ICIS 2022 Proceedings

Data Analytics for Business and Societal Challenges

Dec 12th, 12:00 AM

AI Explainability: Embedding Conceptual Models

Wolfgang Maass
Saarland University, wolfgang.maass@dfki.de

Arturo Castellanos
William & Mary, aacastellanosb@wm.edu

Monica Tremblay
The College of William and Mary, monica.tremblay@mason.wm.edu

Roman Lukyanenko
HEC Montreal, romanl@virginia.edu

Veda C. Storey
Georgia State University, vstorey@gsu.edu

Recommended Citation

Maass, Wolfgang; Castellanos, Arturo; Tremblay, Monica; Lukyanenko, Roman; and Storey, Veda C., "AI Explainability: Embedding Conceptual Models" (2022). *ICIS 2022 Proceedings*. 12.
https://aisel.aisnet.org/icis2022/data_analytics/data_analytics/12

AI Explainability: A Conceptual Model Embedding Method

Short paper

Wolfgang Maass
Saarland University & German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
wolfgang.maass@iss.uni-saarland.de

Arturo Castellanos
William & Mary
Williamsburg, VA, USA
arturo.castellanosbueso@mason.wm.edu

Monica Chiarini Tremblay
William & Mary
Williamsburg, VA, USA
Monica.Tremblay@mason.wm.edu

Roman Lukyanenko
University of Virginia
Charlottesville, VA, USA
romanl@virginia.edu

Veda C. Storey
Georgia State University
Atlanta, GA, USA
vstorey@gsu.edu

Abstract

Artificial intelligence, especially efforts based on machine learning, is rapidly transforming business operations and entire industries. However, as many complex machine learning models are considered to be black boxes, both adoption and further reliance on artificial intelligence depends on the ability to understand how these automated models work – a problem known as explainable AI. We propose an approach to explainability that leverages conceptual models. Conceptual models are commonly used to capture and integrate domain rules and information requirements for the development of databases and other information technology components. We propose a method to embed machine learning models into conceptual models. Specifically, we propose a Model Embedding Method (MEM), which is based on conceptual models, for increasing the explainability of machine learning models, and illustrate through an application to publicly available mortgage data. This machine learning application predicts whether a mortgage is approved. We show how the explainability of machine learning can be improved by embedding machine learning models into domain knowledge from a conceptual model that represents a mental model of the real world, instead of algorithms. Our results suggest that such domain knowledge can help address some of the challenges of the explainability problem in AI.

Keywords: Machine learning, conceptual models, Artificial Intelligence, Model Embedding Method (MEM), explainability

Introduction

Machine learning consists of methods that use data and algorithms to build models that make inferences from provided examples (McCorduck and Cfe 2004). Both the opportunities and limitations of machine learning are rooted in its reliance on building models from data and, therefore, on the quality of the data used to train and test the models (Sheng et al. 2008). As our society's dependence on machine learning grows, it is important to ensure that machine learning models perform well, are compliant with legal and ethical requirements, and are interpretable and transparent for different types of users. This trade-off that is often exacerbated by opaque transformations in the input data (feature engineering) that makes it challenging to assess the effectiveness of the input data on the outcome (Thrun and Ultsch 2021). Numerous challenges persist, including biases, discrimination, lower performance, lack of transparency, and explainability (Arrieta 2020). While popular approaches have emerged for explaining predictions of classifiers Adadi and Berrada 2018), 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 (Probst et al. 2019).

The goal of this research is to develop a Model Embedding Method (MEM) for embedding machine learning models into conceptual models as a means of assessing the alignment between conceptual knowledge of experts and data-driven knowledge. The method leverages the predictions of a machine learning model to link features together based on the importance of the predicted features. Relationships between concepts can be identified by projecting data features onto attributes in the conceptual model. This can be used to evaluate which relationships of the conceptual model are supported by the machine learning model. Thus, the conceptual model becomes an interpretive framework for machine learning models. By embedding the machine learning model into conceptual models in this way, similarities and differences between the two can be identified. If there is strong agreement, there is a high probability that the predictive behavior of the machine learning model will conform to the conceptual model. If there is a low level of agreement, the machine learning model behaves significantly differently than assumed by domain experts. This can lead to misbehavior with unintended side and after effects (Storey et al. 2022) or may uncover novel facts about the domain.

