ISSN 1064-2307, Journal of Computer and Systems Sciences International, 2021, Vol. 60, No. 6, pp. 966–980. © Pleiades Publishing, Ltd., 2021.
Russian Text © The Author(s), 2021, published in Izvestiya Akademii Nauk, Teoriya i Sistemy Upravleniya, 2021, No. 6, pp. 106–121.

ARTIFICIAL INTELLIGENCE

Review of Research in the Field of Developing Methods to Extract Rules From Artificial Neural Networks

A. N. Averkin[a,b,*] and S. A. Yarushev[b,]**

[a] *Federal Research Center "Computer Science and Control," Russian Academy of Sciences, Moscow, Russia*

[b] *Plekhanov Russian University of Economics, Moscow, Russia*

**e-mail: averkin2003@inbox.ru*

***e-mail: sergey.yarushev@icloud.com*

Received June 29, 2021; revised July 3, 2021; accepted July 26, 2021

Abstract—A large-scale review and analysis of the existing methods and approaches to extract rules from artificial neural networks, including deep learning neural networks, is carried out. A wide range of methods and approaches to extract rules and related approaches to develop explainable artificial intelligence (AI) systems are considered. The taxonomy and several directions in studies of explainable neural networks related to the extraction of rules from neural networks, which allow the user to get an idea of how the neural network uses the input data, and also, using rules, to reveal the hidden relationships of the input data and the results found, are explored. This review focuses on the relationship of the most common rule-based explanation systems in AI with the most powerful machine learning algorithms using neural networks. In addition to rule extraction, other methods of constructing explainable AI systems are considered based on the construction of special modules that interpret each step of changing the neural network's weights. A comprehensive analysis of the existing research makes it possible to draw conclusions about the appropriateness of using certain approaches. The results of the analysis will allow us to get a detailed picture of the state of research in this area and create our own applications based on neural networks, the results of which can be studied in detail and their reliability evaluated. The development of such systems is necessary for the development of the digital economy in Russia and the creation of applications that allow making responsible and explainable management decisions in critical areas of the national economy.

DOI: 10.1134/S1064230721060046

INTRODUCTION

The large-scale development of artificial intelligence (AI) systems, including applications based on artificial neural networks, opens up the broadest opportunities for their use in various fields, from emotion recognition systems to predictive analytic systems, medical, and military applications. At the same time, existing systems and applications have one common significant drawback: the impossibility of interpreting the results obtained and decisions made. The well-known problem of the so-called black box imposes significant restrictions on the use of such systems, including legislative ones, since it is impossible to trace the decision-making process by a neural network.

These problems are currently being addressed in the explainable artificial intelligence (XAI) direction. Explainable AI systems help the user understand the decisions made using machine learning methods, which increases the confidence in these systems and enables them to make more effective decisions based on the results of the system. All this allows developers and users to investigate the factors that are used by the neural network in solving a specific problem and understand which parameters of the neural network need to be changed in order to improve the accuracy of its work.

In addition, the study of how neural networks extract, store, and transform knowledge may be useful for the future development of AI techniques. For example, increasing the explainability of artificial neural networks will allow detecting the so-called hidden dependences that are not present in the input data, but appear as a result of their integration into the neural network. Methods for extracting rules from neural networks are one of the connecting elements between symbolic and connectionist models of knowledge representation in AI.

1. THE HISTORY OF EXPLAINABLE AI MODELS

Research in this area can be divided into three stages: in the first stage (starting from 1970) expert systems were developed; in the second stage (mid-1980s), the transition was made from expert systems to knowledge-based systems; and in the third phase (since 2010), deep architectures of artificial neural networks, which required new global research on the construction of explainable systems, have been studied.

We take a closer look at each of these stages.

1.1. First Step. Expert Systems

Rule-based explanation systems. The first and second generations of explainable AI are related to expert systems involving decision-making and diagnosis. The main stage in the development of expert systems came in the early 1970s and contained representations based on knowledge, as well as the use of rules and relationships. The systems had a Q&A toolkit for users and could provide rule-based recommendations and diagnoses. Studies have shown that explaining how computer systems work improves people's confidence in recommender systems [1]. When interpreting the results of the work of the expert system, the explanation of the principle of operation of the system and the rationale for choosing the architecture of the system also differed [2].

At the first stage, in the era of the development of expert explanatory systems, large-scale studies were carried out demonstrating that expert systems with an explanatory component are able to exert a greater influence on decision-making, and the effect was directly proportional to the user's skill level [3–6].

However, many of the first generation expert systems did not deliver the expected benefits. When working on medical expert systems, researchers recognized that physicians would ignore expert system recommendations unless a rationale (justification) was provided for why the system made such recommendations. The initial explanation systems attempted to provide this rationale by describing the main goals and steps used to make a diagnosis. This approach, which Swartout and Moore called "Resume as an explanation myth" [7], also did not completely suit users, and this led to a rethinking of the approaches, goals, and objectives of XAI systems. The first generation appeared in the late 1970s [8] and existed for about 10 years.

Some imperfection of expert systems of the first generation, in turn, gave rise to the first generation of systems that include an explanation block as an obligatory element, for example, Mycin [9] and systems related to it, such as Digitalis Therapy Advisor [10], BLAH [11] and other special purpose explanation systems. Using inference trees, these systems worked by creating logical and probabilistic rules for diagnosing or answering questions. In general, since knowledge and experience were formulated in terms of rules, these rules were described in natural language. A simple explanation was expressed as a rule for making a decision. Such explanations were usually written in a limited natural language, but they were often simply the use of "if–then" production rules for textual descriptions.

In [12], first-generation systems are considered that created explanations by rephrasing the rules for making a decision. In general, expert systems and systems of explanation of the first generation focused on describing the internal states of an intellectual system, as well as its goals and plans. Sometimes it was quite simple, because the rules themselves were a formalization of the rules used by the experts. Sometimes, however, linguistic descriptions bore little resemblance to human explanation or natural language at all. First, they relied only on the format of logical and causal if–then rules and not on providing explanations at a higher level of the reasoning strategy (for example, collecting basic information about a patient, creating a network for alternative explanations, and trying to support a specific hypothesis). Second, some of the rules were logically necessary for the system to work, not necessarily meaningful to users, and not related to the user's standard "how" and "why" questions. Third, domain knowledge was sometimes compiled (for example, causal relationships between symptoms and diagnoses), so it was not included in the explanation.

When Mycin was originally developed, the inability to fully explain the set of rules and their rationale was not considered a flaw, because creating any explanations in a readable form from the written reasoning tree of the program was already a difficult task and a significant advance in the field of AI. Moreover, such associative models were rules of thumb based on demographic and physiological data that were usually familiar to users. The latter were supposed to simply follow the advice of the program after it had been tested and certified by experts. However, attempts to expand or refine these rule sets, use them for learning, or explain high-level associations have been unsuccessful.

1.2. Second Phase. Knowledge-Based Explanatory Systems

By the mid-1980s, the limitations of first-generation explanatory systems faced the problem that it was not enough to simply summarize the inner workings of the system. The generated text could be correct, but it was not necessarily what the user wanted to know or did not understand. Second-generation systems often blurred the line between a consulting program [12], a mentor [13], an advisor [14], and a data entry system [15], but researchers were challenged to go beyond methods that did not improve the understanding or acceptance of expert systems. The main driving force behind second generation systems was the need to develop more abstract structures that would facilitate reuse and ease of system design. NEOMYCIN did this by providing a task- and meta-rule-based diagnostic procedure and a related taxonomic and causal language [16, 17].

