Human-Intelligent Systems Integration
https://doi.org/10.1007/s42454-021-00031-5

RESEARCH ARTICLE

A bi-directional adversarial explainability for decision support

Saveli Goldberg[1] · Eugene Pinsky[2] · Boris Galitsky[3]

Received: 14 May 2020 / Accepted: 26 January 2021

Abstract
In this paper we present an approach to creating Bi-directional Decision Support System (DSS) as an intermediary between an expert (U) and a machine learning (ML) system for choosing an optimal solution. As a first step, such DSS analyzes the stability of expert decision and looks for critical values in data that support such a decision. If the expert's decision and that of a machine learning system continue to be different, the DSS makes an attempt to explain such a discrepancy. We discuss a detailed description of this approach with examples. Three studies are included to illustrate some features of our approach.

Keywords Decision support system · Machine learning · Machine-user interaction

1 Introduction

With rapid progress across a broad range of machine learning applications in recent years, some implications of these advances are also causing concern. One set of issues that may arise as people increasingly rely on these systems is that they diminish the users' sense of responsibility for decisions and outcomes. By reducing the need for human expertise, the use of such systems could gradually lead to a loss of human expertise as well as an accuracy of future decisions. It is well known that a drop of accuracy of ML system is caused by domain evolution, where the training occurred on the original, old data and the current, new data may significantly deviate. The rate of this domain evolution, concept drift (Krawczyk et al. 2017), can be much higher that the self re-training capabilities of the ML.

✉ Eugene Pinsky
epinsky@bu.edu

Saveli Goldberg
sigoldberg@mgh.harvard.edu

Boris Galitsky
boris.galitsky@oracle.com

[1] Radiation Oncology Department, Mass General Hospital, Boston, MA, USA

[2] Department of Computer Science, Met College, Boston University, Boston, MA, USA

[3] Oracle Corp Redwood Shores, Redwood City, CA, USA

Current approaches to addressing these issues focus on improving the *explainability* of decisions generated by ML algorithms and by requiring that humans confirm or approve such ML decisions (Goodman and Flaxman 2017). However, many of the popular and effective methods widely used in machine learning today, such as random forest, neural networks, support vectors machines and many others, do not explain their decision. The solution to this can be consideration of the decision-making algorithm as a black box and based on this, an explanation of the decision is built. Recently, a considerable progress has been made towards such explainability of decisions (Scott et al. 2019; Baehrens et al. 2010; Bourneffouf et al. 2016; Ribeiro et al. 2016). In our opinion, this is an extremely important and promising approach of interactive communication between an expert and machine learning for understanding a machine solutions (Cronin et al. 2008). However, the better ML systems become, the more likely will users stop putting more effort into analyzing or critically evaluating the algorithms' decisions, even if automated explanations are also provided.

The optimal human-machine interaction can be helped by considering such interaction from a game theory perspective.

Game theory can be an efficient tool for the real-time forecasting of decision-makers in an adversarial interaction setting. Classical models from game theory allow for qualitative characteristics of the outcomes of scenarios associated with various forms of behavior of competitive agent. These models can support the design of incentives for driving the goals of these agents such as ML agents.

Multiagent learning is a key problem in AI, including learning how to coordinate adversarial problem-solving

Published online: 20 February 2021

agents. In the presence of multiple Nash equilibria, even agents with non-conflicting interests may not be able to learn an optimal coordination policy. The problem becomes even more complex if the agents do not know the game and independently receive noisy payoffs. So, multiagent reinforcement learning involves two interrelated problems: identifying the game and learning to play. Xiaofeng and Tuomas (2002) presented an optimal adaptive learning, the first algorithm that converges to an optimal Nash equilibrium with probability 1 in any team Markov game. Nash equilibria can be employed by the meta-agent functionality to drive the adversarial environment of a user and an ML.

