurons [7].

A perceptron is a way of predicting an output based on a set of inputs. The underlying logic used to make the distinction between various outputs is based on the individual weights that are applied to the inputs. Multiple perceptrons are combined in a layered structure to enable more complex logic, for example as demonstrated by XOR [8]. These are called feed forward neuronal networks. In such a layered structure, the output of one perceptron is wired as the input of another perceptron. First layer perceptrons receive their input from outside the network and distribute their output to other perceptrons within it. Analog to this, the perceptrons in the output-layer receive their input from within the network and provide an output directed outside the system. The middle layer in this structure is called the hidden layer, as it does not directly interact with the outside [2].

To increase the level of abstraction in the system, more hidden layers are added to the network. These can increase the accuracy of the prediction and enable more complex

input-output relations [9]. The high number of weights to be trained in such a deep network requires large amounts of data in order to successfully train the network. Deep neural networks, also referred to as DL, go through three distinct stages during development.

- **Setup:** The architecture of the network is defined at this stage. This includes the number of layers, as well as the number of neurons per layer. On the individual perceptron level, initial weights are chosen, and an activation function is specified [2].
- **Training:** The weights are adjusted during training to increase the accuracy[10] of the network. This adjustment is an iterative process. It requires large amounts of input data with known output values². Input data is fed into the network and the resulting output is compared with the correct output in the dataset. If the network output is not the same as the correct output, the weights are adjusted [11].
- **Deployment:** Once the error rate is sufficiently small, the training process is halted. At this point the weights are regarded as fixed and the network is deployed. In the deployment stage, new unknown data is fed into the network and the outputs are used to gain insight into the input data.

2.3 Distributed Data

To create an aggregated data basis, centralized IT systems are developed by a single controlling entity. To bypass the single point of failure, current research focuses on decentralized data storage. According to Buterin (2017), (de)centralization can be categorized into three different types [12]:

- Architectural (de)centralization: Compilation of the physical computers of a system
- Logical (de)centralization: Compilation of the interfaces and data structure as a single monolithic object or amorphous swarm
- Political (de)centralization: Compilation of the people controlling the system

The systems of Table 1 are assigned to architectural, political or logical (de)centralization. The Interplanetary File System (IPFS) is a distributed file system that aims to connect computers to a unified data storage [13]. Initially, files are stored on the local IPFS client. Data is replicated via the system if another client requests it. Thus, IPFS is architecturally, politically and logically decentralized. IPFS has no common state as each client can decide by themselves whether to retain data or not. In addition, IPFS is not governed by a single organization or person, thereby justifying political decentralization.

Table 1. Decentralized Examples

<i>Storage System</i>	<i>Architectural</i>	<i>Logical</i>	<i>Political</i>
-----------------------	----------------------	----------------	------------------

² Only supervised learning is considered here.

	<i>(de)centralization</i>	<i>(de)centralization</i>	<i>(de)centralization</i>
IPFS	Decentralized	Decentralized	Decentralized
Dropbox	Decentralized	Centralized	Centralized
BCT	Decentralized	Centralized	Decentralized

In contrast, Dropbox as a cloud storage system serves as an example of logical and political centralization. The system works as one logical platform running under the control of a central organization. Architecturally, it can be considered decentralized as data is replicated on multiple servers for backup.

DLT provides a governance structure providing trust through features such as transparency, immutability, pseudonymity, and integrity. The purpose of DLT is to decentralize record keeping. This technology tracks and documents transactions in a network, whereby consensus between entities must be reached to avoid invalid transactions. DLT forms the principle layer of concepts like Blockchain Technology (BCT) or tangle which defines the framework for protocols like Ethereum [14] or Hyperledger Fabric [15]. Thus, BCT can serve as a supporting layer for storage systems, as it provides functionalities to track and validate entries in the network. Therefore, BCT shows an architectural decentralization. Logically, it is centralized because it is based on a common agreed state and behaves like a single system. Political decentralization is achieved because no entity controls the protocol and decisions are made based on consensus of the network [12].