Machine Learning and Conceptual Modeling

The increase in the use of complex machine learning models has brought up challenges in explaining the decision logic of these models. Transparency research in AI is a growing societal concern (Adadi and Berrada 2018; Arrieta 2020). For example, constrained models and post-hoc explanation techniques can help in building responsible AI systems (Arrieta et al. 2020). A generally overlooked approach to explainability, however, is how to incorporate domain knowledge that a user or designer might possess. By building a digital mental model, the mental feature provides more information that can help in the detection of fake news (Ding et al. 2020). In our work we seek to leverage knowledge of domain experts, externalized as conceptual models, to allow for detection of biases or even unintended behavior of machine learning models. Information Systems Designers use conceptual models to express individual mental models and creation of shared understanding.

Recent research has proposed combining conceptual modeling with artificial intelligence or, specifically machine learning (Bork et al. 2020; Lukyanenko et al. 2019; Lukyanenko et al. 2020; Maass and Storey 2021). 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 building machine learning models or to increase machine learning model transparency and interpretability. 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 individual mental models. This inevitably leads to differences between the shared understanding of the information system design team and the behavior of the machine learning model.

Feature Attribution Models

Machine learning (ML) models are globally fitted to datasets. Predicting an output features means that a prediction shall be as close as possible to real values (ground truth) within a domain; i.e., minimizing a loss function. Some machine learning models provide feature importance values, such as various types of decision trees while feature importance of neural networks are notoriously fragile (Ghorbani et al. 2019). 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 Lundberg and Lee 2017). 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 Shapley 2016. 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 $f'(x)$ (Lundberg and Lee 2017).

Model Embedding Method

Following the theory-grounded arguments for incorporating conceptual models for explainability, we advance a new method, the Model Embedding Method (MEM), motivated by the popular practice of word embedding in Natural Language Processing (Bengio et al. 2000). MEM also benefits from imposing feature weights to the conceptual model. Superimposition is a concept suggested to graphically impose feature weights - outputs of machine learning - onto conceptual models to show which entities (concepts) the weights belong to (Lukyanenko et al. 2020). However, this early idea did not consider how to aggregate these weights per concept nor the contribution of the relationships among concepts to transparency. Hence, although promising, Superimposition falls short of leveraging the domain semantics captured in conceptual models, and merely placed conceptual models as backgrounds for feature weights. The analytical approach of MEM extends prior work on inductive discovery of conceptual models by data-driven models (Maass and Shcherbatyi 2018). Model Embedding is the first method which, driven by theoretical arguments, *embeds* functional behavior of machine learning models into conceptual models to analyze compliance between conceptual models and machine learning models. It also increases transparency and explainability of machine learning models by conceptual models.

The proposed Model Embedding Method consists of four steps:

1. **Marginal contribution**: determination of attribution values for input features to output features
2. **Feature Contribution**: mapping local contributions of input features with associated outcome feature of corresponding attributes of a conceptual model.
3. **Concept Contribution**: identification of directed relationships between concepts based on feature contributions.
4. **Concept Mapping**: interpretation of concept contributions in context of a conceptual model

Marginal contribution

We now use marginal contributions for defining *conceptually generalized contributions* of input features. Given a conceptual model CM with a bidirectional mapping of concept attributes to all features in a dataset O. For each feature marginal contribution for the prediction of outcome features is calculated. Here, SHAP values are used to represent additive contributions of input features to the value of output features according to Shapley's model: represented by vector Φ. Marginal contributions are irrespective of association of features to attributes and concepts; i.e., marginal contributions are determined by machine learning models independent of conceptual models.

Feature contribution

For each concept c in CM, a n-ary *concept contribution* vector g_c^o is constructed by the Hadamard product of marginal contribution 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 for input dataset x^c.

