Ontology Concept Extraction Algorithm for Deep Neural Networks

Andrew Ponomarev, Anton Agafonov
St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)
St. Petersburg, Russian Federation
ponomarev@iias.spb.su, aaagafonov29@gmail.com

***Abstract*—An important drawback of deep neural networks limiting their application in critical tasks is the lack of explainability. Recently, several methods have been proposed to explain and interpret the results obtained by deep neural networks, however, the majority of these methods are targeted mostly at AI experts. Ontology-based explanation techniques seem promising, as they can be used to form explanations using domain terms (corresponding to ontology concepts) and logical statements, which is more understandable by domain experts. Recently, it has been shown, that inner representations (layer activations) of deep neural network can often be aligned with ontology concepts. However, not every concept can be matched with the output of every layer, and it can be computationally hard to identify the particular layer that can be easily aligned with the given concept, which is aggravated by the number of concepts in a typical ontology. The paper proposes an algorithm to address this problem. For each ontology concept it helps to identify neural network layer, which produces output that can be best aligned with the given concept. These connections can then be used to identify all the ontology concepts relevant to the sample and explain the network output in a user-friendly way.**

I. Introduction

Deep artificial neural networks have become an integral part of the modern AI toolbox. In many recognition and prediction problems, they allow one to obtain very accurate results, sometimes even surpassing the human level. However, the application of neural network-based solutions in critical domains is severely limited by the lack of explainability. Failure to understand why a neural network came to a certain conclusion reduces trust to the results, obtained using neural network technology.

The problem of neural networks explainability and interpretability is actively studied [1] and many explanation techniques have been proposed. However, most of these techniques are designed for and can be effectively used only by AI experts, and are not as understandable and easy to use by domain experts. Tying explanations to the domain vocabulary and explicit domain knowledge encoded in the form of domain ontology can increase the understandability of these explanations to domain experts. It is shown experimentally, that building explanations using ontology concepts familiar to human experts, actually increases their understandability [2].

This paper is based on the recent results of [9], showing that internal representations built by a deep neural network (neuron activations in inner layers of a deep neural network) can often be aligned with concepts of a domain ontology relevant to the target concept of the "main" network. The authors of [9] successfully train so-called *mapping networks* allowing to verify if a sample is relevant to the given ontology concept (other than the concept used to train the main net). Knowing the set of relevant concepts one can use an ontology-based inference to form a post-hoc explanation of "main" network (it would be expressed as a set of concepts relevant to the sample and ontology axioms, linking these concepts with each other and target concept). The authors of [9] also propose an algorithm to identify the set of neurons, which activations can be best of all aligned with the specified concept (this process is called "concept extraction"). However, this algorithm requires training of a large number of mapping networks, which may be time-consuming, especially if the ontology is large (mapping networks are trained for each concept).

This paper aims at improving the concept extraction process. We have performed a series of experiments to understand which layers of a deep neural network are most promising for extracting certain concept, characterized by its relationship with the target concept of the main network (we call this concept extractability or concept localization). These experiments allowed us to identify certain patterns, e.g., roughly unimodal shape of concept extractability w.r.t. the layer of the main network. Based on these observations, we propose a heuristic algorithm that can significantly reduce the complexity of concept extraction.

The rest of the paper is structured as following. Section II provides short review of ontology-based explanations of neural networks. Section III contains problem definition. Section IV briefly describes the experiments on ontology concept localization in deep convolutional neural networks and their outcomes. Section V describes the proposed algorithm.

II. Related Work

With the upsurge in eXplainable AI (XAI) research, ontology-based explanation techniques for neural networks attract more and more attention of the research community. Experimental studies [4], [5] show that the use of knowledge of the problem domain, encoded in the form of ontology, improves the quality of explanations.

Variety of ontology-based neural explanation methods have been proposed to-date. The first criterion for their classification is related to whether the entire model is explained (for example, approximating the logic of the model using well-interpreted decision trees [4]) or the result of a prediction for one particular

sample represented by the model (for example, indicating what exactly constituent parts (features) of the sample influenced the decision in one way or another). Another criterion that is important in a practical sense is the division into so-called post-hoc methods and methods for constructing self-explainable models. Post-hoc methods aim at explaining existing models trained without making specific requirements for explainability, while methods for building self-explanatory models present certain requirements for the structure of machine learning models (in particular, deep learning) in order to ensure their explainability.

