ConceptSuperimposition: Using Conceptual Modeling Method for Explainable AI

Wolfgang Maass[1,2], Arturo Castellanos[3], Monica Chiarini Tremblay[3], Roman Lukyanenko[4] and Veda C. Storey[5]

[1]Saarland University, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Germany
[3]William & Mary, Williamsburg, VA, USA
[4]HEC Montréal, Montreal, Québec, Canada
[5]Georgia State University, Atlanta, GA, USA

Abstract

Many artificial intelligence (AI) applications involve the use of machine learning, which continues to evolve and address more and more complex tasks. At the same time, conceptual modeling is often applied to such real-world tasks so they can be abstracted at the right level of detail to capture and represent the requirements for the development of a useful information system to support an application. In this research, we develop a framework for progressing from human mental models of an application to machine learning models via the use of conceptual models. Based on the framework we develop a novel ConceptSuperimposition method for increasing explainability of machine learning models. We illustrate the method by applying machine learning to publicly available data from the Home Mortgage Disclosure Act database which contains the 2020 mortgage application data collected in the United States. The machine learning task is to predict whether a mortgage is approved. The results show how the explainability of machine learning applications can be improved by including domain knowledge in the form of a conceptual model that represents a mental model, instead of relying solely on algorithms. Preliminary results show that including such knowledge can help advance the explainability problem.

Keywords

Artificial Intelligence, Machine Learning, Conceptual Modeling, Model Performance, Mental Models, Framework for Mental Models and Conceptual Models for Machine Learning,, ConceptSuperimposition

1. Introduction

Machine learning consists of methods that use data and algorithms to build models that make inferences about an application from provided examples [1]. Both the opportunities and limitations of machine learning are rooted in its reliance on building models from data and, therefore

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022), Stanford University, Palo Alto, California, USA, March 21–23, 2022.

wolfgang.maass@iss.uni-saarland.de (W. Maass); arturo.castellanosbueso@mason.wm.edu (A. Castellanos); Monica.Tremblay@mason.wm.edu (M. C. Tremblay); roman.lukyanenko@hec.ca (R. Lukyanenko); vstorey@bellsouth.net (V. C. Storey)
https://iss.uni-saarland.de (W. Maass)
0000-0003-4057-0924 (W. Maass); 0000-0002-7477-7379 (A. Castellanos); 0000-0003-1289-6679 (M. C. Tremblay); 0000-0002-8735-1553 (V. C. Storey)

on the quality of the data used to train and test the models [2]. As our society's dependence on machine learning grows, it is important to ensure that machine learning models perform well and are interpretable and transparent. This is a tradeoff that is often augmented by opaque transformations in the input data (i.e., feature engineering), which makes it challenging to assess the effectiveness of the input data on the outcome [3]. Numerous challenges persist, including biases, discrimination, lower performance, lack of transparency, and explainability. While popular approaches have emerged for explaining predictions of classifiers (Layerwise relevance propagation, Local Interpretable Model-Agnostic Explanations, and many others) [4], there are criticisms about explanations being dependent on the choice of hyperparameters and how these models have different explanations for similar instances in the data. The objective of this research is to investigate how to improve machine learning explainability by incorporating domain knowledge of the application. The contribution is to propose a framework for progressing from human mental models to machine learning models through the use of conceptual models. Based on the framework we develop a ConceptSuperimposition method for increasing explainability of machine learning models. We illustrate the application of this method in the home mortgage domain.