Executing DL on distributed data is a current field of research which demands careful system design with a focus on communication and fault tolerance [16]. For example, federated learning is a technique that computes the model training locally on the client's datasets, while updates to the global model are communicated to the coordinating central server. Model training is thereby decoupled from the access to the raw training data. Trust in the coordinating server is required but the privacy and security of potentially sensible data are increased by a decentral model execution. Communication between the nodes and the heterogeneity of system or update intervals are challenges to this approach [17].

3 Related Work

DL has shown remarkable breakthroughs in recent years across various disciplines such as speech recognition, computer vision e.g. image classification and natural language processing such as machine translation [18]. Beyond this, design patterns in related fields have proven to be successful. The named Gang of Four coined the definition of design patterns importantly and summarized experiences of object-oriented software development in 23 design patterns. A well-known programming pattern is the Model-View-Controller (MVC) that decouples a program into the three named objects to increase reuse and flexibility [6]. Furthermore, the Industrial Internet Consortium (IIC) created the

3-tier-architecture that divides an Internet-of-Things (IoT) system into the three layers of edge, platform and enterprise tier. This pattern propagates a centralized data storage on the platform tier [19]. Reinfurt et al. (2016) proposed five design patterns for IoT systems whereas Qanbari et al. (2016) developed four design patterns for the creation of IoT edge applications. Both approaches focus on the edge tier and a more distributed data storage [20, 21]. Considering decentralized stored data, Mendis et al. (2018) created a decentralized DL architecture using BCT services where decentralized DL models are trained with private data and shared among the contributors [22]. Further, Kuo and Ohno-Machado (2018) implemented a ML algorithm for decentralized privacy-preserving predictive modeling in private BCT networks [23]. The above listed projects and activities are considering different aspects of processing and storing data. Thus, the research gap results from a missing comparability of different DL design patterns on distributed data.

4 Method

The research method according to Design Science Research (DSR) uses qualitative methods for the design and evaluation of an artefact [24]. Qualitative methods are used to investigate the research object and provide evidence for the artefact design [25]. The model of design patterns supports the development of IS systems by identifying the integration of DL and DLT based on distributed data. Thus, the research process of this paper follows the approach of design-oriented IS research which underlines the mentioned research goal. The process is divided in to the steps of analysis, design and development, evaluation and diffusion [26, 27].

4.1 Sampling and data collection

Data was collected from literature to receive input for the design phase. The literature review was conducted using the Google Scholar search engine by inserting the listed search strings³ separately or in combination with each other.

Five semi-structured interviews⁴ with experts from the fields of DLT and DL were conducted for the design evaluation [24]. A purposeful sampling was applied that included companies which provide innovative solutions of DL and DLT in distributed data environments. As this is a cutting-edge and dynamic research field, experts that have different roles were selected in order to include a variety of perspectives. Table 2 lists the interviews including the characteristics of the interviewees in chronological order.

The qualitative method of semi-structured interviews guarantees flexibility in the

³ “Artificial Intelligence”, “Machine Learning”, “Deep Learning”, “Artificial Neural Networks”, “Distributed Ledger Technology”, “Distributed Data”, “Design Patterns”, “Architectures”, “Decentralized”, “Centralized”

⁴ Transcripts of all interviews are available at the corresponding research institute

exploration of perceptions and probing of complex and sensitive topics. This characteristic was considered as necessary for the exploration of the problem area [28]. The interviews were conducted in German between March and April 2019. Each interview had a duration of 45 to 60 minutes. A questionnaire was used to provide comparability between the different sessions. It was structured in 3 parts. After the questions concerning the experts themselves and their level of knowledge, the design patterns were presented and discussed with the expert. Finally, the selection criteria were explored.