A number of works are aimed at the formation of self-explainable neural models [8], [10], [11]. Traditionally, they are considered to be less accurate, but there are other views on this matter [12]. The idea is to design the structure of the NN so that it to some extent correspond to the ontology (e.g., one neuron corresponds to each concept).

Post-hoc explanation techniques are very important in practice, because they can be applied to third-party networks, without costly process of re-training and architecture adjustments that may hamper the network predictions quality. Local post-hoc explanation technique is proposed, e.g., in [5]. Similarly to LIME (a popular explanation generic explanation technique, not using ontologies), it is supposed to train a local surrogate model capable of simulating the predictions of the explained model in the vicinity of a given example. However, explanations are generated in the form of rules constructed using the concepts of ontology. The paper [13] also proposes a way to extend existing explanations with ontologies (which is reported to increase the ease of interpretations).

There are also adaptations for global explanation methods to take into account ontologies. In particular, [4] proposes the TREPAN RELOADED algorithm that builds a decision tree approximation and uses the knowledge of the problem area, expressed in the form of ontologies (when forming the next partition while constructing a decision tree).

In general, "black box" approximation is a popular approach to build and explainable model. However, [9] proposes quite different approach. The idea is that to explain the predictions of a given NN one may train several "mapping" networks connecting activations of the network being explained to ontology concepts. Using these networks one can obtain concepts relevant to the current sample and use ontology reasoning to explain the predictions.

This paper follows the ideas introduced in [9], where it is shown that internal representations of a neural network can often be aligned with ontology terms. The process of establishing a relationship between ontology concept and the set of activations of a NN is called *concept extraction*. However, not all concepts can be extracted from any layer, therefore, finding network layers (or even neurons) that can be matched with the given concept becomes a computationally expensive task. The authors of [9] propose an algorithm for it that requires training of $C \times L$ mapping models, where C is the number of concepts and L is the number of layers in the network being explained. We propose a more efficient algorithm, based on some observations about ontology concept localization in neural networks.

III. Problem Definition

The problem addressed in this paper can be defined in the following way. There is a deep neural network (we consider convolutional neural networks for image processing) trained for sample classification, i.e. outputting the probability that the sample under consideration belongs to some class T. This network is later referred to as "main" network and the class T as the target class of the main network. The network may have an arbitrary architecture with a restriction that it should be composed of a sequence of layers (2D convolutions, activation, batch normalization, pooling, etc.), e.g., it may follow one of the popular architecture for object recognition – ResNet, MobileNet, EfficientNet, etc. The network may be trained by third party, it cannot be somehow altered, besides, there may be no sufficient data to train it. However, it is possible to track activations of the network, produced during the forward pass.

There is also a domain ontology, defining a number of concepts that may be relevant to the samples. The target class T of the main network is described in this ontology, therefore, T is connected to other classes and using this ontology one can infer if an individual belongs to class T based on its features (and other concepts it belongs to). According to [9], ontology concepts used in the definition of the target concept T are called "relevant to T", or simply "relevant", as there is only one target concept in one problem. The set of relevant concepts can be denoted as RELEVANT(T).

There is a dataset, connecting samples (of the same type as the inputs of the main net) with relevant concepts, i.e., for the samples of this dataset it is known not only whether they belong or not to the class T, but whether they belong to each of the classes in RELEVANT(T). Although it may be difficult to annotate samples with all the relevant classes, the number of samples in such dataset can be rather small (significantly smaller than the number of images required to train main net) [9].

It is shown in [9], that in many cases using the dataset mentioned above it is possible to train a mapping networks reliably predicting the probability that a sample belongs to the classes in RELEVANT(T) based on activations of the main net. This process is called "concept extraction". However, the total set of activations of the main net is typically very large, and it is inefficient to consider all the activations (besides, it increases the size of the training dataset).