Early in the development of explanatory expert systems, it was recognized that their knowledge base, rules, and explanations could be used to create intelligent mentors. A distinctive feature of intelligent learning systems is that they derive a mental model of the subject area of each student (his knowledge base) based on the student's behavior. These systems were first collectively called computational analytics and intelligence (CAI) systems to distinguish intelligent mentors from simple learning machines of the 1960s, and were adopted by the AI community in education, especially in the 1980s and early 1990s years.

For example, in the GUIDON expert system, a student's request for patient data and his stated hypotheses were used to search back through the MYCIN rule network (or any other network) to determine which rules were not taken into account. The explanation was a network of inferences and sometimes included ambiguities (for example, evidence that a student knows a rule based on previous interactions should be distinguished from its application in a specific case). This is why systems were sometimes called *knowledge-based teachers*, as opposed to learning machines in the 1960s. The main idea was that the subject knowledge of the expert system (for example, the rules of medical diagnostics) was separated from the knowledge base of the training (for example, the rules for managing the dialog). In addition, the process of interpreting domain rules, similar to explaining the behavior of an expert system based on a model (tracing) of its internal processes, was used to explain student behavior by building a model of how the student reasoned.

1.3. Stage Three. New Methods of Explanation Based on Pattern Recognition and Word Processing

After some stagnation related to the slowdown in the development of the direction of knowledge-based systems, the third stage, related to 2017–2021 and reflected in Section 4 of this article, which describes the DARPA Explainable AI program, appeared.

The emergence of the third stage can be clearly demonstrated by analyzing the number of publications on the topics of rule extraction and explainable AI. When analyzing the number of publications over the past 20 years, it can be noted that interest in extracting rules from neural networks has remained more-or-less stable (according to some sources, even periodic, which is related to the three considered generations of explainable AI systems) and increases only with the advent of the third generation of explainable AI systems (Fig. 1), which is mainly related to the creation of the DARPA program.

With the advent of the DARPA program, we can observe explosive growth in the number of publications directly on the topic of explainable AI (since 2018) and its further growth, as shown in Fig. 2.

In systems of the third generation, as in systems of the first generation, attempts were made to explain the inner workings of the system, which in itself remains a serious problem. First generation systems built expert knowledge into rules, often obtained directly from experts, and attempted to construct language descriptions based on expert judgment. These rules have often been transformed into natural language expressions, and a large part of this study has focused on building knowledge representation systems. In systems of the third generation, this task turned out to be much more difficult. The disadvantages of first generation systems related to a poor level of detail and incomprehensible language can become a problem for third generation systems. Since the first generation of systems, computer technology in data visualization, animation, video, etc., has advanced significantly, and many new ideas have been proposed as potential methods for generating explanations. Although the first generation systems supported natural language dialogs and interactivity in question and answer systems, modern systems do this at a higher level than expert systems. An example is the use of reasoned explanatory human-machine dialogs to eliminate inconsistencies in knowledge bases in the process of acquiring knowledge using software for expert systems [18].

Some of the explanations presented in the first generation systems were easier to create than the explanations in the third generation, because they were a direct repetition of hand-coded rules. The current generation of systems may find it harder to provide the simple explanations that first generation systems

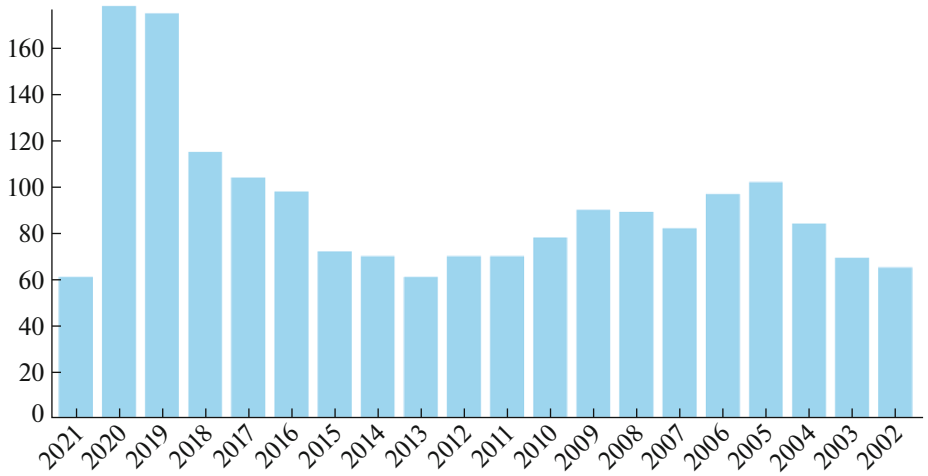

Fig. 1. Number of publications by year in the field of rule extraction from neural networks.

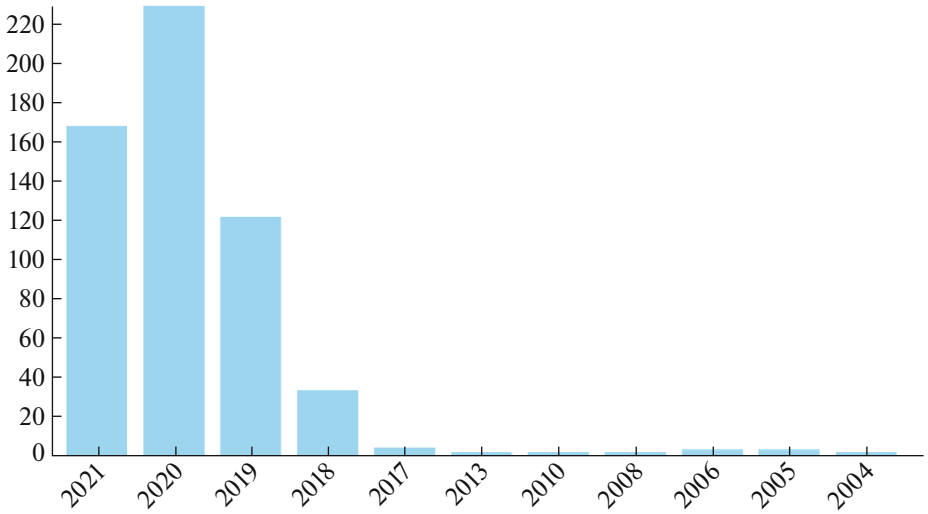

Fig. 2. The number of publications by year on the topic explained by AI (XAI).

produce. However, in a sense, the third generation systems currently being developed reflect the achievements of the first generation systems. Many of these systems are expected to encounter problems similar to those experienced by the first generation systems. Importantly, they can be solved by methods that originated in second generation systems.

2. REVIEW OF RESEARCH ON RULE EXTRACTION FROM ARTIFICIAL NEURAL NETWORKS

Increasing the transparency of neural networks by extracting rules from them has two main advantages. This gives the user some insight into how the neural network uses input variables to make decisions, and allows the hidden features in the neural networks to be revealed when rules are applied to explain how individual neurons work. Identifying critical attributes or causes of neural network errors can be part of the understanding. In an effort to make opaque neural networks more understandable, rule extraction techniques are bridging the gap between precision and clarity [19, 20].