Table 2. Overview of interviewees

#	Expertise area	Expert's role	Organization size	Organization type
1	ML & DL	Researcher	Small	Research institute, AI solution provider and consulting
2	DL & DLT	CEO	Small	AI (customized) solution provider
3	DL	CEO & Product owner	Small	AI solution provider
4	Virtual Reality & DL	Shareholder & Sales lead	Small	Product & service integrator in technical customer support
5	DL & Industrial Internet	Researcher	Large	Research institute

4.2 Data analysis

For the qualitative analyses of the interviews, they were recorded and transcribed to identify concepts and their relations. Based on a existing design pattern model derived from the literature review and related work, the interview transcripts were openly coded to identify new, specific design patterns, dimensions or selection criteria. After each interview the resulting code was discussed among the authors and, following the inductive approach, identified concepts were integrated in the code system. This led to a process of iterative revisions between data collection and data analysis [29]. Lastly, the resulting code system was evaluated by comparing it with the results from the literature review after restructuring them to fit the existing design pattern categories. Categories were built by identified similarities between the concepts which resulted in a design pattern model. The resulting design pattern model is presented in the following section.

5 Design pattern model and Selection Criteria

The resulting model of design patterns is visualized in Figure 1. The design patterns are segmented horizontally and vertically into three main parts according to the various categories of data storage and DL deployment. On the horizontal dimension, the process steps of model setup, training and deployment are considered separately resulting in different characteristics. The training of the model is always executed at the location where the historical data is stored. For example, in design pattern 'B' the fragmented DL models are set up separately, but they are trained on the aggregated data storage. Lastly, the trained model is deployed on the various entities, indicated by 'DL_{A, B, C, D}'. The legend shown in Figure 1 lists the components of the design pattern model. The grey coloring indicates if the DL process is either coordinated separately, through a single party or the network. Additionally, the arrows display either the transfer of the training data, model or both.

Fragmented DL and data: These describe the isolated storage of historical data, setup, training or deployment of DL. Generated data are kept on private systems. Hence, the DL models are setup, trained and deployed internally. The experts name design pattern 'A' as the most common architecture in for example healthcare. Fragmented data is indicated in Figure 1 by 'Data_{A, B, C, D}' as well as the separated setup, training and deployment of the DL model which is represented by 'DL_{A, B, C, D}'.

Centralized DL and aggregated data: The historical data are stored centrally. Furthermore, DL model(s) are setup and deployed at a central location where it is also coordinated (indicated in grey). However, the training of the model can be executed in a distributed manner, as shown in pattern 'D'. The experts argue that most healthcare providers are moving towards centralized DL and central data storage (design pattern 'E'). Companies specialized in analytics provide capabilities to store and analyze data that most healthcare institutions omit, the expert said. Thus, data is transferred to the provider where it is analyzed by deploying a centrally trained model 'DL_{ABCD}'. Analytical results are then provided to the institutions or accessed through an interface.

Decentralized DL and distributed data: A common model 'DL_{ABCD}' or different models 'DL_{A, B, C, D}' are setup and deployed in a decentralized, coordinated manner without a central party. However, the training of the model can also be executed centrally, like in design pattern 'H'. In the third horizontal layer of the data dimension, historical data from the entities is shared among each entity, indicated by 'Data_{ABCD}' on all entities. Thus, each entity has access to the common historical data. The experts name various advantages of decentralized design patterns. Thus, the model accuracy is improved by the increased amount of data accessible for DL model training. Design pattern 'I' shows a decentralized model in which the data, as well as the model itself, are shared among the entities coordinated by a mechanism that supports distributed consensus definition. In contrast, design pattern 'C' describes common data on all nodes whereas the model is kept separate between the entities. Design pattern 'F' represents a central model deployment on

decentralized training data which is coordinated by a central unit.

5.1 Selection Criteria

The selection criteria support the decision for a suitable design pattern depending on the particular use case or problem. The criteria should not be considered separate from each other and require to be observed thoroughly for the design pattern selection. Based on the area of application and characteristics of data, the criteria vary in significance. For example, in finance data privacy, data consistency and trust might be more relevant than data volume transfer in manufacturing. The three categories data, performance and syndication divide the list of criteria into groups. Each category consists of various properties which are described in the following paragraphs.