The problem, therefore, is to identify subset of neurons of the main network, from which the relevant concepts can be extracted with maximal reliability. In this paper, we consider only subsets formed by activations of entire layers of the main network. Therefore, the problem reduces to selecting for each relevant concept the layer of the main net, from which the respective concept can be extracted best (the respective mapping network has the best prediction quality, measured by some classification metric, e.g., ROC AUC).

IV. Critical Observations

We have done a series of experiments on ontology concept localization in (convolutional) neural networks. This section summarizes the outcomes of these experiments and describes

some of the observed patterns important for the proposed algorithm.

The goal of the experiments was to analyze the localization of ontology concepts in neural networks of various architectures to find some patterns (e.g., some dependency of the concept localization from the relationship between this concept and the target concept of the network being explained). By concept localization here we mean describing layers of a neural network by the reliability of extracting this concept from the output of the respective layer by a specified mapping network architecture. Or, in a more narrow sense, from which layer the concept can be extracted with maximum precision.

In our experiments we use the Explainable Abstract Trains Dataset (XTRAINS), introduced in [14] – a synthetic image dataset, designed to facilitate research on explanation and justification. The dataset contains 500000 images of 152 × 152, depicting schematic representations (drawings) of trains with some random background (one train in each image – see Fig. 1).

The trains can be assigned certain categories based on visual characteristics: types of wagons (e.g., length, wall shape, roof shape, number of wheels), their load (depicted as a set of geometrical shapes drawn inside a wagon), wheel size, wagons' spacing, train's position and angle.

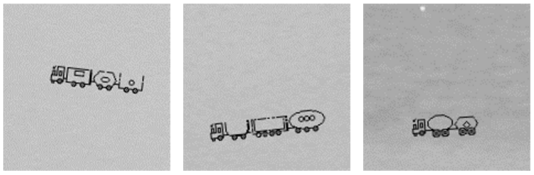

Fig 1. Sample images from the trains dataset

$$\text{TypeA} \equiv \text{WarTrain} \sqcup \text{EmptyTrain}$$

$$\text{WarTrain} \sqsupseteq \exists \text{has.ReinforcedCar} \sqcap \exists \text{has.PassengerCar}$$

$$\text{EmptyTrain} \equiv \forall \text{has.}(\text{EmptyWagon} \sqcup \text{Locomotive}) \sqcap \exists \text{has.EmptyWagon}$$

Fig 2. Relevant fragment of the ontology

All possible categories are defined in the ontology, accompanying the dataset. A subset of this ontology is shown in Fig. 2. Some of the ontology concepts correspond directly to visual appearance of a wagon, e.g. EmptyWagon is a wagon that do not carry any load (no geometric symbols inside wagon representation). Other concepts have logical definitions in terms of the ontology. Train types are typically defined using role "has" that establishes a connection between a train and a wagon. E.g., EmptyTrain is defined as a train that can have only empty wagons and locomotive and must have at least one empty wagon.

Artificial concept TypeA is introduced to serve as a classification target of the network being explained. It is connected to other ontology concepts (defined using them), and the dataset can be used to check if explanation technique can detect this connection and leverage it for explanations. This is to model a situation when a network is trained on some complex target that, however, can be logically expressed via other domain concepts. There are two other similar artificial concepts for the same purpose in the dataset: TypeB and TypeC. Though in this paper we show the results with TypeA only, the results of experiments with other types are very similar and follow the same patterns.

Each image of the dataset is annotated with binary attributes, corresponding to ontology concepts. Therefore, for each image not only target label is known (e.g., whether it belongs to TypeA or not), but also relation to all ontology concepts, which allows one to train and verify concept extraction models.

In our experiments, we trained convolutional networks of various architectures for binary classification of target concepts (TypeA, TypeB or TypeC). Following [9], we call them "main" networks. In particular, we trained a convolutional neural network of a "custom" architecture (consisting of several convolutional layers with ReLU activation and batch normalization followed by a couple of fully-connected layers), as well as ResNet-18 and MobileNet-V2. In total, we had 9 networks (three architectures for each of the three target concepts). All of these networks had prediction quality for the target concept about 0.99 (ROC AUC) on a test set (100000 images). Then, for each of these networks we trained concept mapping networks of two various architectures. Concept mapping networks are trained for each concept, using output of some layer of a main network. We evaluate concept extraction quality by ROC AUC.