In order for, for example, a neural network to be used in critical applications, for example, in aviation or electric power, an explanation is required not only of the principles of its operation but also of how the neural network obtained the result. In these cases, it is extremely important that the user of the

system has the opportunity to check the output of the artificial neural network under all possible input conditions [21].

To formalize the task of extracting rules from a neural network, the following definition can be used: Given the parameters of the trained neural network and the data on which it was trained, create a description of the neural network hypothesis that is understandable, but close to the behavior of the specified network.

To distinguish between different approaches to rule extraction from neural networks, a multidimensional taxonomy was introduced in [21]. The first dimension it describes is the cardinality of the extracted rules (for example, IF–THEN rules or fuzzy production rules). The second dimension is called transparency and describes the strategy behind the rule extraction algorithm. If a method uses a neural network only as a black box, regardless of the architecture of the neural network, we call it a pedagogical approach. If instead the algorithm takes into account the internal structure of the neural network, we call this approach decomposition. If an algorithm uses components of both pedagogical and decomposition methods, then this approach is called eclectic. The third dimension is the quality of the extracted rules. Since quality is a broad term, it is divided into several criteria: quality, accuracy, consistency, and intelligibility. While quality measures the ability to correctly classify previously unseen examples, accuracy measures the degree to which rules mimic the behavior of a neural network [19]. Accuracy can be thought of as accuracy in relation to the output of the neural network. Consistency can only be measured when the rule extraction algorithm involves training a neural network instead of processing an already trained neural network. The extracted rule set is considered consistent when the neural network generates rule sets that correctly classify test data for different training epochs. Comprehensibility is seen as a measure of the size of the rules, that is, short rules with fewer rules are considered more comprehensible.

In this study, we will focus on only three described criteria. In accordance with [22], we focus on methods that do not impose special requirements on how the neural network was trained before the rules were extracted. In addition, only algorithms capable of extracting rules from feedforward neural networks are analyzed, regardless of any other characteristics of the architecture. In accordance with [23], it is necessary that the algorithm has the greatest degree of generality.

Let us analyze some methods for extracting rules that meet these characteristics. Let us start with the decomposition approach. As mentioned earlier, decomposition approaches for extracting rules from neural networks operate at the neuronal level. Typically, the decomposition approach analyzes each neuron and generates rules that mimic the behavior of that neuron. We consider below the layer-by-layer CT rule extraction algorithm, the Tsukimoto polynomial algorithm, and the rule extractor via decision tree induction.

The CT algorithm was one of the first approaches for extracting rules from neural networks [24]. It describes each neuron (layer by layer) with IF–THEN rules by heuristically looking for combinations of input attributes that exceed the neuron's threshold. The recording module is used to retrieve rules that relate to the original attributes of the input, not the outputs of the previous level. To find suitable combinations, the CT method uses a tree search, that is, the rule (represented as a node in the tree) at this level generates its start nodes by adding an additional available attribute. In addition, the algorithm uses a number of heuristics to stop the tree from growing in situations where further improvement is not possible.

Tsukimoto's polynomial algorithm for extracting rules from a neural network is very similar to the CT method. It uses a multilevel decomposition algorithm to extract IF–THEN rules for each neuron, and also tracks the strategy for finding input configurations that exceed the neuron's threshold. The main advantage of Tsukimoto's algorithm is its computational complexity, which is polynomial, while the CT method is exponential. The algorithm achieves polynomial complexity by finding appropriate terms using the space of multidimensional functions. In the second step, these terms are used to create IF–THEN rules. The training data is then used to improve the accuracy of the rules. In the last step, Tsukimoto's algorithm tries to optimize intelligibility by removing irrelevant attributes from the rules.

Another method of rule extraction by induction of a decision tree was introduced in [25]. Their CRED algorithm transforms each output unit of the neural network into a solution where the tree nodes are tested against the hidden layer nodes and the leaves represent the class. After this step, the intermediate rules are retrieved. Then, for each branch point used in these rules, a different decision tree is created using the branch point on the input layer of the neural network. In new trees, the leaves do not directly select the class. Extracting the rules from the second decision tree leads us to describe the state of hidden neurons, consisting of input variables. As a final step, the intermediate rules describing the output layer through the hidden layer are replaced with those that describe the hidden layer based on the inputs of the neural network. They are then combined to construct rules describing the output of the neural network based on its input.

Table 1. Algorithms for extracting rules from neural networks

Algorithm	Network type used	Algorithm type	Retrieved rule type
DIFACON-miner	Multilayer perceptron	Decomposition	IF–THEN
CRED	Too	>>	Decision tree
FERNN	>>	>>	M-of-N, IF–THEN
CT	>>	>>	IF–THEN
Tsukamoto's algorithm	Multilayer perceptron, recurrent neural network	>>	IF–THEN
TREPAN	Multilayer perceptron	Pedagogical	M-of-N, decision tree
HYPINV	>>	>>	Hyperplane rule
BIO-RE	>>	>>	Binary rule
KDRuleEX	>>	>>	Decision tree
RxREN	>>	>>	IF–THEN
ANN-DT	>>	>>	Binary decision tree
RX	>>	Eclectic	IF–THEN
Kahramanli and Allahverdi's algorithm	>>	>>	IF–THEN
DeepRED	Deep neural network	Decomposition	IF–THEN

Pedagogical approaches do not take into account the internal structure of the neural network. The goal of pedagogical approaches is to consider trained neural networks as an integral object or as a black box [26]. The main idea is to extract the rules by directly matching the input data with the output data [27].

Pedagogical approaches work with a neural network as a function. This function sets the output of the neural network for an arbitrary input, but does not provide an understanding of the internal structure of the neural network or its weights. For a neural network, this class of algorithms tries to find a relationship between the possible input variations and outputs generated by the neural network, with some using predefined training data and some not.

Rule extraction based on interval analysis uses confidence interval analysis (VIA) to extract rules that mimic the behavior of neural networks. The main idea of this method is to find the input intervals in which the output signal of the neural network is stable, that is, the predicted class is the same for slightly changing input configurations. As a result, the analysis of confidence intervals provides the base for reliable correct rules.

Rule retrieval using sampling consists of several methods that follow more or less the same strategy for extracting rules from a neural network using sampling, that is, they create a vast dataset as the base for rule retrieval. After that, the selected dataset is passed to the standard learning algorithm to generate rules that simulate the behavior of the network. In [28], it was proved that the use of sampled data is more efficient than using only training data in rule extraction problems.

One of the first methods to follow this strategy was Trepan's algorithm [29]. It works in a similar way to the divide-and-conquer algorithm, looking for bifurcations in the training data for individual instances of different classes. The main differences from divide and conquer are the best strategy for expanding the tree structure, additional branch points, and the ability to select additional training examples at deeper nodes in the tree. As a result, the algorithm also creates a decision tree, which, however, can be transformed into a set of rules if necessary.

The binary input and output rule extraction (BIO-RE) algorithm is only capable of processing a neural network with binary or binarized input attributes. BIO-RE creates all possible combinations of input data and requests them from the neural network. Using the neural network output, a truth table 1 is created for each example. It is also easy to move from the truth table to rules, if necessary.