Data privacy: For example, personal data are protected by various regulations in different domains, like the GDPR. A breach in this data could result in legal actions. For data analytics through third parties, it is sometimes necessary to extract metadata from personal records. Thus, a respective design pattern fulfilling this criterion needs to be selected. Peteiro and Guijarro (2013) support this criterion to be especially important in decentralized DL. Based on this knowledge, a systems designer is more likely to select a pattern for fragmented data, e.g. patterns ‘I’ and ‘J’.

Data quality: The data generated in institutions apply to the characteristics of volume, velocity, variety and veracity. In order to guarantee that the model is accurate to the real state, the input data needs to be in the right quality. Therefore, it is necessary to preprocess data in the right manner. An expert described this step as “shaking the data” (interview 3) in order to ensure efficient predictions and model accuracy. After this, the model deployment can be generalized. Expert 5 added that 70% of data analytics is focused on data cleaning. The process of data cleaning is a cumbersome task and best done centrally. Therefore, high data quality allows the selection of decentralized DL patterns, whereas a low data quality tends to necessitate the selection of centralized DL.

Data accessibility: Data is generated and stored at different location and protected by various mechanisms in a network of entities. To increase the accuracy of the DL model a high amount of data from ideally different locations is necessary. Accessibility describes the difficulty of accessing relevant data. A design pattern in which training data needs to be transferred, data accessibility plays an important role. Expert 1 defined data availability as critical for analytics and stated that it provides potential to identify new processes and increase productivity. High data accessibility requirements tilt the decision towards the selection of aggregated data and decentralized data patterns, as the data availability aids an immediate data transfer.

Data consistency: It describes the adherence to and validity of the data according to the definition of the data storage system protocol or governance structure. If data is identified as inconsistent, the transaction to the data set will be rolled back. Both Experts 3 and 4 described this criterion as important to provide inferences on the same syntactical

and semantical basis. Thus, a high consistency is especially relevant in aggregated data and decentralized data patterns in order to avoid data alignments to the protocol structure.

Data significance: Expert 1 defined this criterion as important because it influences the previous characteristics. It includes the significance of data for the data creator, as well as for the entities receiving the data. To measure such a criterion is seen as difficult, requiring a combination of metrics, or can alternatively be purely subjective. For example, fragmented data patterns, or the patterns 'I' and 'L' support the deployment of DL on confidential data as these patterns guarantee data privacy. Additionally, aggregated data patterns could be selected for confidential data, if a high trust requirement or policies and regulations avoid data misuse.

Performance - Model accuracy: To achieve relevant accuracy, training cycles require data in the correct quality, high amount and consistency which is mostly provided by pattern 'E'. Especially with distributed data sets, this criterion provides challenges to be solved. Expert 5 adds that an inaccurate trained model will lead to inconsistencies if it is further distributed. Moreover, different applications have different requirements to the model accuracy. For example, in cancer treatment the accuracy must be at 99% in order to reduce the error rate immensely.

Performance - Data volume transfer: As stated earlier, the training of a DL model requires a great amount of data to potentially decrease the error rate. The amount of data transferred depends on the data type. For example, sharing training data between entities or with a central location results in a higher amount than by solely transferring the DL model. Thus, fragmented data patterns provide an advantage as data reside on the nodes.

Performance - Communication overhead: The coordination necessary for the development of a DL model requires different communication efforts between the entities depending on which pattern they are trying to advance. For example, in a centralized DL a central party coordinates the process solely, whereas in decentralized DL various entities coordinate the process. As a result, this has implications on the communication overhead and selection of the pattern.

Performance - Resource availability: Depending on the requirements, DL models require different computing resources such as Graphic Processing Units (GPUs) or Central Processing Units (CPUs). These resources are available in different configurations and their combination influences the performance of DL model training and deployment. Expert 2 emphasized that in a decentralized design pattern the availability of high-performance computers is relevant but due to budget restrictions it is not feasible. Therefore, the consideration of DL pattern installation costs is essential which favors the selection of centralized patterns resulting in less coordination efforts.