Model-free learning for multi-agent stochastic adversarial games is another important area of research. Reinforcement learning algorithms can be extended beyond zero-sum games, and they can be employed in a real-world state-action spaces. Casgrain et al. (2019) proposed a data efficient Deep-Q-learning approach for model-free learning of Nash equilibria for general-sum stochastic games. The algorithm uses a local linear-quadratic expansion of the stochastic game, delivering analytically solvable optimal actions. This expansion is parameterized by a neural network to assure sufficient flexibility to acquire the features the environment without exhaustive navigation through it. In the case of the current study, such the stochastic game can be a foundation of the meta-agent functionality to control the interaction between a user and an ML.

Among the applications of game theory are energy and power systems relying on game theoretic models in a broad spectrum of applications. In particular, these types of approaches have been implemented in the modeling of various aspects of smart grid control.

The use of game theoretic models creates new opportunities for modeling dynamic economic interactions between utility providers and consumers inside a distributed electricity market (Ni et al. 2015). Another example study is the investigation of *crowdfunding* as an incentive design methodology for the construction of electric vehicle charging piles.

In the majority of the game theoretic modeling applications, results are generated purely by simulation without the use of real data. Also, existing applications of game theory do not propose any novel techniques for learning the underlying utility functions that dynamically predict strategic actions. Due to these limitations, one cannot reasonably expect to learn (or estimate) user functions in a gaming setting nor generalize results to broader scenarios. In real-life applications, the game theoretic models are not known a priori; therefore, the developed methods should have some way to account for data-driven learning techniques. explored utility learning and incentive design as a coupled problem both in theory and in practice under a Nash equilibrium model (Ratliff Lillian et al. 2014). Ioannis et al. (2019) present a general learning framework that leverages game theoretic concepts for learning models of occupant decision making in a competitive setting and under a discrete set of actions. The authors also presented their utility learning approaches in a platform-based design flow for smart buildings.

We are currently pursuing research to build intelligent human-machine interactions by introduce a bi-directional adversarial meta-agent or decision support system (DSS) between the user and the ML algorithm (Galitsky and Goldberg 2019; Goldberg et al. 2019). This adversarial decision support process supports and testing conflicting one-way positions taken by the OD and the expert, as a contribution to the conflict resolution situation. This DSS restructures the interaction between a user and the ML, in particular, in order to mitigate the potential loss of expertise and restore a fuller sense of responsibility to users.

Central to this is the requirement that a user makes a first unassisted decision (Goldberg et al. 2019). This initial decision is provided as an input to the algorithm before the algorithm generates its own automated decision. The DSS is trying to find *weaknesses* in the decision of the user, which may be, in particular, be a result of the user's *cognitive bias* (Plous 1993). If the user's decision continues to differ from the decision of the ML, the DSS helps the user identify the reasons for this discrepancy. In our opinion, the proposed architecture could form the basis for successful modeling of expert behavior in the presence of abnormal machine decisions. This issue was examined in detail in Illankoon et al. (2019) including the model proposed there.

In traditional Machine Learning setting, a user specifies a set of input parameters. The ML algorithms uses a training set of "similar" inputs to derive its decision. The user does not know that that a slight change in any of these inputs could result in a different result (Fig. 1).

By contrast, when using Decision Support Systems, not only is the user given a ML decision but this decision is explained and the DSS finds the values of input parameters that will force it to change its decision. Informing the user of these critical values is important as it alerts the user to pay more attention to these parameters.

2 Example of a decision support system

We present a classification problem for three animals: a wolf, a greyhound and a coyote, relying on the following parameters: animal length, skin color, height, speed, tail length and tail direction (Table 1).