ANN-DT is another decision-based sampling technique for describing the behavior of a neural network. The general algorithm is based on the CART algorithm with some variations in the original implementation. ANN-DT uses a scaling-up sampling technique to ensure that the majority of the training sample is representative. This is achieved using the nearest neighbor method, which calculates the distance from the sample point to the closest point in the training dataset and compares it to the reference value.

The idea of creating a large set of examples at the first stage is also implemented by the STARE algorithm. Like BIO-RE, STARE generates large truth tables for training. The advantage of STARE lies in its ability to not only handle binary and discrete attributes but also to handle continuous inputs. To generate truth tables, the algorithm rebuilds the input data. An example of a pedagogical approach using a training set is KDRuleEx. Similarly to the Trepan algorithm, this algorithm also generates additional training examples when there is too little data for the next split points. KDRuleEx uses a genetic algorithm to create new learning examples. This technique results in a decision table that can be converted to IF–THEN rules, for example.

An eclectic approach is a set of rule extraction methods that include elements of both the pedagogical and decomposition approaches. In particular, the eclectic approach uses knowledge about the internal architecture and the weight vectors in the neural network in addition to the symbolic learning algorithm.

The fast rule search approach in a neural network includes the FERNN approach, which first tries to identify the corresponding hidden neurons and the inputs to the network. For this step, a decision tree is built using the C4.5 algorithm. The rule extraction process results in the generation of M-of-N and IF–THEN rules. With a set of correctly classified training examples, FERNN analyzes the activation values of each hidden vertex for which the activation values are sorted in ascending order. The C4.5 algorithm is then used to find the best split point to form the decision tree. The table below shows a comparison of algorithms for extracting rules from neural networks by type of neural network, type of algorithm, and type of extracted rule [30].

3. NEURAL FUZZY MODELS IN THE TASKS OF EXTRACTING RULES FROM ARTIFICIAL NEURAL NETWORKS

The most interesting part of this study is rule extraction using neuro-fuzzy models. Fuzzy rule-based systems (FRBS) developed with fuzzy logic have become a field of active research over the past few years. These algorithms have proven their strengths in tasks such as managing complex systems and creating fuzzy controls. The relationship between both artificial neural networks (ANNs) and FRBS approaches has been carefully studied and shown to be equivalent. This leads to two important conclusions. First, we can apply what was found for one of the models to another. Second, we can translate the knowledge embedded in the neural network into a more cognitively acceptable language: fuzzy rules. In other words, we get a semantic interpretation of neural networks [31–33].

In order to get a semantic interpretation of the deep learning black box, neural networks can be used instead of the last fully connected layer. For example, the adaptive neural fuzzy interference system (ANFIS) is a multilayer feedforward network. This architecture has five layers such as a fuzzy layer, a production layer, a normalization layer, a defuzzification layer, and an output layer. ANFIS combines the advantages of a neural network and fuzzy logic. Below we have given a classification of the most well-known neuro-fuzzy approaches.

Considering the architecture of neural fuzzy models, three methods of combining ANNs and fuzzy models can be distinguished [34, 35]:

—neuro-FIS, in which ANN is used as a tool in fuzzy models;

—fuzzy ANNs, in which the classical ANN models are fuzzified;

—neuro-fuzzy hybrid systems, in which fuzzy systems and ANN are combined into hybrid systems [36, 37].

Based on these techniques, neuro-fuzzy models can be divided into three classes [38–40].

Cooperative neuro-fuzzy models. In this case, part of the ANN is initially used to define fuzzy sets and/or fuzzy rules, where only the resulting fuzzy system is subsequently executed. In the learning process, membership functions are determined, and fuzzy rules are formed based on the training sample. Here the main task of the neural network is to select the parameters of the fuzzy system.

Parallel neuro-fuzzy models. The neural network in this type of model works in parallel with the fuzzy system, providing input to the fuzzy system or changing the output of the fuzzy system. The neural network can also be a postprocessor of the output data from a fuzzy system.

Hybrid neuro-fuzzy models. The fuzzy system uses a training method, as does the ANN, to adjust its parameters based on the training data. Among the presented classes of models, the models of this particular class are most popular, as indicated by their application in a wide range of real problems [41–44].

The most popular hybrid models include the following architectures.

The fuzzy adaptive learning control network (FALCON) [45], which has a five-layer architecture. There are two linguistic nodes per one output variable. The first node works with a training sample (training pattern) and the second node is the input for the entire system. The first hidden layer labels the input

sample according to membership functions. The second layer defines the rules and their parameters. Training takes place based on a hybrid unsupervised algorithm to determine the membership function and the rule base, and it uses the gradient descent algorithm to optimize and select the final parameters of the membership function.

ANFIS [46] is a well-known neuro-fuzzy model that has been used in many applications and research areas [47]. Moreover, a comparison of the architectures of neural fuzzy networks showed that ANFIS shows the minimum error in the prediction problem. The main disadvantage of the ANFIS model is that it imposes serious requirements on computing power [48].

The system of generalized approximate intelligent control based on reasoning (GARIC) [49] is a neuro-fuzzy system using two neural network modules, an action selection module and a state assessment module, which is responsible for assessing the quality of the action selection by the previous module. GARIC is a five-layer feedforward network.

The neural fuzzy controller (NEFCON) [50] was developed to implement the Mamdani-type fuzzy inference system. Links are defined using fuzzy rules. The input layer is a fuzzifier, and the output layer solves the defuzzification problem. A network is trained based on a hybrid reinforcement learning algorithm and an error backpropagation algorithm.

The fuzzy inference and neural network system in fuzzy inference software (FINEST) [51] is a parameter setting system. The fuzzy predicates, implication function, and combinatorial function are tuned.

A system for automatically constructing a neural network of fuzzy inference (SONFIN) [52] is similar to the NEFCON controller, but instead of implementing fuzzy inference of the Mamdani type, it implements the Takagi–Sugeno inference. In this network, the input sample is processed using the aligned clustering algorithm. When identifying the structure of the precondition part, the input space is divided in a flexible manner according to an algorithm based on aligned clustering. The system parameters are partially tuned using the least squares method and the preconditions are tuned using the backpropagation method.

Dynamically developing fuzzy neural network (dmEfuNN) and (EFuNN) [53]. In EFuNN, all nodes are created in the learning process. The first layer passes the training data to the second, which calculates the degree of fit with a predefined membership function. The third layer contains sets of fuzzy rules, which are prototypes of input-output data, which can be represented as hyperspheres of fuzzy input and output spaces. The fourth layer calculates the degree to which the output membership function has labeled the input data, and the fifth layer defuzzifies and calculates the numerical values of the output variable. DmEfuNN is a modified version of EFuNN. The main idea is that for all input vectors the set of rules is dynamically selected, the activation values of which are used to calculate the dynamic parameters of the output function. While EFuNN implements Mamdani-type fuzzy rules, dmEFuNN uses the Takagi–Sugeno type.