Syndication - Policies and regulations: The first expert named the environment of syndication between entities as also essential for the DL model development. For instance, policies and regulations are significant for the interchangeable relations between entities exchanging and analyzing data. In a case where sensitive data is moved cross-institutionally, it is necessary to consider or even establish rules and regulations to achieve

smooth operations. Patterns of decentralized DL or data in which entities interact with each other get enabled by policies and regulations that secure the interaction.

Syndication - Trust: The first expert said that trust between the entities is necessary to create or enhance business models. Data also provides power and transparency to its user which needs to be treated with trust, as otherwise it has critical consequences for the data owner, Expert 3 said. A trusted party or the use of technology, like DLT, can provide trust. On the other side, Expert 2 named resistance against new technology as a major challenge to overcome in order to reach integration of new technologies. Consequently, based on the trust between the entities and in technologies the location of the data storage is decided.

5.2 Application of the design pattern model

The combination of the design patterns with the selection criteria supports the user in selecting the most appropriate pattern. After an explanation of the model, all experts used the model to describe their current solution and way of development. Furthermore, they identified solutions that exist on the market and associated them with the respective model.

The first expert described the healthcare sector to be implementing mostly pattern ‘A’ due to ‘data privacy’, ‘regulatory’ and ‘trust’ reasons. Nowadays, progressions to pattern ‘E’ can be identified because the significance of central analysis is recognized, as explained by Expert 1. Although, Expert 2 asserted that centrally stored heterogeneous data provide a challenge. All experts agreed that pattern ‘E’ is mostly implemented in the AI field. Expert 5 added that in pattern ‘D’ data is not stored centrally as in pattern ‘E’ and remarked about the comparability between the patterns.

Further, Expert 5 views the development of decentralized solutions to be one in which the institutions are enabled to train and deploy a model that is centrally provided, or a centrally trained model deployed in a decentralized manner. Expert 2 linked the second alternative to pattern ‘H’. According to Expert 4, the challenge is the transfer of real-time data to the trained DL model which has implications on ‘data volume transfer’. If these models detect an anomaly, they either adapt their parameters and execute an action, or transfer messages to a centralized entity. Expert 3 argued that it depends on the ‘performance’ requirements. If short response times are required, the data transfer to a central system is not suitable, which justifies pattern ‘D’. Pattern ‘G’ was further discussed with Expert 5 who emphasized that the data must be in the same structure in order to derive cross-institutional results. The expert mentioned the importance of balancing the hardware resources between the entities as the amount of communication to coordinate is markedly higher. Pattern ‘K’ was mapped to the solution of Kaggle [30] in which a dataset was provided and different data scientists competitively formulated models to solve the problem.

The development of a DL model that learns from different institutional data without

leaking private information is a crucial challenge in healthcare. Pattern ‘G’ visualizes this approach by training each classifier on different data subsets. An ensemble strategy combines the separately trained classifiers. This learning approach leads to significant increases in speed and accuracy in the model training and deployment [31].

For the future, Expert 1 foresaw the implementation of the decentralized patterns ‘I’ or ‘L’. The integration of DLT could be a solution which first requires the entities to develop their systems in order to integrate such a technology. Additionally, powerful hardware resources are required on the decentralized entities which Expert 2 does not yet view as available. However, the decentralized storage of data provides the potential to analyze data across different nodes avoiding redundancies and improving ‘model accuracy’.