2. Machine Learning and Conceptual Modeling

Supervised learning guides the learner in acquiring knowledge in a domain through examples, so new cases can be handled in a manner most appropriate based on the knowledge learned from similar cases. Modern supervised machine learning has been taking advantage of the availability of data, and developing methods and techniques (e.g., deep learning neural networks, reinforcement learning), which rely on large volumes of high-quality data for performance improvement. The increase in the use of complex machine learning models has brought about challenges in explaining the decision logic of these models. Transparency research in AI is a growing societal concern and a growing research area [4, 5]. A generally overlooked approach to explainability, however, is how to incorporate domain knowledge that a user or designer might possess. This knowledge would manifest itself in mental models which contribute to the development of conceptual models that support the interpretation of machine learning models and outcomes. Conceptual modeling formally describes "some aspects of the physical and social world around us for the purposes of understanding and communication" [6, p. 2]. Humans use conceptualizations about domains for representing specific or abstract situations in domains. Recent research has proposed combining conceptual modeling with artificial intelligence or, specifically machine learning [7, 8, 9, 10, 11, 12, 13]. The main argument is that doing so can provide reliable rules about the domain without being dependent on extracting them from the data. Despite these efforts, conceptual models are rarely used in the process of machine learning. At the same time, machine learning invariably relies on human mental models – representations of reality in the minds of data scientists or users of machine learning models, who either develop or interpret machine learning solutions, in light of their mental models. Unlike conceptual models, mental models are not explicit, and hence may contain biases.

Research in psychology and other disciplines (e.g., philosophy, cognitive science), dealing with decision-making, has argued that, when making decisions, humans construct one or more

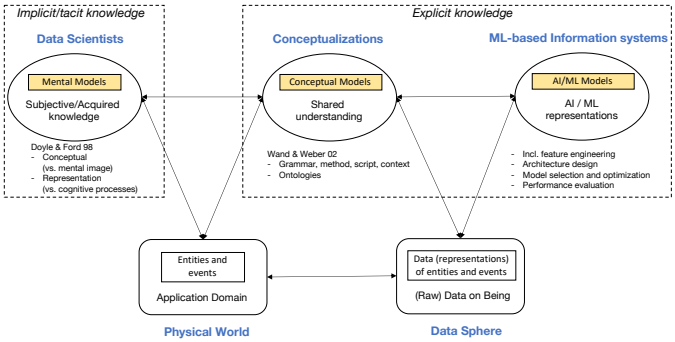

Figure 1: Framework for Mental Models and Conceptual Models for Machine Learning

mental models of a domain [14, 15]. A mental model is a representation of a possible state of affairs in reality [15]. Mental models are the central mechanism for coping with diversity and change, allowing someone to act in an informed and effective manner. In a typical machine learning scenario, we may encounter different customers purchasing goods or services from a store. However, we can reduce this complexity by constructing a handful of mental models that segment customers (e.g., loyal repeat customer, high volume customers, one-time buyer). Mental models, are important for problem-solving, including in ML contexts, but are also prone to bias and error. If left unchecked, the biases arising from the suboptimal formation and use of mental models, could affect the judgment of machine learning developers, and result in a variety of machine learning problems. Hence, we seek to augment mental models with conceptual models so the representations will be more formal and externalized and better equipped at mitigating the cognitive biases inherent in mental models. If, in reality, mental models are inconsistent, or even contradictory, with the machine learning model or the data, it is possible that: (1) subjects have selected inadequate conceptual statements for a situation, (2) perception and measurement about reality are distorted, or (3) conceptual statements are flawed. In this paper we, therefore, propose improving explainability of complex machine learning models based on the externalizing the mental models via conceptual models. Mental models [14] can represent components, states and structural relations, basic operational rules and on general scientific principles, events and processes, and ontological knowledge. They represent individual knowledge extracted from either physical or abstract domains that is used for various cognitive tasks, including understanding, communication, navigation, and decision making. Mental models are often understood as surrogate models used for simulation and prediction as well as a means for constructing advanced mental models [16]. Conceptual models are shared representations expressed in various forms, such as texts and graphics. According to model theory, conceptual models are projected and abbreviated representations made by human experts and used for a purpose [17]. In contrast, machine learning models are statistical abstractions derived by algorithmic fitting mathematical functions to data according to a purpose given by an objective functions. Data used for model fitting and also for construction of conceptual models are representations of domains themselves. In essence, mental models are individually constructed, conceptual models are socially constructed, and machine learning models are

algorithmically constructed. Designing information systems means that all three model types are synchronized and balanced by negotiation for finding a satisfiable equilibrium between these models. (See Figure 1). A gap exists between mental models and conceptual models versus machine learning models because internal representations of machine learning models are generally inaccessible. Thus, research is needed on interpretability and explainability of machine learning models as a means for alignment of all three model types.