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

$\bar{g}_c^o$ is the scaled mean value for all marginal contributions in g_c^o for input dataset x^c; i.e., relative contribution of feature to an outcome feature. Values are min-max scaled without shifting by the mean $(x/(max - min))$. This allows comparison of different feature contributions. For heterogeneous concept definitions, a dominant feature is selected that is associated with the most important attribute of a concept.

Concept contribution

Attributes of conceptual models and features of datasets used for machine learning are natural counterparts. By mapping data schemas of both, feature contributions are transferred as attribute contributions of conceptual models. Data schema mappings can be complex (cf. Batini et al. 1986). In the following, we assume one-to-one mappings. Concept contributions depend on the type of outcome features. Categorical outcome attributes require classification models while cardinal attributes are predicted by regression models. Concept contributions is the sum of feature contributions of all features mapped with attributes of a concept c on outcome feature o in output concept O. Counteracting feature contributions reduce effect sizes of concept contributions while aligned feature contributions increase effect sizes. We define $f_O(X)$ as the sum of feature contributions of input concepts on feature o mapped to output concept O, excluding contributions of features mapped to O due to implicit strong collinearity with the output feature o:

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

$\kappa_O(X)$ calculates the mean of feature contributions of a concept $c \in C$ on another concept O.

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

κ_O is determined for permutations over all 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 concept given input x.

So far, conceptual-modeling scripts (cf. Wand and Weber 2002) are designed according to syntactic, semantic, and pragmatic properties (Lindland et al. 1994). It does not matter if the attributes serve as input for the prediction of other attributes. Therefore, concepts are designed independent of predictive consistency of attributes. For information systems that use machine learning models, conceptual models can also support the design and implementation of better machine learning models. This means that inconsistencies of concept definitions with respect to predictions are to be minimized so that they better reflect the behavior of intended machine learning models. A conceptual model provides higher predictive consistency if attributes are homogeneous in predicting attributes of concepts that they are connected with by conceptual relationships. This is discussed in more detail in the following.

A concept c with little concept contribution $\kappa_O(X)$ on an output concept c_p has a weak conceptual relationship with c_p, i.e., the outcome is only weakly affected by the presence of c_i. Conceptual relationships based on concept contributions are directed from c_i to c_p due to the game-theoretic construction of Shapley values. However, $\kappa_O(X)$ is the mean value over all feature contributions. Therefore, opposing feature contributions can single out one another. Analysis of feature contributions in the the context of an input concept is needed for identification of such effects. Various cases can be distinguished. First, feature contributions have little effect, i.e., the $\kappa_O(X)$ is a sufficient proxy for feature contributions. Second, feature contributions are mainly positive or negative. This indicates that a concept is homogeneously defined relative to an outcome concept, i.e. the distribution of feature contributions of a concept towards an outcome concept is homogeneous. This can be seen as support for a clear conceptual relationships between both concepts from a data analytical perspective in addition to syntactic, semantic and pragmatic qualities (Lindland et al. 1994). In contrast, if the distribution of feature contributions are multi-modal with negative and positive feature contributions, this can be interpreted as weak support of conceptual relationships between both concepts. Predictive inconsistency is a metric for misalignment between conceptual models and machine learning models.

Concept Mapping

Concept contribution abstracts from features to concepts and determines directed contributions between directly connected concepts (cf. *Game 1* Michalak et al. 2013). Additive feature attribution properties are

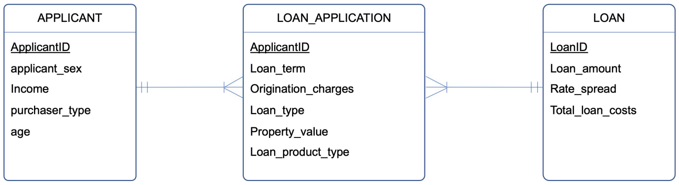


Figure 1. Conceptual Model for the HMDA loan dataset

not maintained because of a lack of output values at the concept level. Therefore, concept contribution is a score that measures directed local contribution of one concept to another derived by feature contributions.