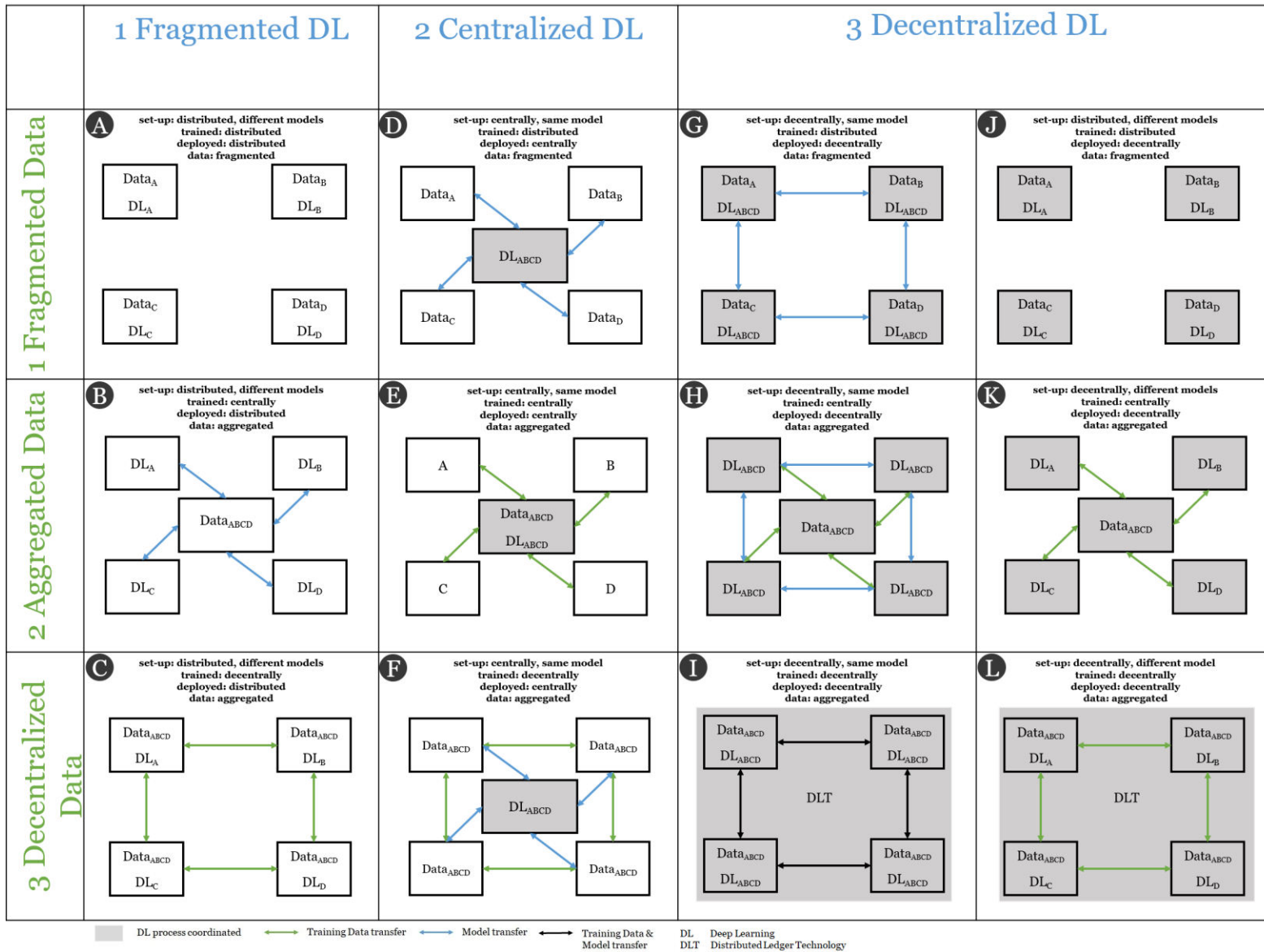


Figure 1: Design Pattern Model

6 Discussion

An advantage of the design pattern model is the differentiation of the different DL phases, Expert 2 emphasized. Furthermore, the expert sees the model to be practically applied in supporting consultants in order to define a solution based on customer requirements. Product development can be added into the model which supports communication between stakeholders. Non-IT experienced users need guidance to use such a model as the model is too complicated for such roles, the Expert 3 emphasizes. Additionally, Expert 5 suggested explaining the model by the application of a use case.