3. *ConceptSuperimposition* Method

Following the Framework for Mental Models and Conceptual Models for Machine Learning (cf. Figure 1), we advance a new method – ConceptSuperimposition. This is a method which extends our early work on Superimposition method. Superimposition adds structural semantic information to the outputs of machine learning to support explainability [10]. While this information is absent in current ML practice, it is routinely employed by humans to understand their day-to-day experiences.

As per our Framework, a machine learning model is a model of some domain (e.g., credit card fraud, image classification, online auctions). The machine learning model is a set of rules for estimating a value of interest or discriminating among the cases of interest based on previously provided domain examples.

Superimposition maps the output of machine learning models (i.e., the features, rules and transformation functions) onto a conceptual model of the domain. The method suggests indicating inside the conceptual model information about the rules of the machine learning models. This step depends on the type of machine learning model. For example, if a regression model is used, these rules can be represented as feature weights or feature coefficients. These coefficients can be appended to the attributes in the conceptual model, or the attributes can be highlighted differently to indicate the different relative importance of each attribute. Features are conceived and defined based on domain knowledge. For instance, the feature $property\_value$ is necessary for evaluating a loan application. A conceptual model clusters features into concepts (e.g., classes, entity types). The relations (e.g., association, type of, part of) provide connections between the concepts. Generally, concepts that are directly connected via a relation are semantically closer to each other than indirectly connected concepts.

The final step of the Superimposition method involves analyzing the resulting conceptual model to gain a clearer understanding of the underlying rules machine learning models use to makes its decisions, and to identify opportunities to improve the machine learning model further.

A key limitation of the early version of the Superimposition method is in the lack of representation of concepts and their relationships. The method merely attached feature weights to the conceptual model. Yet, the understanding of the impact of the concepts (e.g., entity types) on the target is also of critical importance. Furthermore, the early formulation of the Superimposition lacked formalization. We address both shortcomings in the formalized and holistic ConceptSuperimposition method. We propose a ConceptSuperimposition method for assessing the alignment between conceptual models and machine learning models. This method consists of three steps:

1. **Marginal contribution**: determination of Shapley values for predicting features
2. **Feature Contribution**: local contribution of outcome features associated with outcome concepts.
3. **Concept Contribution**: local contributions on outcome concepts.

3.1. Marginal contribution

Machine learning (ML) models are globally fitted to datasets. Often, ML models are black boxes without direct access to feature contributions to outcomes. Therefore, simpler models (surrogate models) are locally fitted ex-post to ML models. Surrogate models provide information on local contribution of features to outcomes (e.g., LIME or SHAP [18]). SHAP (Shapley Additive Explanation) values are Shapley values of a conditional expectation function of the machine learning model, i.e., the fitted model is used for determining local contribution of single features to an outcome.

Shapley values formalize coalition games and determine additive marginal contributions of single players to an overall payoff of the coalition of players. They are defined by an operator ϕ that assigns for each game v a vector of payoffs $\phi(v) = (\phi_1, ..., \phi_n)$. $\phi_i(v)$ is player i's marginal and additive contribution to the outcome of a game over all permutations with all other players [19]. Shapley values are locally accurate, i.e. match the original model $f(x)$, is not affected by missing values and are consistent wrt. unequality relation between two models $f(x)$ and $f'(x)$ [18]. The Shapley value of a feature i is a weighted mean of its marginal value of feature i, averaged over all possible subsets of features:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)]$$

with F the set of all features and $f_{S\cup\{i\}}$ with a model trained with feature i and $f_S(x_S)$ without. SHAP values determine additive contributions of input features X on the prediction of an output feature y, i.e., $y = f(X)$, agnostic to the machine learning model used with f the original model and Φ a vector of all Shapley value ϕ_i for all independent features. Input feature $x \in X$ is transformed into simplified vector $z \in Z$ with $z \in \{0, 1\}$, i.e., SHAP value $\phi_i z_i$ is zero if feature z_i is missing and ϕ_i carries the whole *marginal contribution* of a feature to an outcome otherwise.