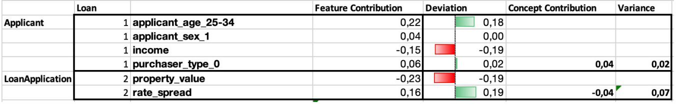
We call the attribution of concepts by concept contributions *Concept Mapping*. It closes a cycle between conceptual models, data and ML models that consists of three steps. First, conceptual models provide constraints on data that is considered for ML model development. Second, data is used for constructing 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 evaluating whether identified patterns from data are consistent with conceptual models, resulting in a shared understanding by the actors involved.

Example

To illustrate the application of the Model Embedding Method, 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 United States under the Home Mortgage Disclosure Act. The dataset consists of a sample of 3,481,348 applications for single-family, principal residence 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%). Figure 1 provides 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 $loan\_application$.

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 (0.994), precision (0.999), recall (0.994), F-measure (0.996).



	Loan		Feature Contribution	Deviation	Concept Contribution	Variance
Applicant	1	applicant_age_25-34	0,22	0,18		
	1	applicant_sex_1	0,04	0,00		
	1	income	-0,15	-0,19		
	1	purchaser_type_0	0,06	0,02	0,04	0,02
LoanApplication	2	property_value	-0,23	-0,19		
	2	rate_spread	0,16	0,19	-0,04	0,07

Figure 2. Example feature contributions for *Loan*

In Figure 2, feature contributions of input concepts $Applicant$ and $LoanApplication$ on $Loan$ are presented. For instance, $income$ has a feature contribution value of -0.15 on $Loan$ and its dominant feature $LoanAmount$; i.e. $income$ is weak in predicting $Loan$, while $property\_value$ predicts a reducing effect and $rate\_spread$ predicts an increasing effect on $Loan$ (cf. Figure 2).

$\kappa_c(x)$	Applicant	LoanApplication	Loan	Decision
Applicant	1	0.09	0.04	-0.11
LoanApplication	-0.05	1	-0.04	0.02
Loan	0.40	0.09	1	-0.18

Table 1. Concept contributions $\kappa_c(x)$ of initial model

Concept contributions

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

For each input concept (applicant, loan application, and loan), concept contributions are determined. A strong concept contribution exists between $Loan$ and $Applicant$ (0.40), i.e., $Loan$ supports conceptual relationships to $Applicant$ (cf. Table 1). Analysis of feature contributions show that all values are negative, i.e. pointing in the same direction. Other concept contributions to the three input concepts are rather weak.

In contrast, concept contributions of the three input concepts to $Decision$ are consistent for $Loan$ but not for $Applicant$ and $LoanApplication$. $purchaser\_type\_0$ and $property\_value$ are strongly negative while the others are positive, resulting in small concept contribution values. Similar inconsistencies exist for $LoanApplication$ to $Applicant$ with a strongly negative effect of $loan\_amount$, $Applicant$ to $Loan$ with negative effect of $income$, and $LoanApplication$ to $Loan$ with strongly negative effect of $property\_value$.

We use a collector pattern for integrating all attributes that lead to inconsistencies. For this, concept $Context$ is created with attributes $purchaser\_type\_0$ and $property\_value$. Concept contributions show that no inconsistencies are present for this revised conceptual model with respect to predicting $decision$. This provides a conceptual model that is properly aligned with the behavior of the corresponding machine learning model.

Concept mapping

Merging inconsistent attributes into another concept improved concept contribution of input concepts to the outcome concept $decision$ (cf. Figure 3). $Applicant$ and $LoanApplication$ positively contribute to $decision$ while $Loan$ and $Context$ have a negative predictive effect on $decision$. Concept $Context$ has a strong negative predictive effect on $decision$ due to negative feature contributions of $purchaser\_type\_0$ (-0.52). Inspection of the HDMA data description reveals that *code 0* indicates an unknown value; not a missing value. Proper values are, for instance, *Fannie Mae*, *Freddie Mac*, *private securitizer*, *commercial bank savings bank*, or *credit union mortgage*.