In the following text, DLT will be discussed as a specific case in this research work. The first expert described the usage of DLT in pattern 'I' or 'L' as a development in the far future. In pattern 'I' all the entities use the same model and train it in a federated way, whereas in pattern 'L' each entity uses its own model. In pattern 'L', the models can be used to analyze 'Data_{ABCD}' from different viewpoints, like analytics on cancer, diabetes, etc. DLT can then be used to vote on a specific model and log the results. Furthermore, Expert 2 estimated the potential of DLT in providing tamper-resistance and integrity in networks in which multiple entities interact. However, currently the integration of DLT will have an impact on the system's performance, the expert argued. In both patterns the entities can be coordinated by using a DLT protocol. Thus, the entities can securely and privately provide respective data that can be used for the model training. Furthermore, the transparency of the transactions and any data actions provide trust between the entities which can foster openness of data exchange. This would encourage the possibility to setup, train and deploy a model on a decentralized network using the combined capacity of each entity. Technically, the meta-data of the models can be saved in a DLT protocol whereas the model itself can be saved in a mass decentralized storage, like IPFS. This will bypass the storage limitations of the current DLT protocols. Furthermore, with the possibility to use crypto assets, economic structures can be implemented to enable services between the entities [15]. For example, healthcare institutions can sell anonymized patient data to a research institute focusing on cancer prevention. The use of DLT provides interoperability between different systems of the healthcare stakeholders and, as such, guarantees the creation of a comprehensive data history of a patient. In addition, the use of decentralized DL analytics supports the decision making of the various stakeholders using the syndication of several entities.

The model provides comparability of the design patterns and transparency which supports the design of a solution and thus, saves costs in a later implementation phase. In communication between different stakeholders, the model provides a tool that supports discussion and promotes understanding between different parties.

This work has some limitations. The evaluation was conducted by using qualitative semi-structured interviews with five experts. Due to the limited number of experts, the evaluation of the design pattern model can justify its application in practice and profoundness only based on initial statements, and furthermore, can guarantee no generalization. Thus, future studies on the model and its selection criteria need to be conducted. For example, further use cases from different areas need to be examined in

order to testify the design pattern structure and the selection criteria.

7 Conclusion

In this study, a model consisting of 12 different design patterns showing the deployment of DL on distributed data was developed. Additionally, the result provides a list of criteria divided into the categories of 'Data', 'Performance' and 'Syndication' to assist its application and support the selection of a respective pattern.

In conclusion, the application of design pattern models shows potential for academia and practice. Rapidly progressing research within DL and DLT result in a highly complex solution space. An abstraction layer provides transparency and comprehensibility between different solutions and supports the integration of technologies in a field in which no generalized view is provided. Especially in domains like healthcare, a decentralized DL model deployment can create value whereby different institutions maintain their own data silo but are at the same time restricted from transferring sensitive data to a centralized storage. DLT provides trust between entities. Its integration enacts disintermediation and provides direct entity-to-entity interactions. Based on this, the stated potential can be realized. For instance, this supports the patient to get treated using the vast knowledge that the healthcare ecosystem provides. Finally, the paper provides a view of the integration of both technologies and emphasizes its potential.

References

1. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J.T.: Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics* 19, 1236–1246 (2018)
2. Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. MIT Press, Cambridge, Massachusetts, London, England (2016)
3. Erbay, H., Yıldırım, N.: Technology Selection for Digital Transformation: A Mixed Decision Making Model of AHP and QFD. In: Durakbasa, N.M., Gencyilmaz, M.G. (eds.) *Proceedings of the International Symposium for Production Research 2018*, 104, pp. 480–493. Springer International Publishing, Cham (2019)
4. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Quarterly*, 75–105 (2004)
5. Mehta, N., Pandit, A.: Concurrence of big data analytics and healthcare: A systematic review. *International journal of medical informatics* 114, 57–65 (2018)
6. Gamma, E., Vlissides, J., Johnson, R., Helm, R.: *Design patterns. Elements of reusable object-oriented software*. Addison-Wesley, Boston (1998)
7. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65, 386 (1958)
8. Yanling, Z., Bimin, D., Zhanrong, W.: Analysis and study of perceptron to solve XOR problem. In: *Proceedings*, pp. 168–173. IEEE, [Piscataway, NJ] (2002)
9. He, K., Zhang, X., Ren, S., Sun, J.: *Deep Residual Learning for Image Recognition* (2015)

10. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18 (2017)
11. LeCun, Y.: A theoretical framework for back-propagation. In: Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA (1988)
12. Buterin, V.: The Meaning of Decentralization, <https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274>
13. Protocol Labs: IPFS Documentation, <https://docs.ipfs.io/>
14. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper 151, 1–32 (2014)
15. Burkhardt, D., Werling, M., Lasi, H.: Distributed Ledger. Definition & Demarcation. In: Conference proceedings ICE/IEEE ITMC. IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), pp. 1–9. IEEE, Piscataway, NJ (2018)
16. Li, M., Andersen, D.G., Park, J.W., Smola, A.J., Ahmed, A., Josifovski, V., Long, J., Shekita, E.J., Su, B.-Y.: Scaling Distributed Machine Learning with the Parameter Server. In: 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14), pp. 583–598. USENIX Association, Broomfield, CO (2014)
17. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-Efficient Learning of Deep Networks from Decentralized Data (2017)
18. Yan Liu, Jimeng Sun: Deep Learning Models for Health Care: Challenges and Solutions. *Complex Intell. Syst.* (2017)
19. Lin, S.W., Miller, B., Durand, J., Bleakley, G., Chigani, A., Martin, R., Murphy, B., Crawford, M.: The Industrial Internet of Things Volume G1: Reference Architecture. IIC:PUB: G1:V1.80:20170131 (2017)
20. Reinfurt, L., Breitenbücher, U., Falkenthal, M., Leymann, F., Riegg, A.: Internet of things patterns. In: Eloranta, V.-P. (ed.) Proceedings of the 21st European Conference on Pattern Languages of Programs, pp. 1–21. ACM, New York, NY (2016)
21. Qanbari, S., Pezeshki, S., Raisi, R., Mahdizadeh, S., Rahimzadeh, R., Behinaein, N., Mahmoudi, F., Ayoubzadeh, S., Fazlali, P., Roshani, K., et al.: IoT Design Patterns: Computational Constructs to Design, Build and Engineer Edge Applications. In: IEEE First International Conference on Internet-of-Things Design and Implementation, pp. 277–282. Piscataway, NJ (2016)
22. Mendis, G., Sabounchi, M., Wei, J. and Roche, R.: Blockchain as a Service: An Autonomous, Privacy Preserving, Decentralized Architecture for Deep Learning (2018)
23. Kuo, T.-T., Ohno-Machado, L.: ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks (2018)
24. Hevner, A., Chatterjee, S.: Design Research in Information Systems. Theory and Practice. Springer Science+Business Media LLC, Boston, MA (2010)
25. Sarker, S., Xiao, X., Beaulieu, T.: Guest editorial: qualitative studies in information systems: a critical review and some guiding principles. *MIS Quarterly* 37, iii–xviii (2013)
26. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on design-oriented information systems research. *European Journal of Information Systems* 20, 7–10 (2011)
27. Peffers, K., Tuunanen, T., Niehaves, B.: Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research. *European Journal of Information Systems* 27, 129–139 (2018)
28. Myers, M.D., Newman, M.: The qualitative interview in IS research: Examining the craft. *Information and Organization* 17, 2–26 (2007)

15th International Conference on Wirtschaftsinformatik,
March 08-11, 2020, Potsdam, Germany

29. Mayring, P., Fenzl, T.: Qualitative content analysis: theoretical foundation, basic procedures and software solution. In: Baur, N., Blasius, J. (eds.) *Handbuch Methoden der empirischen Sozialforschung*, 3, pp. 633–648. Springer Fachmedien Wiesbaden, Wiesbaden (2019)
30. Kaggle Inc.: Kaggle is the place to do data science projects, <https://www.kaggle.com/>
31. Peteiro-Barral, D., Guijarro-Berdiñas, B.: A survey of methods for distributed machine learning. *Prog Artif Intell* 2, 1–11 (2013)