4. REVIEW OF APPROACHES TO THE DEVELOPMENT OF EXPLAINABLE AI SYSTEMS

We now explore different models of explainable AI. Almost all of them are related to third-generation explanatory systems and the DARPA program, which began in 2018 [54]. The DARPA explainable AI (XAI) program seeks to create AI systems whose learning models and solutions can be understood and properly validated by end users. Achieving this goal requires methods for constructing more explicable models, developing effective explicable interfaces, and understanding the psychological requirements for an effective explanation. Explainable AI is needed for users to understand, properly trust, and effectively manage their smart partners. DARPA sees XAIs as AI systems that can explain their decision to a human user, characterize their strengths and weaknesses, and how they will behave in the future. DARPA's goal is to create more human-readable AI systems through effective explanations. XAI development teams solve the first two problems by creating and developing Explainable Machine Learning (ML) technologies, developing principles, strategies and methods of human-computer interaction to generate effective explanations. Another XAI development team tackles the third challenge by combining, extending, and applying psychological explanatory theories that the development teams will use to test their systems. The development teams evaluate how a clear explanation of XAI systems improves the user experience, confidence, and productivity.

Russia is also paying attention to the direction of explainable AI. Thus, in 2020, Nizhny Novgorod State University won the competition of large scientific projects from the Ministry of Education and Science of the Russian Federation with the project "Reliable and logically transparent AI: technology, verification, and application for socially significant and infectious diseases" [55]. The main result of the project should be the development of new methods and technologies that allow us to overcome the two main

barriers of machine learning and AI systems: the problem of errors and the problem of explicitly explaining solutions. The project manager, Professor Alexander Gorban, explained the main idea of the project as follows: "These issues are closely related: without the ability to read logically, AI errors will remain unexplained. Additional training of the system within the framework of existing methods can damage existing skills and, on the other hand, can require huge resources, which is impractical in serious tasks. For example, the well-known cognitive computing system IBM Watson has failed in the personalized medicine market due to systematic errors in diagnosing and recommending cancer treatment, the sources of which could not be found and eliminated."

The following is a brief description of the explainable AI models and research centers that are doing this research in the framework of the DARPA Explainable AI program [54].

1. Deep Explainable AI (DEXAI) at the University of California, Berkeley (UCB). The UCB team, including researchers from Boston University, University of Amsterdam, and Kitware, is developing a human-readable AI system through explicit structural interpretation and introspective explanation, which has predictable behavior and provides an appropriate degree of trust [56]. The key issues of deep explainable AI (DEXAI) consist of generating accurate explanations of the model's behavior and chooseing those that are most useful to the user. UCB addresses the first problem by creating implicit or explicit explanation models: they can implicitly represent complex hidden representations in understandable ways, or they can build explicit structures that are inherently understandable. These DEXAI models create a set of possible explanatory actions. For the second problem, UCB proposes rational explanations that use the user's belief model in deciding which explanatory actions to take. UCB is also creating an explanation interface based on these innovations and principles of interactive development. Autonomous DEXAI models are used to drive vehicles (using the Berkeley Deep Drive dataset and the CARLA simulator) [57] and in strategic game scenarios (StarCraft II). For analytic data, DEXAI uses Visual Question Answers (VQA) and filtering techniques (for example, using large datasets such as VQA-X and ACT-X for VQA and activity recognition tasks), xView, and Distinct [58].

2. Causal Models to Explain Machine Learning (CRA). The aim of the Charles River Analytics Team (CRA) (including researchers at the University of Massachusetts and Brown University) is to create and provide causal explanations for machine learning using causal models (CAMEL). CAMEL explanations are presented to the user as stories in an interactive and intuitive interface. CAMEL includes a causal probabilistic programming framework that integrates concepts and teaching methods from causal modeling [59] with probabilistic programming languages [60]. The generative probabilistic models, presented in the language of probabilistic programming, naturally express cause-and-effect relationships; they are well suited for the task of explaining machine learning systems. CAMEL examines the internal representation of a machine learning system to reveal how it represents user-defined concepts of the natural domain. Then it builds a causal model of their impact on the operation of the machine learning system, conducting experiments in which areas of agreement are systematically included or removed. After learning, it uses causal models to derive explanations for the predictions or actions of the system. In the area of data analysis, CAMEL addresses the problem of detecting pedestrians (using the INRIA pedestrian dataset) [61], and CRA is working on activity recognition problems (using ActivityNet). CAMEL's autonomy property is demonstrated in the Atari Amidar game, and the CRA works in StarCraft II.

3. Explore and communicate explainable views for analytics and autonomy. A team at the University of California Los Angeles (UCLA) (with researchers from Oregon State University and Michigan State University) is developing interpretable models that combine representational paradigms, including interpreted deep neural networks, compositional graphical AND/OR graphs, and models that produce explanations on three levels (compositionality, causality, and utility). The UCLA system contains an execution module that performs tasks with multimodal inputs, and an explain module that explains its perception, cognitive reasoning, and decisions to the user. The runtime module outputs interpreted representations in the form of a spatial, temporal and causal analysis graph (STC-PG) for 3D scene perception (for analytics) and task scheduling (for autonomy). STC-PG are compositional, probabilistic, interpretable, and based on the principles of deep neural networks, and they are used for image and video analysis. The explain module displays the explanatory syntax graph in the form of a dialog [62], localizes the corresponding subgraph in the STC-PG, and determines the user's intentions. UCLA covers both XAI problem areas using a common presentation and output structure. In the field of data analysis, UCLA demonstrated its system using a network of video cameras for understanding a scenario and event analysis. UCLA's autonomy is shown in scenarios using robots performing tasks on virtual reality platforms with realistic physics, and in an autonomous vehicle driving game.

4. Testing deep adaptive programs with informed information. Oregon State University (OSU) is developing tools to explain the actions of trained agents that perform consistent decision-making and

determine the best principles for developing user interfaces with explanations. The OSU explainable agent model uses explainable deep adaptive programming (xDAP), which combines adaptive programming, deep reinforcement learning (RL), and explainability. With xDAP, programmers can create agents that represent solutions that are automatically optimized through deep RL when interacting with the simulator. For each point of choice, deep RL connects a trained deep decision neural network (dNN) that can provide high performance but is inherently not explainable. After the initial xDAP training, the xACT program trains an explanatory neural network [63] for each dNN. They provide a sparse set of explain functions (x-functions) that encode the properties of the dNN decision logic. Such x-functions, which are neural networks, are not originally interpreted by humans. To solve this problem, xACT allows domain experts to attach interpretable descriptions to x-functions; and xDAP programmers, to annotate environment reward types and other concepts that are automatically built into dNNs as "annotation concepts" during training.

The OSU explanation user interface allows users to navigate through thousands of agent solutions and get visual and natural language explanations. Its design is based on the theory of information gathering, which allows the user to efficiently navigate to the most useful explanatory information at any time. OSU tackles the issue of autonomy and has demonstrated xACT in scripts using a custom RTS game engine. Pilot studies have provided information to explain user interface design by describing how users navigate an AI-powered game and seek to explain game decisions [64].

5. General training and explanation. A Palo Alto Research Center (PARC) team (including researchers from Carnegie Mellon University, the Army Cyber Institute, the University of Edinburgh, and the University of Michigan) is developing an interactive explanatory system that could explain the capabilities of the XAI system driving a simulated unmanned aerial system. Explanations of the XAI system should communicate what information it uses to make decisions and whether it understands how everything works. To address this problem, PARC (COGLE) and its users are establishing a common framework for defining which terms to use in explanations and their meanings. This is provided by the PARC introspective discourse model, which alternates between learning and explanation.