Imagine a Zoo CRM environment where a human visitor saw an animal at a distance and wants to know whether

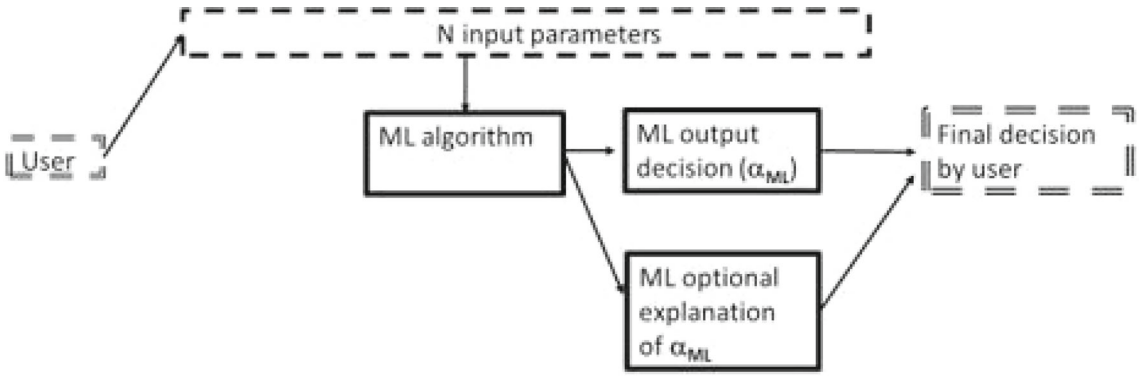

Fig. 1 Standard incorporating of ML algorithm in decision making

it is a wolf, a greyhound or a coyote. Image recognition algorithms are unlikely to be helpful for Zoo CRM in this case since dogs, wolves and coyotes are similar, especially when seen from far away in a cage. Imagine the user can enter key features such as size and fur color into the CRM DSS and iteratively converge to a solution. In total, there are 6 features describing the animal (length, color, height, speed, tail length and tail direction. In our example, 4 features (length, color, speed and tail length) are numerical and the other 2 features (color and tail direction) are categorical.

Human agent and DSS have different models of a phenomenon such as an animal. They cannot exchange model parameters but instead they can encourage each other to pay attention to particular parameters they think are important for recognition.

As a start, given some input values, the user makes an initial decision that the animal is a wolf. Let us assume that this decision was based on length, color, height, tail direction and that these features are most important for our expert. The DSS may agree with such a decision. However, a small change in these parameters (at the level of measurement errors) could cause the ML system to change its decision.

The DSS would then look for the most critical parameters that would determine such a change. In our example, such a parameter could be speed or tail direction. The DSS would then ask the user as to how reliable are these parameters. As long as the user and ML decisions are different, the DSS would be trying to explain why ML decision is different from of an expert. An expert may agree or disagree with such an explanation and makes the final decision.

We now present this session more precisely on a step-by-step basis:

Step 1 A human expert takes a sample and attempts to solve a problem. Let us imagine the following parameters as identified by her:

Length = 115 cm with the range of possible errors [100 − 130]

Color = 'light grey' with the range [white . . . grey]

Height = 70 cm with the range [55 − 85]

Speed = 40 km/h with the range [35 − 45]

Tail.length = long with the range [average]

Tail.direction = down with range [straight]

Step 2 Expert decides that it is a wolf, since

Length = 115

Color = light grey

Height = 70 cm

Tail.direction = down

Step 3 Selected features are Length, Color, Height, Tail.direction

Step 4 DSS: If turn length = 115 cm into 100 cm and height = 70 cm into 55 $\mapsto$ coyote

If Tail.direction = straight $\mapsto$ dog

If without correction $\mapsto$ wolf

DSS is asking human about the tail:

Tail.direction = straight and Tail.length = average, nevertheless $\mapsto$ wolf.

Now the new set of feature values:

Tail.length = average with the range [short . . . long]

Tail.direction = straight with range [down . . . up]

Table 1 Classification features

Feature(s)	Wolf	Coyote	Greyhound
Length, sm	100–160	75–100	100–120
Height, sm	80–85	45–55	68–76
Color	Gray	Light gray	Any
Speed, km/h	Up to 60	Up to 70	70
Tail length, sm	Long	Average	Long
Tail direction	Down	Down	Not down

Step 5 DSS ↦ dog since (Step 6)
Tail.direction = straight
Speed = 40 km/h
(Explanation only for dog vs. wolf)
Expert: what if Tail.direction = down?
DSS: still dog since can only be wolf, not coyote
Speed = 40 km/h
Tail.length = average
Expert: What if both Tail.direction = down and speed = 35 km/h?
DSS: then it becomes wolf
Expert: What if Tail.direction = down and tail.length = long?
DSS: wolf

Step 6 Now the human expert can do the final judgment.