Layer number	Layer type	WarTrain	EmptyTrain	ReinforcedCar	PassengerCar	EmptyWagon
2	BN	0.5000	0.5000	0.5000	0.5000	0.5000
4	BN	0.5000	0.5000	0.5000	0.5000	0.5000
7	BN	0.6225	0.7444	0.7244	0.5723	0.6063
9	BN	0.9508	0.9718	0.9984	0.9927	0.9619
12	BN	0.9380	0.9769	0.9893	0.9855	0.9157
14	BN	0.9911	1.0000	1.0000	0.9982	0.9884
17	BN	0.9866	0.9998	0.9999	0.9978	0.9537
19	BN	0.9991	1.0000	0.9997	0.9981	0.6907
20	FC	0.9996	1.0000	0.9996	0.9989	0.6583
21	FC	0.9993	0.9999	0.9995	0.9985	0.6684

Fig 3. ROC AUC for concept extraction by shallow mapping network from some of the layers of ResNet-18 main network for TypeA trains classification

Two mapping network architectures were considered: the first architecture (referred to as shallow mapping network) consists of an input layer and one output neuron, and the second network (referred to as 10-5 mapping network) consists of an input layer, followed by three fully connected layers containing 10, 5 and 1 neuron, respectively (hidden layers have ReLU activation). Next, we trained concept mapping networks for each ontology concept *relevant* to the type of the corresponding train. Relevant concepts are those concepts that participate in the definition of the target concept (type of train) – see the ontology fragment in Fig. 2.

The quality data (ROC AUC) of the obtained mapping networks for ResNet-18 main network architecture classifying TypeA trains is shown in Fig. 3.

Based on the experiments, following observations has been made. Most of the concepts relevant to the target concept of the main nets can be reliably extracted from activations of some layer of the main NN even by shallow mapping networks. It basically confirms the findings of [9]. Some concepts are harder to extract (e.g., EmptyWagon), however, they also can be extracted in some configurations.

Main networks with more layers have some advantage: the ontology concepts can be extracted from MobileNet-V2 or ResNet-18 with the same or better reliability than from custom architecture network having only 7 layers. One can hypothesize that complex multi-layered networks produce richer set of inner abstractions that can more easily be aligned with the specified set of abstractions (ontology).

As expected, more expressive mapping networks can extract concepts more reliably – the concept extraction quality of 10-5 mapping network is superior w.r.t. the concept extraction quality of the shallow one. However, very complex mapping networks would require more training effort (training data for mapping networks have to be annotated with all relevant concepts).

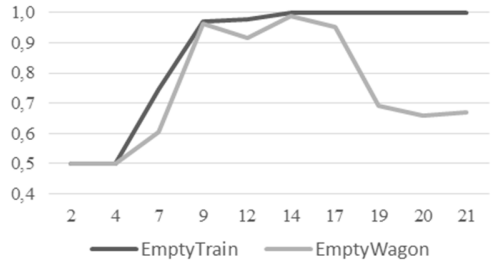

Fig 4. Higher (EmptyTrain) and lower (EmptyWagon) level concept localization

The concept extraction quality has roughly unimodal form (w.r.t. the layer of the main network). For most of the "high-level" concepts close to the target concept of the main net extraction quality maximum is located somewhere near the head of the main network (the last several fully-connected layers), however, some lower-level concepts (e.g., EmptyWagon) seem to be recognized by the main net on earlier layers, and then are "forgotten" – replaced by more complex concepts, used in the definition of the target concept. We can see this process, for example, in Fig. 3 and Fig. 4. Inner representation can be aligned with EmptyWagon concept (strictly speaking, "has.EmptyWagon") with increasing quality until layer 14. This concept is important to define EmptyTrain, which, in its turn, is used to define target concept (TypeA). After layer 14 the network's inner representation is well-aligned with EmptyTrain and it doesn't "need" "has.EmptyWagon" any more, and the latter starts to disappear from the inner representation.