COGLE's layered architecture separates information processing into comprehension, cognitive modeling, and learning. The learning layer uses repetitive and hierarchical deep neural networks with limited bandwidth to create abstractions and compositions on the states and actions of unmanned aerial systems to support the understanding of generalized patterns.

COGLE's annotation interfaces support performance analysis, risk assessment, and training. The first interface is a map that tracks the actions of unmanned aerial systems and divides the path of action or decision (flight) into explainable segments. The second interface tools allow users to explore and assess system competencies and make predictions about mission performance. COGLE is being demonstrated on the ArduPilot Software-in-the-Loop Simulator and on the discrete abstract simulation test bed. Its quality is evaluated by drone operators and analysts. Competency-based assessment will help PARC determine how best to develop suitable models that are understandable for the domain.

6. Explainable reinforcement learning (RL) at Carnegie Mellon University. Carnegie Mellon University is creating a new discipline of explainable RL to enable dynamic human-machine interaction and adaptation for maximum team productivity. Scientists have two goals: to develop new methods for studying explainable RL algorithms and to create strategies that can explain the existing black box problems. To achieve the first goal, Carnegie Mellon is developing methods to improve model learning for RL agents to take advantage of model-based approaches (the ability to visualize plans in the interior of a model) while combining them with the benefits of model-less approaches (simplicity and maximum performance). These include methods that progressively add states and actions to models of the world after matching hidden information has been discovered, study models through end-to-end training on complex optimal control algorithms, explore general DL models that use rigid body physics [65], and study predictions of states using repetitive architectures [66].

Carnegie Mellon University is also developing methods that can explain the actions and plans of the black box's RL agents. Methods include answering questions such as "Why did the agent choose a particular action," or "What training data influenced this choice the most." For this the university has developed methods that generate agent descriptions from behavior logs and detect outliers or anomalies. Carnegie Mellon University has tackled autonomy and has demonstrated explainable RL in several scenarios, including OpenAI Gym, Atari games, autonomous vehicle simulations, and mobile service robots.

7. Explainable generative adversarial networks. The SRI International team (including researchers from the University of Toronto, the University of Guelph, and the University of California, San Diego) is developing an explainable machine learning framework for multimodal data analysis that generates understandable explanations with the rationale for decisions, accompanied by visualizations of the input

data used to generate inferences. The deep attention-based representation system for explainable generative adversarial networks (DARE/X-GANS) employs DNN architectures similar to models of attention in visual neuroscience. It identifies, extracts, and presents evidence to the user as part of the explanation. Attention mechanisms provide the user with the means to explore the system and work together. DARE/X-GANS uses generative adversarial networks (GANs) that learn to understand data by creating it while learning representations with explanatory power. GANs become explainable with interpreted decoders. This includes generating visual evidence for the given text queries using chunked text generation [67], with the chunks being interpreted features such as human poses or bounding boxes. This evidence is then used to find the requested visual data.

8. A system of answers to explainable questions. The Raytheon BBN Technologies team (including researchers from Georgia Institute of Technology, Massachusetts Institute of Technology, and the University of Texas at Austin) is developing a system that answers any natural language (NL) questions users ask about media and provides interactive possible explanations as to why the user got the answer received. Explainable Answering Questions System (EQUAS) studies explainable DNN models in which internal structures (e.g., individual neurons) are aligned with semantic concepts (e.g., wheels and steering wheel) [68]. This allows neural activations in the network during the decision-making process to be translated into NL explanations (for example, "this object is a bicycle because it has two wheels and a handlebar"). EQUAS also uses neural imaging techniques to highlight the input areas related to the neurons that most influenced its decisions. To express case-based explanations, EQUAS stores indices and extracts cases from its training data that support its selection. Rejected alternatives are recognized and excluded using contrasting language, visualization, and examples. The four modes of explanation correspond to the key elements of argument building and interactive learning: didactic statements, visualizations, cases, and rejection of alternatives.

9. Controlled probabilistic logic models. A team at the University of Texas at Dallas (UTD) (including researchers from UCLA, Texas A&M, and Indian Institute of Technology Delhi) is developing a unified approach to XAI using controlled probabilistic logic models (TPLM). TPLM is a family of representations that includes decision trees, binary decision diagrams, section networks, maximal decision diagrams, first-order arithmetic circuits, and controlled Markov logic [69]. UTD extends TPLM to generate explanations of query results. For scalable inference, the system applies new algorithms to answer complex explanatory queries using techniques such as generalized inference, variational inference, and combinations of these.

10. Texas A&M University (TAMU). The TAMU team (including researchers from Washington State University) is developing an interpretable DL framework that uses simulation learning to apply explainable shallow models and facilitate domain interpretation with visualization and interaction. Interpretable system learning algorithms extract knowledge from DNN for appropriate explanations. Its DL module connects to the template generation module using the interpretability of shallow models. Learning outcomes are displayed to users with visualizations including coordinated and integrated views. The TAMU system processes image data [70] and text [71] and is applied in the XAI analytics problem domain. It provides an efficient interpretation of the detected inaccuracies from a variety of sources while maintaining a competitive detection performance. The TAMU system combines model-level and instance-level interpretability to generate explanations that are easier for users to understand. The system has been deployed to address multiple challenges using data from Twitter, Facebook, ImageNet, CIFAR-10, online health forums, and news websites.

11. Explaining the model using the optimal choice of training examples (Rutgers University). Rutgers University is expanding the capabilities of Bayesian learning to enable automatic explanation by choosing the subset of the data that is most representative of the model's inference. This approach also allows explaining the conclusions of any probabilistic generative and discriminative model, as well as deep learning models [72]. Rutgers University is also developing a formal theory of human-machine interaction and supporting interactive explanations of complex compositional models. Common among these is an approach based on human learning models that promote explainability and carefully controlled behavioral experiments to quantify explainability. Explaining with Bayesian Learning introduces a dataset, a probabilistic model, and an inference method, and returns a small subset of examples that best explain the inference of the model. It has been demonstrated that this approach facilitates understanding of large bodies of texts, as measured by a person's ability to accurately compose a summary of a body of text after short, guided explanations. Rutgers University is focusing on the data analysis problem area and has demonstrated its approach in images, text, their combinations (such as VQA), and structured modeling using temporal causation.

CONCLUSIONS

This article reviews the existing algorithms for extracting rules from ANN networks and machine learning models. Some of the modern algorithms fall into three categories: decompositional, pedagogical, and eclectic. Particular attention is paid to the extraction of rules from neural fuzzy networks. The study of fuzzy logic culminated in the late 20th century, and has since begun to decline. This decline may be partly due to a lack of results in machine learning. Rule extraction is one way to help understand neural networks. This research will pave the way for fuzzy logic researchers to develop AI applications and solve complex problems that are also of interest to the machine learning community. Experience and knowledge in the field of fuzzy logic are suited for modeling ambiguities in big data, modeling ambiguity in knowledge representation, and providing non-inductive inference transfer learning. It also discusses rule extraction from deep learning networks that currently provide an acceptable solution to a variety of AI problems. This is a new field of machine learning that is believed to take machine learning one step further in the field of pattern recognition and text understanding. However, in terms of explanations, it is still a black box model. In the past few years, the problem has been expanded to include the general concept and knowledge extraction from machine learning algorithms: explainable AI. Advances in machine learning and the rise in computing power have led to the development of intelligent systems that can be used to recommend a movie, diagnose cancer, make investment decisions, or drive a car without a driver. However, the effectiveness of these systems is limited by the inability to explain decisions and actions to the user. The DARPA Explainable AI program develops and evaluates a wide range of new machine learning methods: modified deep learning methods that study explainable functions; methods that explore more structured, interpretable causal patterns; and inductive model methods that derive an explainable model from any black box model. The technologies and results obtained show that these three strategies deserve further study and will provide future developers with design options that increase productivity and explainability.