3 Example in a medical domain

We now consider a special "case" of CRM such as medical. A physician ("expert user") needs to make a diagnosis for a patient and has to differentiate between cold, flu and allergy as shown in Table 2 (NIH News in Health 2014):

Let us assume that this physician describes patient symptoms to the ML, provides his preliminary diagnosis as flu and notes that this decision was made based on "high" temperature of 100.6 °F, "a strong headache" and "a strong chest discomfort". The DSS asks to confirm "strong chest discomfort" and additional symptoms of "stuffy" and "sore" throat. Now imagine the physician revises the symptom from "strong chest discomfort" to "mild chest discomfort" and leaves the other two symptoms, "stuffy and sore throat" unchanged, and does not change the initial diagnosis. The DSS outputs the decision cold and reports that for the diagnosis "flu" it lacks "high" temperature like 101.5 °F. The physician now decides that such the revision is insignificant and maintains the initial diagnosis, or accepts this argument and changes the diagnosis to "cold".

4 Computing decisions with explanations

Let $x = (x_1, \ldots, x_n)$ be a vector of the n input parameters to the algorithm. x_i can be continuous (numerical) or categorical (Boolean) variable. Let X be a set of x. Let $v = (v_1, \ldots, v_n)$ be the particular input values entered by the user. Let us represent the example from the previous section as v = (temperature) 100.6 °F, headache(strong), stuffy_nose(strong), sore_throat("moderate"), chest discomfort("strong")). Let $D = \{\alpha_j\}, \quad j = 1, \ldots, k$ be the set of k possible decisions or output classes. Let $\alpha_U \in D$ be the initial unassisted decision of the user. Additionally we allow the user to mark a subset of input parameters $(v_1, \ldots, v_m), \quad m \leq n$ as being particularly important to their decision a_U (Fig. 2).

We define the decision function f which maps an input vector v and a class $\alpha \in D$ to confidence $c \in [0, 1]$:

$$f(\alpha, x) : \alpha, x \mapsto [0, 1].$$

Let α_{ml} be the algorithm decision based on the user-provided input values v

$$f(\alpha_{ml}, v) = \max f(\alpha, x) \text{ for all } \alpha \in D$$

For any parameter of x, its value x_i may have bias or error. Therefore, we define $\Omega(x_i)$ such that

$$\Omega(x_i)^- < \Omega(x_i) < \Omega(x_i)^+$$

as the set of values which are considered within the error bounds for x_i. The bias includes the uncertainty of an object and uncertainty of the assessor. When there is an uncertainty in assessing a feature, we have the phenomena

Table 2 Medical domain

Symptoms	Cold	Flu	Airborne allergy
Fever	Rare	Usual, high (100 °F–102 °F) sometimes higher especially in young children lasts 3–4 days	Never
Headache	Uncommon	Common	Uncommon
General aches, pain	Slight	Usual, often severe	Never
Fatigue, weakness	Sometimes	Usual, can last up to 3 weeks	Sometimes
Extreme exhaustion	Never	Usual, at the beginning of the illness	Never
Stuffy running nose	Common	Sometimes	Common
Sneezing	Usual	Sometimes	Usual