V. Algorithm

This section introduces the proposed algorithm, as well as its analysis. The algorithm primarily exploits two observations (see Sec. IV):

- concepts that are "closer" to the target concept of the main network are better extracted from top layers;
- the concept extraction reliability with the specified extractor architecture is roughly unimodal.

Input of the algorithm consists of a trained neural network G^τ, the ontology O, the list L of G^τ layers, which activations should be used for concept extraction, the training dataset for mapping networks $\left(X, y_{RELEVANT(\tau)}\right)$, testing dataset for mapping networks $\left(X^{Test}, y^{Test}_{RELEVANT(\tau)}\right)$, and a class of mapping networks M. It is assumed, that the neural network G^τ is trained for binary classification of input samples $x \in X$ to one of the ontology classes τ, which is typically done by a sigmoid activation of the last network layer, i.e. $G^\tau(x) \in [0; 1]$. We also assume that it is possible to extract activations of a given layer, obtained during forward pass via the given network, we will denote these values as $G^\tau_{l_i}(x)$, where l_i is a layer number, $i \in \{1, \dots, |L|\}, l_i \in L$.

The ontology O consists of the set of concepts C and the set of definitions D. Each definition have the form of a triple (c, *rel*, *expr*), where c is the concept being defined, *rel* is some relation (equivalent, subclass of) and *expr* is some expression over the concepts. We also assume that it is possible to obtain the set of concepts, used in the expression *expr*, we denote the procedure to do that as `CONCEPTS`(*expr*).

Algorithm 1 Concept extraction

cs, parent := `CONCEPTS_ORDER_BFS`(τ, O)
best[τ] := $|L|$
for $c \in cs$:
  best[c] := $|L|$
for $c \in cs$:
  l_{TOP} := *best*[*parent*[c]]
  best[c] := `SEARCH`(c, l_{TOP})
return $\{(c, L[best[c]]) \mid c \in cs\}$

According to the observation, that concepts close to the target concept are located in the layers closer to the head, the algorithm starts with ordering the ontology concepts by its *definition distance* from τ, the target concept of the network G^τ. This ordering is done by `CONCEPTS_ORDER_BFS` procedure, which implements a breadth-first search in the ontology, interpreting concepts as nodes and definitions as sets of edges, connecting the concept being defined to all concepts, used in the definition. The desired side-effect of this procedure is that the

Algorithm 2 CONCEPTS_ORDER_BFS(c_0, O)

```
Input: concept c0, ontology O.
D := DEFINITIONS(O)
q := queue(∅)
output := []
parent := ∅
PUSH(q, c0)
while not EMPTY(q):
   c := POP(q)
   for c', rel, expr ∈ D:
      if c = c':
         for c'' ∈ CONCEPTS(expr):
            if c'' not in parent:
               parent[c''] := c
               output := output + c''
               PUSH(q, c'')
return output, parent
```

Algorithm 3 SEARCH_LINEAR(c, l_{TOP})

```
Input:
- concept c, best layer for which is searched for;
- l_TOP, top layer of the network to consider.
best_layer := None
best_value := None
l := l_TOP
while l ≥ 1:
   m := FIT(M, G^τ_L[l](X), y_c)
   v := EVALUATE(m, G^τ_L[l](X^Test), y_c^Test)
   if best_value is None or v > best_value:
      best_value := v
      best_layer := l
   if best_layer – l > θ:
      break
   l := l - 1
return best_layer
```