Concept contributions on $Decision$ are positive for $Applicant$ and $Loan$ but negative for $LoanApplication$ and $Context$. Interpretability on $Decision$ means that higher attribute values for $Applicant$ and $Loan$ have a positive effect on $Decision$ while lower attribute values for $LoanApplication$ and $Context$ have a negative effect. Similar interpretation is applied between input concepts. For instance, higher attribute values of $Applicant$ have a negative effect on $LoanApplication$. This can be interpreted as socio-economic moderation effects on loan applications. Note that $Context$ has a strong predictive effect on $Decision$ but also $Applicant$ and $Loan$. This indicates that $purchaser\_type\_0$ and $property\_value$ are leaking features, i.e., at least one highly correlates with $action\_taken$. Analysis shows that $purchaser\_type\_0$ correlates with $Action\_Taken$ with -.76.

This is an example how concept contributions κ_p can be leveraged for automatically scrutinizing conceptual models associated with datasets and database implementations. In this example, we found support for a revised conceptual model with argument for a collector concept $Context$. The revised conceptual model better supports predictive behavior of the underlying data set and, thus, is better aligned with the machine learning model for predicting decision of loan applications. MEM also helped to identify leaking features ($purchaser\_type\_0$). In summary, the revised conceptual model is means for interpretation of the machine

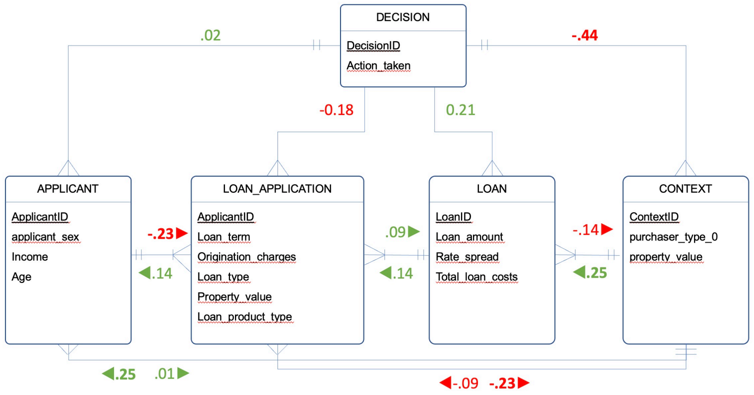


Figure 3. Revised concept model with conceptual contribution values

learning model.

Conclusion and Future Research

With the Model Embedding Method, we introduce a method that derives relationships between concepts (concept contributions) and uses this, in turn, to analyze the consistency of conceptual models. Predictive analysis provides a novel perspective for evaluation of conceptual models extending prior approach (Wand and Weber 2002). MEM trains several machine learning models including the model that is actually used by the information system for prediction of a particular outcome feature. We presented an example for predicting decisions on loan approval. MEM trained additional machine learning models between all concepts and provided insightful information on inconsistent concept definitions. This was used to revise the conceptual model and provides an improved capability for predicting the outcome concept $decision$. Therefore, the revised conceptual model has an improved alignment with the machine learning model and can be used for interpretation of the machine learning behavior. This means that MEM works in both directions: (1) embedding of machine learning models into conceptual models; and (2) embedding conceptual models into machine learning models. This provides a novel means for explainable AI based on conceptual models.

The Model Embedding Method addresses an important challenge of machine learning explainability and bridges conceptual model and machine learning (cf. Maass and Storey 2021). Future work is needed to apply the method to other examples in other domains. In future studies, we plan to evaluate the increased transparency due to the new method by conducting interviews, focus groups, and laboratory experiments with the stakeholders seeking to understand the decision logic behind machine learning models.

To illustrate one such evaluation, consider a laboratory experiment with target decision makers. The context is evaluating a trained machine learning model for screening applicants for a job. In one condition, the participants receive an output of a regression in a form of a formula along with the list of features sorted by predictive importance. This corresponds to a common practice in using feature weights for explainability. In the treatment condition we show the formula along with the MEM-based conceptual model. The participants would then be asked to perform activities which ascertain their level of understanding of the rules of the machine leaning model. In addition, we will administrate a self-reported questionnaire to collect measures of perceived model transparency and intentions to use this model in a real-world setting.