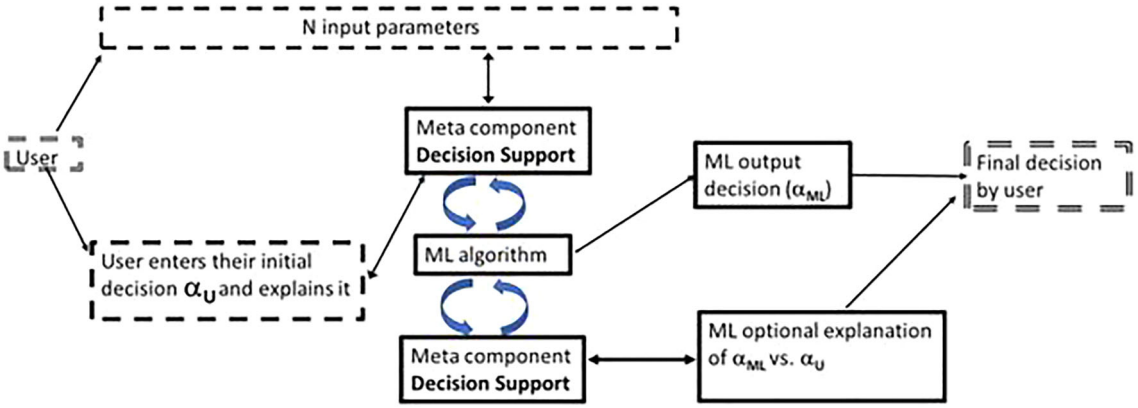

Fig. 2 Proposed user interaction flow

of "confirmation bias" and "selective perception" (Galitsky and Shpitsberg 2016; Lee et al. 2013).

We introduce a feature normalization x_i^{new} for each ith dimension, set based on the following four thresholds: $a_{0i}, a_{1i}, a_{2i}, a_{3i}$ (Goldberg 2007; Shklovsky-Kordi et al. 2005):

$$
\begin{aligned}
&x_i < a_{0i}: && \text{strong\_deviation: } x_i^{new} = 0 + x_i/a_{0i}\\
&a_{1i} < x_i < a_{2i}: && \text{abnormal: } x_i^{new} = 1\\
& && +(x_i - a_{1i}/(a_{2i} - a_{1i})\\
&a_{2i} < x_i < a_{3i}: && \text{normal: } x_i^{new} = 2\\
& && +(x_i i - a_{2j})/(a_{3i} - a_{2i})\\
&a_{3i} < x_i < a_{4i}: && \text{abnormal: } x_i^{new} = 3\\
& && +(x_i - a_{3j})/a_{4i} - a_{3i})\\
&a_{4i} < x_i: && \text{strong\_deviation: } x_i^{new} = 3 + x_i/(a_{4i})
\end{aligned}
$$

Thus, normalized parameters will belong to five intervals: [0, 1], [1, 2], [2, 3] and [3, 4], [4, ∞].

Based on this definition, we compute $X \Longleftrightarrow X^{new}$. Now we define the similarity between the object x and y as a vector distance $||x - y||$.

Division of the measured value by the accepted average value accomplishes the normalization. The calculation is executed separately for *normal*, *abnormal* and *strong_deviation* value. To define a range of sub-normal values, a team of experts empirically established the score of acceptable parameters. They are determined for certain combination of features and certain objects. If a parameter stays within the defined *abnormal* or *normal* range, no special action is required. The *strong_deviation* range covers all the zone of possible values beyond the abnormal values.

For example, in medicine, the standard scale for fever is as follows: if the body temperature is less than 95.0 °F, then it is a strong deviation. If it is in the range 95.0 °F to 96.8 °F, then it is considered abnormal. If it is in the range 96.9 °F to 99.5 °F, then it is normal. If the range is 99.6^{oo}F to 101.3 °F, then it is abnormal, and if it is greater than 101.3 °F, then it is a strong deviation. However, the norm for a flu is 100 °F to 102 °F, the norm for a cold is 99.6 °F to 101.3 °F, the norm for allergy is 96.9 °F to 99.5 °F and any higher fever is a strong deviation. This is illustrated in Fig. 3.