3.2. Feature contribution

We now use marginal contributions for defining *conceptually generalized contributions* of input features. Given a conceptual model CM with a bidirectional mapping of concepts to all features in O. For each concept c in CM, a n-ary *concept contribution* vector g_c^o is constructed by Hadamard product of Shapley vector Φ and input vector x^c for an output feature o in output concept O. x^c has only feature values associated with concept c and value 0 everywhere else. Vector g_C^o is the contribution of all input concepts on an outcome feature o.

$$g_c^o = \Phi \circ x^c \text{ and } g_C^o = \Pi_{c \in C} g_c^o \circ \mathbf{1}_n$$

First the average of all SHAP values per input feature is calculated. $\bar{g}_c^o$ is the normalized value for all g_c^o , ie., relative contribution of feature to an outcome feature.

By normalization over all input data X, $\bar{g}_c^o$ is determined that provides relative contributions of each feature of concept c with respect to output feature o, ie. input features are superimposed on c relative to o.

3.3. Concept contribution

Concept contributions depend on the type of outcome features, i.e. application of classification or regression. For binary classification, we define $f_O(X)$ as the sum of contributions of all feature contributions of input concepts on a feature o in output concept O except contributions of features in O due to implicit strong collinearity with the output feature o:

$$f_O(X) = \sum_{c \in C \setminus \{O\}} \bar{g}_c^{o\top} \cdot \mathbf{1}_n$$

For regression, feature contributions are evaluated relative to features of concept O. The mean of output features except outcome feature o are calculated (ω). Only those input features are considered with a larger feature contribution than ω.

$$f_O(X) = \sum_{c \in C \setminus \{O\}} \bar{g}_c^{o\top} \cdot (1/\omega_n - 1)$$

Concept contributions are the weighted summation of all summed up feature contributions. The weight w is the count of all features minus selected features plus 1. This accounts for decreasing feature contribution values with the number of features. Only features with strong feature contributions are selected. If, for instance, only one input feature is selected, feature contributions are not affected while a large w will reduce feature contributions on concept contributions.

$$\kappa_O(X) = 1/w * \sum_{o \in O} f_O(x)$$

It shall be noted, that $\kappa_O(X)$ depends on the type of prediction. Values for $\kappa_O(X)$ for classifications can be compared with one another and for regression respectively but $\kappa_O(X)$ cannot be compared between classification and regression.

κ_O is determined for permutations over all homogeneous concepts $c \in C$, i.e., κ_O is determined for all concepts $c \in C$. This provides a measure for local contributions of input concepts on a homogeneous concept given input x. Concept contributions on heterogeneous concepts requires a functional model that integrates features $o \in O$.

A concept c with little concept contribution $\kappa_O(X)$ on an output concept c_p has a weak conceptual relation with c_p, i.e., the outcome is only weakly affected by the presence of c_i. Conceptual relations are directed form c_i to c_p due to the game-theoretic construction of Shapley values. The semantics of conceptual relations are constrained by the concepts.