resulting list *cs* contains only concepts, relevant to τ that can be potentially aligned with the activations of G^{τ}. The concept extraction process starts with concepts close to τ and proceeds to more distant ones. CONCEPTS_ORDER_BFS also returns *parent* mapping, and *parent*[*c*] is the concept with the smallest distance from τ, that there is definition (*parent*[*c*], *rel*, *e*), $c \in$ CONCEPTS(*e*). For each examined concept *c* a search for the best layer from which *c* can be extracted is initiated. The levels are evaluated using mapping networks from the class *M,* which are fitted to the training data and then evaluated on test data using some quality metric (typically, ROC AUC). We call the quality metric value "concept expression". The range of this search is constrained by the distance from τ, in particular, the concept is searched for only in layers that are farther from the network G^{τ} head than the best matching layer for the *parent*[*c*]. There are several search strategies possible, i.e. several implementations of the SEARCH procedure. We experimented with two implementations – linear search and golden-section search. Both methods utilize the unimodality of the matching reliability of the given concept with the number of layers. Linear search starts with the layer most close to the G^{τ} head, and sequentially evaluate mappings until it finds a layer after which there is no improvement during θ layers (we call θ "patience"). This parameter is introduced, because concept expression does not follow strictly unimodal shape, therefore, larger values of θ reduce the chances of missing the global maximum. Golden-section search is an algorithm for finding maximum of a unimodal function based on divide-and-conquer technique. On each iteration, it splits all the layers into three parts and excludes one of these parts. An advantage of this algorithm is that it requires only one function evaluation per iteration (except the first iteration, which requires two) and even in worst case, the number of function evaluations is proportional to the logarithm of the number of layers. The number of function evaluations is an important criterion here, because each such evaluation requires fitting a model to the activations of the layer and calculating the quality metric (e.g., ROC AUC), which typically takes dozens of minutes. In spite the golden-section search has better asymptotical properties, in practice, for this problem linear search is as efficient, first, because maximum is typically located within 2-3 layers from the start of the range, second, because the concept expression follows not strictly unimodal shape, and golden-section is more likely to miss the global maximum.

Let's illustrate the proposed algorithm using concept extraction scenario discussed in Sec. IV. Main network is a

i	*L* [i]	WarTrain	EmptyTrain	ReinforcedCar	PassengerCar	EmptyWagon
1	2	0,5000	0,5000	0,5000	0,5000	0,5000
2	4	0,5000	0,5000	0,5000	0,5000	0,5000
3	7	0,6225	0,7444	0,7244	0,5723	0,6063
4	9	0,9508	0,9718	0,9984	0,9927	0,9619
5	12	0,9380	0,9769	0,9893	0,9855	0,9157
6	14	0,9911	**1,0000**	**1,0000**	0,9982	**0,9884**
7	17	0,9866	0,9998	0,9999	0,9978	0,9537
8	19	0,9991	**1,0000**	0,9997	0,9981	0,6907
9	20	**0,9996**	**1,0000**	0,9996	**0,9989**	0,6583
10	21	0,9993	0,9999	0,9995	0,9985	0,6684

Fig 5. Results of concept localization by the algorithm

neural network of ResNet-18 architecture, trained for binary classification of concept TypeA, which, in this case, is the target concept (τ). Following the data, presented in Sec. IV we will search for ontology concepts in only subset of ResNet-18 layers, in particular, L = [2, 4, 7, 9, 12, 14, 17, 19, 20, 21] (these are outputs of skip-connection structures and all fully-connected layers). Relevant ontology fragment is specified in Fig. 2. Class of mapping networks M can be set as shallow networks with sigmoid activation.

Concept ordering using `CONCEPTS_ORDER_BFS` results in the following order (parent concept is shown in square brackets):

- WarTrain [TypeA]
- EmptyTrain [TypeA]
- ∃has.ReinforcedCar [WarTrain]
- ∃has.PassengerCar [WarTrain]
- ∀has.(EmptyWagon ⊔ Locomotive) [EmptyTrain]
- ∃has.EmptyWagon [EmptyTrain]

In this order, the concepts are examined to find the best concept expression. If for this purpose we use linear search (`SEARCH_LINEAR`, Algorithm 3) with patience value 1, then the set of analyzed layers for each concept would be as shown in Fig. 5. Bolded cells are the layers where the concept is expressed best of all, cell in ellipse show the layer identified by the search procedure, while rectangles highlight examined layer ranges for each concept. Note, that ranges for ∃has.ReinforcedCar, ∃has.PassengerCar and ∃has.EmptyWagon start from the 9-th position in the list L (corresponding to 20-th layer of the main net), because "parent" concepts for these concepts are best expressed in this layer.