The normalization can be defined for categorical parameters also. For example, for allergy any general aches, pain is abnormal ($x_i^{new} = 3$) and only No General Aches, pains is normal ($x_i^{new} = 2$). We expect that, when implementing a DSS based on this approach, the thresholds is provided by domain experts using empirically established knowledge of what values of the input parameters are normal or abnormal for a given decision class.

Based on this definition, we can define a mapping between the input parameters X and the normalized parameters X^{norm}: $X \mapsto X^{norm}$ and $X^{norm} \mapsto X$. Using this normalization, we substitute $[x_1, \ldots, x_n]$ for $[x_1^{norm}, \ldots, x_n^{norm}]$. Now we can define the distance between strings x and y in a standard way as

$$||x - y|| = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$

5 An overall step-by-step DS

Here is the user interaction flow (Fig. . . .):

Step 1 Expert user input : $v = [v_1, \ldots, v_n] \in X$

Step 2 Initial unassisted decision α_U of the user. For example, flu.

Step 3 Expert user indicates m out of n input values $[v_1, \ldots, v_m]$ as being particularly important to his decision α_U. For example, (Fever = 38.1 , strong Headache, strong Chest Discomfort)

Step 4 Now DSS verified the decisions of user α_U without sharing α_{ml}. In order to determine how stable α_U is relatively to perturbations of v within error bounds Ω, we compute α_{ml} by means of Stability Assessment Algorithm.

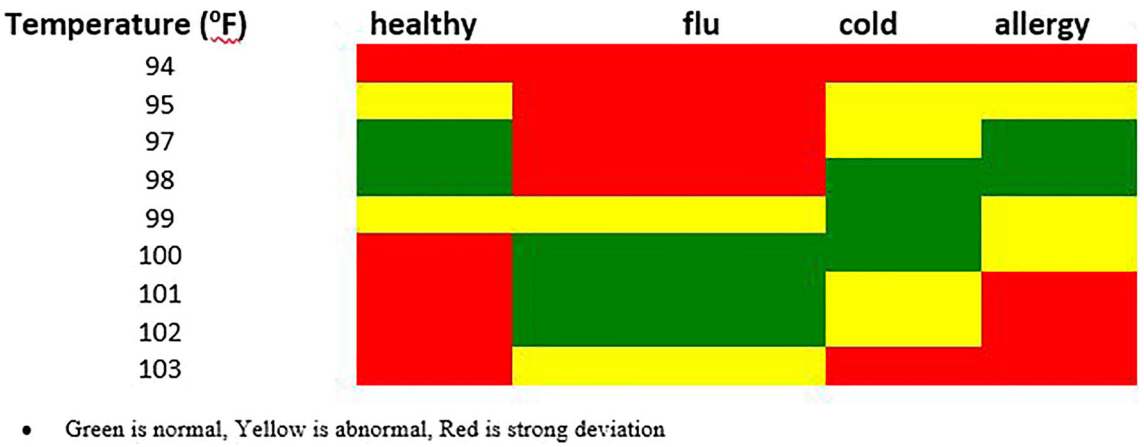

Fig. 3 Normal/abnormal ranges

If α_{ml} does not match α_U go to Step 5. If α_{ml} matches α_U then α_U is selected as a preliminary solution, and we proceed to Step 6.

Example if we have (Fever = 100.6 °F, strong Headache, strong Chest Discomfort Fever, strong Stuffy, moderate Sore Throat 100.6 °F, as user noted, α_{ml} = flu, but if we have (moderate Headache, moderate Chest Discomfort, strong Stuffy, strong Sore Throat) as obtain from $\Omega(v)$ then α_{ml} = cold).

Step 5 Since $\alpha_U \neq \alpha_{ml}$ we iteratively work with the user to see if we can converge on a stable decision. We apply *Discovering Abnormal Parameters* Algorithm.

We could, at this point, just show α_{ml} to the user, but we specifically avoid doing this in order to prevent the user from unthinkingly changing their decision to α_{ml}. Instead we use a more nuanced, indirect approach where we try to find the parameter whose value v_i from the ones indicated by the user to be in the set proving α_U, v_i is such that its possible deviation affects α_U in the highest degree.