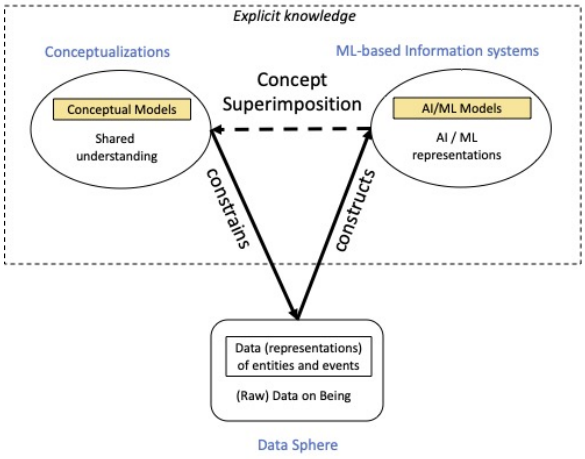

Figure 2: ConceptSuperimposition

3.4. ConceptSuperimposition

Feature contributions provide a directed similarity measure between input features and output features analog to local similarity measures in social networks (e.g., [20]). Feature contributions are consistent with the properties of additive feature attribution methods, ie. local accuracy, missingness and consistency [18].

Concept contribution abstracts from features to concepts and determines directed contributions between directly connected concepts (cf. *Game 1* [21]). Additive feature attribution properties are not maintained because lack of output values on concept level. Therefore, concept contribution is a score that measures directed local contribution of one concept to another derived by feature contributions.

We call the attribution of concepts by concept contributions *ConceptSuperimposition*. It closes a cycle between conceptual models, data and ML models that consists of three steps. First, conceptual models provide constraints on data that is considered for ML model development. Second, data is used for constructing ML models. Concept contributions elevate patterns found by ML models to a conceptual level that is fed back to conceptual models. Thus, concept contributions can be used for confirmation of conceptual models, i.e., for evaluation whether identified patterns from data are consistent with conceptual models and, thus, shared understanding of actors involved.

4. Example

To illustrate the application of ConceptSuperimposition, we use publicly available data (10 GB) from the Home Mortgage Disclosure Act (HMDA) website (https://www.consumerfinance.gov/data-research/hmda/). This data contains the 2020 mortgage application data collected in the U.S. under the Home Mortgage Disclosure Act. The dataset consists of a sample of 3,481,348 applications for single-family, principal residence

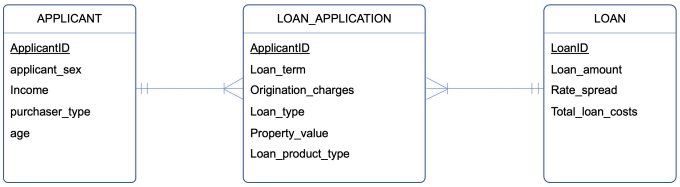

Figure 3: Conceptual Model for the HMDA loan dataset

purchases, i.e. 5% of the HMDA dataset. The data is comprised of 99 variables including payment history, credit history, credit mix, demographics, income, and characteristics of the loan (e.g., purpose of the loan, interest rate, total loan costs), and census data (e.g., census tract, tract population).The target variable is to predict whether a mortgage is originated (target = 1) or denied (target = 0). Of the sample applications in the dataset, 44.3% of the applications were for refinancing, 32.42% were for home purchase, 14.96% for cash-out refinancing. 83.56% of the applications were approved. 69.99% of the applications belonged to White applicants and 6.15% to Black or African American applicants (approval rate for Whites was 85.33% and Blacks was 71.51%). (see Figure 3 with a fragment of the conceptual model).

For comparison of concept contributions, data needs to be standardized. Categorical features are transformed by one-hot-encoding except output feature y. After data engineering, the dataset contains 52 features from which 20 are associated to the concept *applicant*, 14 to the concept *loan* and 18 to the concept $loan\_application$.

4.1. Feature Contribution

All five concepts are used for making a binary decision on loan applications, i.e. concept *Decision* with a feature $action\_taken$. We use XGBoost Classification for predicting $action\_taken$. Performance metrics for the model on the test dataset are: accurary (0.995), precision (0.999), recall (0.995), F-measure (0.997).

For all three concepts, i.e., *Applicant*, *Loan* and *LoanApplication*, we determined feature contributions on the concept *Decision* modeled by a single feature ($ActionTaken$) and three permutations, i.e. (1) *LoanApplication* and *Loan* on *Applicant*, (2)) *Applicant* and *Loan* on *LoanApplication* and (3) *Applicant* and *LoanApplication* on *Loan*.