FUNDING

This study was supported by the Russian Foundation for Basic Research (grant no. 20-17-50199) under the Expansion Program.

REFERENCES

1. M. Bilgic and R. J. Mooney, "Explaining recommendations: Satisfaction vs. promotion," in *Proceedings of the Beyond Personalization Workshop, 2005,* Vol. 5, p. 153.
2. W. R. Swartout and J. D. Moore, "Explanation in second generation expert systems," in *Second Generation Expert Systems* (Springer, Berlin, 1993), pp. 543–585.
3. B. Chandrasekaran, M. C. Tanner, and J. R. Josephson, "Explaining control strategies in problem solving," IEEE Expert **4** (1), 9–15 (1989).
4. J. S. Dhaliwal and I. Benbasat, "The use and effects of knowledge-based system explanations: theoretical foundations and a framework for empirical evaluation," Inform. Syst. Res. **7**, 342–362 (1996).
5. M. M. Eining and P. B. Dorr, "The impact of expert system usage on experiential learning in an auditing setting," Inform. Syst. **5**, 1–16 (1991).
6. D. S. Murphy, "Expert system use and the development of expertise in auditing: A preliminary investigation," Inform. Syst. **4**, 18–35 (1990).
7. D. M. Lamberti and W. A. Walace, "Intelligent interface design: An empirical assessment of knowledge presentation in expert systems," MIS Quart. **14**, 279–311 (1990).
8. *Artificial Intelligence, The Reference Book,* Ed. by V. N. Zakharov, E. V. Popov, D. A. Pospelov, and V. F. Khoroshevskii (Radio Svyaz', Moscow, 1990) [in Russian].
9. E. V. Popov, *Expert Systems: Solving Unformalized Tasks in a Dialogue with a Computer* (Nauka, Moscow, 1987) [in Russian].
10. W. R. Swartout, "A digitalis therapy advisor with explanations," in *Proceedings of the 5th International Joint Conference on Artificial Intelligence* (Cambridge, 1977), Vol. 2, pp. 819–825.
11. J. L. Weiner, "BLAH, A system that explains its reasoning," Artif. Intell. **15**, 19–48 (1980).
12. W. R. Swartout, C. Paris, and J. Moore, "Explanations in knowledge systems: Design for explainable expert systems," IEEE Expert **6** (3), 58–64 (1991).
13. W. J. Clancey, *Intelligent Tutoring Systems: A Tutorial Survey* (Stanford Univ. Dep. Comput. Sci., Stanford, 1986).
14. R. Sinha and K. Swearingen, "The role of transparency in recommender systems," in *Extended Abstracts of CHI'02 Conference on Human Factors in Computing Systems* (Minneapolis, 2002), pp. 830–831.
15. T. Gruber, "Learning why by being told what," IEEE Expert **6** (4), 65–75 (1991).

16. W. J. Clancey, "Details of the revised therapy algorithm," in *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* (Addison-Wesley, Reading, MA, 1984), pp. 133–146.
17. W. J. Clancey, "From GUIDON to NEOMYCIN and HERACLES in twenty short lessons," AI Magazine **7** (3), 40 (1986).
18. A. Arioua, P. Buche, and M. Croitoru, "Explanatory dialogs with argumentative faculties over inconsistent knowledge bases," Expert Syst. Appl. **80**, 244–262 (2017).
19. U. Johansson, T. Lofstrom, R. Konig, C. Sonstro, and L. Nilsson, "Rule extraction from opaque models-a slightly different perspective," in *Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA'06), Orlando, FL, USA, 2006,* pp. 22–27.
20. M. Craven and J. Shavlik, "Rule extraction: Where do we go from here," in *University of Wisconsin Machine Learning Research Group Working Paper* (Wisconsin, 1999), pp. 99–108.
21. R. Andrew, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial networks," Knowledge-based Syst. **8**, 373–389 (1995).
22. S. Thrum, "Extracting provably correct rules from artificial neural networks," Technical Report (Inst. Inform. III, Bonn, 1993).
23. M. Craven and J. W. Shavlik, "Using sampling and queries to extract rules from trained neural networks," in *Proceedings of the 11th International Conference, Rutgers Univ., New Brunswick, USA, 1994,* pp. 37–45.
24. L. Fu, "Rule generation from neural networks," IEEE Trans. Syst. Man Cybern. **24**, 1114–1124 (1994).
25. M. Sato and H. Tsukimoto, "Rule extraction from neural networks via decision tree induction," in *Proceedings of International Joint Conference on Neural Networks (IJCNN'01)* (Washington, DC, 2001), Vol. 3, pp. 1870–1875.
26. A. B. Tickle, R. Andrew, M. Golea, and J. Diederich, "The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks," IEEE Trans. Neural Networks **9**, 1057–1068 (1998).
27. K. K. Sethi, D. K. Mishra, and B. Mishra, "KDRuleEx: A novel approach for enhancing user comprehensibility using rule extraction," in *Intelligent Systems, Modelling and Simulation (ISMS), Proceedings of the 3rd International Conference, Kota Kinabalu, Malaysia, 2012,* pp. 55–60.
28. U. Johansson, T. Lofstrom, R. Konig, C. Sonstrod, and L. Niklasson, "Rule extraction from opaque models-a slightly different perspective," in *Proceedings of the 5th International Conference on Machine Learning and Applications, ICMLA'06, Orlando, FL, USA, 2006,* pp. 22–27.
29. M. Rangwala and G. R. Weckman, "Extracting rules from artificial neural networks utilizing TREPAN," in *Proceedings of IIE Annual Conference, Orlando, Florida, 2006.*
30. T. Hailesilassie, "Extraction algorithm for deep neural networks: A review," Int. J. Comput. Sci. Inform. Secur. **14**, 376–381 (2016).
31. A. Averkin and S. Yarushev, "Hybrid neural networks and time series forecasting," Commun. Comput. Inform. Sci. **934**, 230–239 (2018).
32. G. Pilato, S. A. Yarushev, and A. N. Averkin, "Prediction and detection of user emotions based on neuro-fuzzy neural networks in social networks," in *Proceedings of the 3rd International Scientific Conference on Intelligent Information Technologies for Industry IITI'18, Sochi, Russia,* Adv. Intell. Syst. Comput. **875**, 118–126 (2018).
33. A. N. Averkin, G. Pilato, and S. A. Yarushev, "An approach for prediction of user emotions based on ANFIS in social networks," in *Proceedings of the 2nd International Scientific and Practical Conference on Fuzzy Technologies in the Industry FTI 2018–CEUR Workshop, Ostrava-Prague, Czech Republic, 2018,* pp. 126–134.
34. X.-H. Jin, "Neurofuzzy decision support system for efficient risk allocation in public-private partnership infrastructure projects," J. Comput. Civ. Eng. **24**, 525–538 (2010).
35. X.-H. Jin, "Model for efficient risk allocation in privately financed public infrastructure projects using neuro-fuzzy techniques," J. Constr. Eng. Manag., 1003–1014 (2011).
36. V. V. Borisov, A. S. Fedulov, and M. M. Zernov, *Fundamentals of Hybridization of Fuzzy Models,* Vol. 9 of *Fundamentals of Fuzzy Mathematics Series* (Goryachaya Liniya-Telekom, Moscow, 2017) [in Russian].
37. D. Rutkowska, M. Piliński, and L. Rutkowski, *Neural Networks, Genetic Algorithms and Fuzzy Systems* (Naukowe PWN, Warszawa, 2008) [in Polish].
38. S. Rajab and V. Sharma, "A review on the applications of neuro-fuzzy systems in business," Artif. Intell. Rev. **49**, 481–510 (2018).
39. S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: Survey in soft computing framework," IEEE Trans. Neural Network **11**, 748–768 (2000).
40. J. Vieira, F. Morgado-Dias, and A. Mota, "Neuro-fuzzy systems: A survey," WSEAS Trans. Syst. **3**, 414–419 (2004).