After finding this parameter, we report to the user that the value they provided for this parameter is to some degree inconsistent with α_U. We then give the user the option to change their initial α_U.

If the user maintains the same decision α_U, then α_U is set as a preliminary decision and we proceed go to Step 6.

If user changes their decision, go to Step 2 (unless this point is reached a third time, in which case go to Step 6 to avoid putting too much pressure on the user (Goldberg et al. 2010)).

Step 6 Compute decision α_{ml} based on unchanged input values $f(\alpha_{ml}, V)$. α_{ml} is set as a decision of DSS and is shown to the human expert along with the set of key features which has yielded α_{ml} instead of α_U. Explainability of DSS algorithm is in use here.

Step 7 The human expert can modify v and observe respective decisions of DSS. DSS can in turn change its decision, and provide an updated explanation. Once the human expert obtained DSS decision for all cases of interest, she obtains the final decision.

Hence in the 3rd step, the human expert explains her decision, and in the 6th step the ML explains its decision. In the 5th step, DSS assesses the stability of human experts' decision with respect to selected features. In the 7th step, the human expert does the same with DSS decisions. So the 6th step is inverse to the 3rd and the 7th is inverse to the 5th.

For a DSS to handle explainable decision support, explanation format should be simple and have a natural representation, as well as match the intuition of a human expert. Also, it should be easy to assess DSS explanation stability with respect to deviation of decision features. It is worth mentioning that the available methods such as Baehrens et al. (2010) where DSS is a black box, similar to the current setting, do not obey all of these requirements.

We show the overall architecture of bi-directional explainable DSS in Fig. 4:

6 Three bi-directional DSS algorithms

Algorithm for step 4: Stability assessment In this step the DSS checks whether α_{ml} is stable when the input parameters are perturbed within the error bounds $[\Omega^{lower}(v_i : \Omega^{upper}(v_i]$. If, when entering the input values, the user also marked a subset of input parameters $(v_1, \ldots, v_m)$ as particularly important to their decision α_U, then the DSS only adds noise to this subset. This is because, given the user expert's focus on these parameters, they are the ones more likely to contain user bias.

Let us consider a n-dimensional space $(\Omega(v_1), \ldots, \Omega(v_m), v_{m+1}, \ldots, v_n)$. In the dimensions 1 to m it is a