4.2. Concept contributions

For each input concept (here: applicant, loan application, and loan), concept contributions are determined. As a result, $applicant$ provides a strong concept contribution $\kappa_{applicant}(X)$ of 0.610 to *LoanApplication* while $loanapplication$ provides a relatively high concept contribution $\kappa_{loanapplication}(X)$ of 3.177 to $loan$ (cf. Table 2). We only used one feature per outcome concept for simplicity.

For the HDMA dataset, concept contributions indicate a strong directed connection from *applicant* to $loan\_application$ (cf. Figure 4). This supports the establishment of a relation between these two concepts. Same holds for $loanapplication$ to $loan$ and $loan$ to $applicant$. The latter concept contribution is not captured by the original conceptual model (cf. Figure 3).

Concepts		**Decision (C)**	**Applicant (C)**	**Loan Application (R)**	**Loan (R)**
	Features (total 21)	*ActionTaken*	*PurchaseType*	*PropertyValue*	*LoanAmount*
Applicant	income	-0.014	-0.014	0.161	0.387
	purchaser_type	0.266	0.265		
	age			0.449	
Loan Application	loan_term	0.081	-0.081		
	origination_charges	0.073	0.073		0.868
	loan_type	0.034	0.034		0.567
	property_value	0.037			
	loan_product_type				1.742
Loan	loan_amount	-0.025	-0.026		
	rate_spread	-0.026	-0.026		
	total_loan_costs	0.160	0.160		-0.838

Table 1
Feature contributions for two outcome concepts based on classification (C) and two by regression (R)

$\kappa_c(x)$	Applicant (C)	LoanApplication (R)	Loan (R)
Applicant	1	**0.610**	0.387
LoanApplication	0.0262	1	**3.177**
Loan	**0.108**	mc	1

Table 2
Concept contributions $\kappa_c(x)$

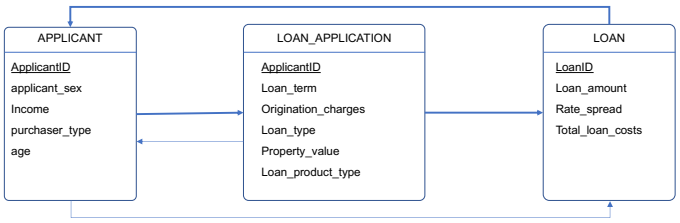

Figure 4: ConceptSuperimposition onto HMDA conceptual model

This is an example how concept contributions κ_p can be leveraged for automatically scrutinization of conceptual models associated with datasets and database implementations. In this example, we found support for a revised conceptual model with relations between *loan* towards *applicant*. It shall be noted that concept contributions do not provide link semantics, such as cardinalities or type of links. Proposals for revisions based ConceptSuperimposition help domain experts and ML developers and business analysts to align conceptual models with data and ML models.

5. Conclusion

This research proposes that ML models and conceptual models can be used effectively for analyzing prior conceptual knowledge used for data selection. A conceptual model represents agreed-upon domain knowledge. In this work we first provide a theoretical basis for using conceptual models in the domain of explainable AI. Within use the Mental Models Framework to develop a formalized and holistic ConceptSuperimposition method. The method is formalized and its utility is demonstrated in the application to the home mortgage domain.

The new ConceptSuperimposition method can be used to improve ML explainability and should be especially effective for situations where there is insufficient data to extract all relevant domain knowledge in a data-driven manner. Future work is needed to apply the framework and method to other examples in other domains. In future studies we hope to evaluate the increased transparency due to the new method by conducting interviews, focus groups, and laboratory experiments with the stakeholders looking to understand the decision logic behind machine learning models. We also plan to investigate the benefit of this method together with other existing approaches to explainability. We do not position this method as an alternative, rather, we believe it could complement existing methods by extrapolating their outputs onto the conceptual models.