In this example, the algorithm was able to find the layers where the concepts are best of all expressed by examining about a half (23 out of 50) of possible concept-layer pairs. With larger main nets architectures containing more layers (e.g., ResNet-52, MobileNet-V2, EfficientNet, etc.) the advantage is even more impressive. Although the algorithm is based on a heuristic, in our experiments, we did not find situations, where it fails to find best concepts expressions (with reasonable patience setting). However, this situation is possible, and it requires further research.

VI. Conclusion

We have proposed a heuristic algorithm for efficient ontology concept extraction from deep neural networks. The layers identified with the algorithm and the respective mapping networks can be used to generate ontology-based post-hoc explanations for some classification neural networks, which can improve human-AI interaction.

Although the algorithm successfully solves the problem, it has several limitations that should be addressed in further work. For example, currently we consider only concepts of the same class. In the provided example, target concept (TypeA) and extracted concepts (e.g., EmptyTrain or ∃has.ReinforcedCar) are all subclasses of Train concept (it is the most general concept related to each image). Other ontology concepts can also be considered. It would also be interesting to consider sets of neurons spread across several layers. Another important but challenging direction for future research is theoretical investigation of performance guarantees of the algorithm.

Acknowledgment

The research is funded by the Russian Science Foundation (project 22-11-00214).

References

[1] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, 2021.

[2] R. Confalonieri, T. Weyde, T. R. Besold, and F. Moscoso Del Prado Martín, "Trepan reloaded: A knowledge-driven approach to explaining black-box models," *Frontiers in Artificial Intelligence and Applications*, vol. 325, pp. 2457–2464, 2020.

[3] R. Confalonieri, T. Weyde, T. R. Besold, and F. M. del P. Martín, "An Ontology-based Approach to Explaining Artificial Neural Networks," 2019.

[4] R. Confalonieri, T. Weyde, T. R. Besold, and F. Moscoso del Prado Martín, "Using ontologies to enhance human understandability of global post-hoc explanations of black-box models," *Artificial Intelligence*, vol. 296, p. 103471, 2021.

[5] C. Panigutti, A. Perotti, and D. Pedreschi, "Doctor XAI: An ontology-based approach to black-box sequential data classification explanations," in *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 629–639.

[6] G. Bourguin, A. Lewandowski, M. Bouneffa, A. Ahmad, and T. Ontologically, "Towards Ontologically Explainable Classifiers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. LNCS 12892, 2021, pp. 472–484.

[7] Z. A. Daniels, L. D. Frank, C. Menart, M. Raymer, and P. Hitzler, "A framework for explainable deep neural models using external knowledge graphs," in *Proc. SPIE 11413, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*, 2020, p. 73.

[8] V. Bourgeais, F. Zehraoui, M. Ben Hamdoune, and B. Hanczar, "Deep GONet: self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data," *BMC Bioinformatics*, vol. 22, pp. 1–24, 2021.

[9] M. de Sousa Ribeiro and J. Leite, "Aligning Artificial Neural Networks and Ontologies towards Explainable AI," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 6, pp. 4932–4940.

[10] J. Voogd, P. de Heer, K. Veltman, P. Hanckmann, and J. van Lith, "Using Relational Concept Networks for Explainable Decision Support," in *3rd IFIP Cross Domain Conference for Machine Learning and Knowledge Extraction, CD-MAKE 2019, LNC 11713*, 2019, pp. 78–93.

[11] H. A. Abbass, S. Elsawah, E. Petraki, and R. Hunjet, "Machine Education: Designing semantically ordered and ontologically guided modular neural networks," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 948–955.

[12] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, May 2019.

[13] T. S. Perdih, N. Lavrac, and B. Skrlj, "Semantic Reasoning from Model-Agnostic Explanations," in *SAMI 2021 - IEEE 19th World Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 2021, pp. 105–110.

[14] M. de Sousa Ribeiro, L. Krippahl, and J. Leite, "Explainable Abstract Trains Dataset," Dec. 2020.