We also plan to investigate the benefit of the MEM method compared to other existing approaches to explainability, such as LIME. We suggest that MEM complements existing methods by extrapolating their outputs onto the conceptual models.

References

Adadi, A. and Berrada, M. (2018). "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)," *IEEE access* (6), pp. 52138–52160.

Arrieta, A. B. e. a. (2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion* (58), pp. 82–115.

Arrieta, A. B., Diaz-Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., et al. (2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion* (58), pp. 82–115.

Batini, C., Lenzerini, M., and Navathe, S. B. (1986). "A comparative analysis of methodologies for database schema integration," *ACM computing surveys (CSUR)* (18:4), pp. 323–364.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). "A neural probabilistic language model," *Advances in neural information processing systems* (13).

Bork, D., Garmendia, A., and Wimmer, M. (2020). "Towards a Multi-Objective Modularization Approach for Entity-Relationship Models." in *ER Forum/Posters/Demos,* pp. 45–58.

Ding, J., Hu, Y., and Chang, H. (2020). "BERT-based mental model, a better fake news detector," in *Proceedings of the 2020 6th international conference on computing and artificial intelligence,* pp. 396–400.

Ghorbani, A., Abid, A., and Zou, J. (2019). "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence,* vol. 33. 01, pp. 3681–3688.

Lindland, O. I., Sindre, G., and Solvberg, A. (1994). "Understanding quality in conceptual modeling," *IEEE software* (11:2), pp. 42–49.

Lukyanenko, R., Castellanos, A., Parsons, J., Tremblay, M. C., and Storey, V. C. (2019). "Using conceptual modeling to support machine learning," in *International Conference on Advanced Information Systems Engineering,* Springer, pp. 170–181.

Lukyanenko, R., Castellanos, A., Storey, V. C., Castillo, A., Tremblay, M. C., and Parsons, J. (2020). "Superimposition: augmenting machine learning outputs with conceptual models for explainable AI," in *International Conference on Conceptual Modeling,* Springer, pp. 26–34.

Lundberg, S. M. and Lee, S.-I. (2017). "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems,* pp. 4768–4777.

Maass, W. and Shcherbatyi, I. (2018). "Inductive Discovery by Machine Learning for Identification of Structural Models," in *Conceptual Modeling - 37th International Conference, ER 2018, Xi'an, China, October 22-25, 2018, Proceedings,* J. Trujillo, K. C. Davis, X. Du, Z. Li, T. W. Ling, G. Li, and M. Lee (eds.). Vol. 11157. Lecture Notes in Computer Science. Springer, pp. 545–552.

Maass, W. and Storey, V. C. (2021). "Pairing conceptual modeling with machine learning," *Data Knowl. Eng.* (134), p. 101909.

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

Michalak, T. P., Aadithya, K. V., Szczepanski, P. L., Ravindran, B., and Jennings, N. R. (2013). "Efficient computation of the Shapley value for game-theoretic network centrality," *Journal of Artificial Intelligence Research* (46), pp. 607–650.

Probst, P., Boulesteix, A.-L., and Bischl, B. (2019). "Tunability: Importance of hyperparameters of machine learning algorithms," *The Journal of Machine Learning Research* (20:1), pp. 1934–1965.

Shapley, L. S. (2016). *17. A value for n-person games,* Princeton University Press.

Sheng, V. S., Provost, F., and Ipeirotis, P. G. (2008). "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,* pp. 614–622.

Storey, V. C., Lukyanenko, R., Maass, W., and Parsons, J. (2022). "Explainable AI," *Communications of the ACM* (65:4), pp. 27–29.

Thrun, M. C. and Ultsch, A. (2021). "Swarm intelligence for self-organized clustering," *Artificial Intelligence* (290), p. 103237.

Wand, Y. and Weber, R. (2002). "Research commentary: information systems and conceptual modeling—a research agenda," *Information systems research* (13:4), pp. 363–376.