In future work, we will address the integration of concept contributions for classification and regression.

References

[1] P. McCorduck, C. Cfe, Machines who think: A personal inquiry into the history and prospects of artificial intelligence, CRC Press, 2004.

[2] V. S. Sheng, F. Provost, P. G. Ipeirotis, Get another label? improving data quality and data mining using multiple, noisy labelers, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 614–622.

[3] M. C. Thrun, A. Ultsch, Swarm intelligence for self-organized clustering, Artificial Intelligence 290 (2021) 103237.

[4] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (xai), IEEE access 6 (2018) 52138–52160.

[5] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[6] J. Mylopoulos, Conceptual modeling and telos, in: P. Loucopoulos, R. Zicari (Eds.), Conceptual Modeling, Databases, and CASE: An Integrated View of Information Systems Development, Editors John Wiley & Sons, 1992, pp. 49–68.

[7] W. Maass, V. C. Storey, Pairing conceptual modeling with machine learning, Data & Knowledge Engineering (2021) 101909.

[8] U. Reimer, D. Bork, P. Fettke, M. Tropmann-Frick, Preface of the first workshop models in ai., in: Modellierung (Companion), 2020, pp. 128–129.

[9] D. Bork, A. Garmendia, M. Wimmer, Towards a multi-objective modularization approach for entity-relationship models., in: ER Forum/Posters/Demos, 2020, pp. 45–58.
[10] R. Lukyanenko, A. Castellanos, V. C. Storey, A. Castillo, M. C. Tremblay, J. Parsons, Superimposition: augmenting machine learning outputs with conceptual models for explainable ai, in: International Conference on Conceptual Modeling, Springer, 2020, pp. 26–34.
[11] R. Lukyanenko, A. Castellanos, J. Parsons, M. C. Tremblay, V. C. Storey, Using conceptual modeling to support machine learning, in: International Conference on Advanced Information Systems Engineering, Springer, 2019, pp. 170–181.
[12] W. Maass, I. Shcherbatyi, Inductive discovery by machine learning for identification of structural models, in: J. Trujillo, K. C. Davis, X. Du, Z. Li, T. W. Ling, G. Li, M. Lee (Eds.), Conceptual Modeling - 37th International Conference, ER 2018, Xi'an, China, October 22-25, 2018, Proceedings, volume 11157 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 545–552.
[13] W. Maass, I. Shcherbatyi, Data-driven, statistical learning method for inductive confirmation of structural models, in: T. Bui (Ed.), 50th Hawaii International Conference on System Sciences, HICSS 2017, Hilton Waikoloa Village, Hawaii, USA, January 4-7, 2017, ScholarSpace / AIS Electronic Library (AISeL), 2017, pp. 1–10.
[14] D. Gentner, A. L. Stevens, Mental models, Psychology Press, 2014.
[15] P. N. Johnson-Laird, Mental models: Towards a Cognitive Science of Language, Inference, and Consciousness, Harvard Univ Press, Cambridge, MA, 1983.
[16] P. N. Johnson-Laird, Comprehension as the construction of mental models, Philosophical Transactions of the Royal Society of London. B, Biological Sciences 295 (1981) 353–374.
[17] H. Stachowiak, Allgemeine modelltheorie, Springer, 1973.
[18] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st international conference on neural information processing systems, 2017, pp. 4768–4777.
[19] L. S. Shapley, A value for n-person games, in: H. W. Kuhn, A. W. Tucker (Eds.), Contributions to the Theory of Games (AM-28), Volume II, Princeton University Press, 2016, pp. 307–318.
[20] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A: statistical mechanics and its applications 390 (2011) 1150–1170.
[21] T. P. Michalak, K. V. Aadithya, P. L. Szczepanski, B. Ravindran, N. R. Jennings, Efficient computation of the shapley value for game-theoretic network centrality, Journal of Artificial Intelligence Research 46 (2013) 607–650.