41. J. Kim and N. Kasabov, "HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems," Neural Network **12**, 1301–1319 (1999).

42. K. V. Shihabudheen and G. N. Pillai, "Recent advances in neuro-fuzzy system: A survey," Knowl.-Based Syst. **152**, 136–162 (2018).

43. I. Z. Batyrshin, A. O. Nedosekin, A. A. Stetsko, V. B. Tarasov, A. V. Yazenin, and N. G. Yarushkina, *Fuzzy Hybride Systems: Theory and Practice,* Ed. by N. G. Yarushkina (Fizmatlit, Moscow, 2007) [in Russian].

44. Z. J. Viharos and K. B. Kis, "Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement," Measurement **67**, 126–136 (2015).

45. C. T. Lin and C. S. G. Lee, "Neural network based fuzzy logic control and decision system," IEEE Trans. Comput. **40**, 1320–1336 (1991).

46. J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," IEEE Trans. Syst. Cybern. **23**, 665–685 (1993).

47. H. Naderpour and M. Mirrashid, "Shear failure capacity prediction of concrete beam-column joints in terms of ANFIS and GMDH," Pract. Period. Struct. Des. Constr. **24** (2) (2019).

48. L. Fan, "Revisit fuzzy neural network: Demystifying batch normalization and ReLU with generalized hamming network," in *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, 2017,* pp. 1920–1929.

49. H. R. Bherenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," IEEE Trans. Neural Networks **3**, 724–740 (1992).

50. D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation," Fuzzy Sets Syst. **101**, 261–271 (1999).

51. S. Tano, T. Oyama, and T. Arnould, "Deep combination of fuzzy inference and neural network in fuzzy inference," Fuzzy Sets Syst. **82**, 151–160 (1996).

52. J. Ch. Feng and L. Ch. Teng, "An online self constructing neural fuzzy inference network and its applications," IEEE Trans. Fuzzy Syst. **6**, 12–32 (1998).

53. N. Kasabov and Qun Song, "Dynamic evolving fuzzy neural networks with 'm-out-of-n' activation nodes for on-line adaptive systems," Technical Report TR99/04 (Department of Inform. Sci., Univ. Otago, Otago, 1999).

54. D. Gunning and D. Aha, "DARPA'S explainable artificial intelligence (XAI) program," AI Magazine **40** (2), 44–58 (2019).

55. A. N. Gorban', "Errors of data-based intelligence," in *Proceedings of the International Conference on Intelligent Systems in Science and Technology, and the 6th All-Russian Scientific and Practical Conference on Artificial Intelligence in Solving Urgent Social and Economic Problems of the XXI Century, Perm, 2020,* pp. 11–13.

56. R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, New York, 2017), pp. 804–813.

57. J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *Proceedings of the International Conference on Computer Vision* (IEEE, New York, 2017), pp. 2942–2950.

58. L. A. Hendricks, T. Darrell, and Z. Akata, "Grounding visual explanations," in *Proceedings of the European Conference of Computer Vision (ECCV), Munich, Germany* (Springer, 2018).

59. K. Marazopoulou, M. Maier, and D. Jensen, "Learning the structure of causal models with relational and temporal dependence," in *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence, Association for Uncertainty in Artificial Intelligence, Amsterdam, Netherlands, 2015,* pp. 572–581.

60. A. Pfeffer, *Practical Probabilistic Programming* (Manning, Greenwich, CT, 2016).

61. M. Harradon, J. Druce, and B. Ruttenberg, "Causal learning and explanation of deep neural networks via autoencoded activations," arXiv: 1802.00541v1 [cs.AI] (2018).

62. L. She and J. Y. Chai, "Interactive learning for acquisition of grounded verb semantics towards human-robot communication," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017,* Vol. 1, pp. 1634–1644.

63. Z. Qi and F. Li, "Learning explainable embeddings for deep networks," in *Proceedings of the NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning, Long Beach, 2017.*

64. J. Dodge, S. Penney, C. Hilderbrand, A. Anderson, and M. Burnett, "How the experts do it: Assessing and explaining agent behaviors in real-time strategy games," in *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Assoc. for Comput. Machinery, New York, 2018), pp. 1–12.

65. F. Belbute-Peres and J. Z. Kolter, "A modular differentiable rigid body physics engine," in *Neural Information Processing Systems, Deep Reinforcement Learning Symposium, Long Beach, CA, 2017.*

66. A. Hefny, Z. Marinho, W. Sun, S. Srinivasa, and G. Gordon, "Recurrent predictive state policy networks," in *Proceedings of the 35th International Conference on Machine Learning* (Int. Machine Learning Soc., Stockholm, Sweden, 2018), pp. 1949–1958.

67. P. Vicol, M. Tapaswi, L. Castrejon, and S. Fidler, "MovieGraphs: Towards understanding human-centric situations from videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, New York, 2018), pp. 4631–4640.

68. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," in *Proceedings of the International Conference on Learning Representations, San Diego, CA, 2015.*

69. V. Gogate and P. Domingos, "Probabilistic theorem proving," Commun. ACM **59** (7), 107–15 (2016).

70. M. Du, N. Liu, Q. Song, and X. Hu, "Towards Explanation of DNN-based prediction and guided feature inversion," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Assoc. Comput. Machinery, New York, 2018), pp. 1358–1367.

71. J. Gao, N. Liu, M. Lawley, and X. Hu, "An interpretable classification framework for information extraction from online healthcare forums," J. Healthcare Eng. **2017**, 2460174 (2017).

72. S. C.-H. Yang and P. Shafto, "Explainable artificial intelligence via Bayesian teaching," in *Proceedings of the 31st Conference on Neural Information Processing Systems Workshop on Teaching Machines, Robots and Humans, Long Beach, CA, 2017.*