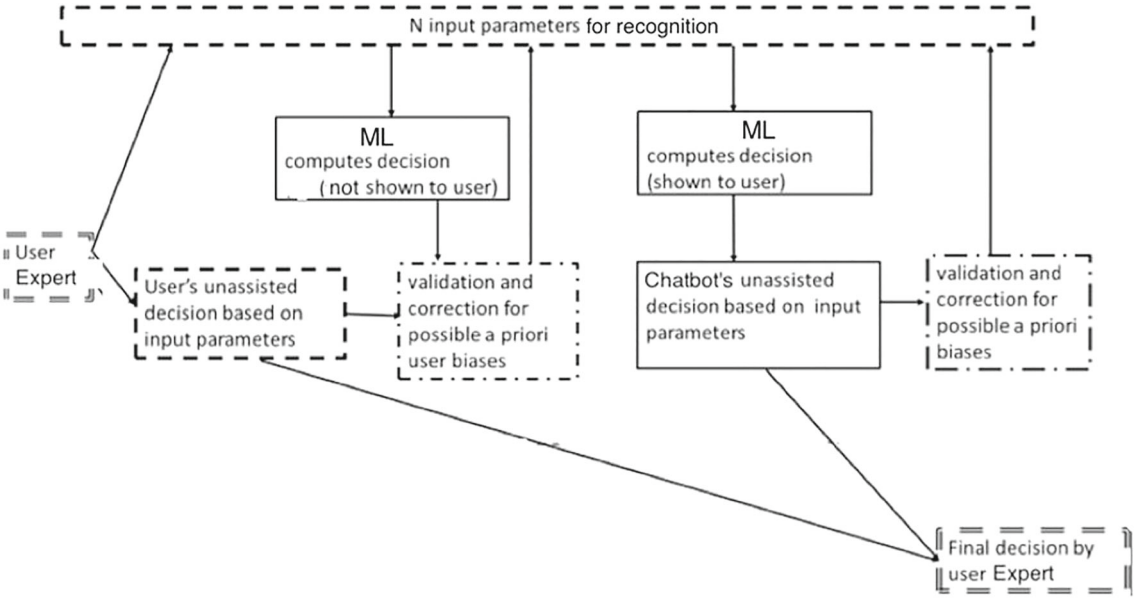

Fig. 4 Bi-directional explainable DSS

parallelepiped, and in dimension $m+1,\ldots,n$ it is a plane.

Let $\Omega(v)$ be a set of points where for each dimension $\Omega(v_i)^- < \Omega(v_i) < \Omega(v_i)^+$ for dimensions $i < m+1$ and v_i for dimensions $i > m$. Let α be the decision of DSS where $f(\alpha, x) - f(\alpha_U, x) > 0$ with $x \in \Omega(v)$ and $\alpha \in D$. Out of these pairs, let us select the pair (α_{ml}, y) which relies on a minimum number of important dimensions $1,\ldots,m$.

In our example, the precise specification of initial parameters gives the same result by the expert and by the ML. However, in the vicinity of these parameters, it is possible to find both cold and allergy diagnoses. However, for the cold diagnosis it may be enough to just lower temperature or not severe headache or strong chest discomfort, whereas for allergy we would need changes in at least 5 parameters. Therefore, the machine learning diagnosis $\alpha_{m;}$ is chosen to be *cold*.

7 Algorithm for step 5

Discovering suspicion parameters and a deviations in parameters for α_U The DSS asks the expert to reconsider the input values of the input parameters for which $\mathbf{v}'$ deviates from $\mathbf{v}$. The expert user may then realize that these input values imply a different α_U and change their initial α_U to a different α'_U. Alternatively, if the input values have a subjective component or contain errors or bias, the user may adjust the input values. In either case, if changes are made, the DSS goes back to Step 4 with the new values but does this no more than 3 times to avoid endless iteration.

Let us imagine an expert is presented with a "suspicious" parameter for α_U to support her/his decision.

From the explanation of an expert (i.e., the point at which we have the minimum)

$$\min f(\alpha_U, [v'_j]),\ \ j = 1,\ldots m, v'_i \in \Omega(v_i)$$

And the most important parameter for $\alpha_{ml} in \Omega(v_i)$ where we have the maximum

$$\max f(\alpha_{ML}, [v'_j]),\ \ j = 1,\ldots n,\ \ \ v'_i \in \Omega(v_i)$$

If $\alpha_U = \alpha_{ml}$ at point v , but $\alpha_U \neq \alpha_{ml}$ in $\Omega(v)$ and we would like to indicate more important parameters whose change would lead to decision α_{ml}. To that end, we need to look for the direction where the distance from v to α_{ml} is minimal (Fig. 5).

In this case, there is no need to get an explanation from an expert for decision α_U. However, our task in the 5th step of the algorithm also consists in creating a conflict between the choice of an expert and the ML. Our experiments showed that this usually creates the prerequisites for the expert to make the optimal decision. Therefore, the choice of a clarifying question as shown above, taking into account the expert's explanation of his decision, seems to be a preferred way.

The user expert is then suggested to consult parameter i delivering maximum value $|y_i^{new} - v_i^{new}|,\ \ i = 1,\ldots,m$. Here y_i is the ith dimension of vector $\mathbf{y}$ when feature normalization procedure is fixed. If human decision deviates from the DSS decision in initial data, meta-agent needs to focus on a single parameter value from $\{v_1,\ldots,v_n\}$

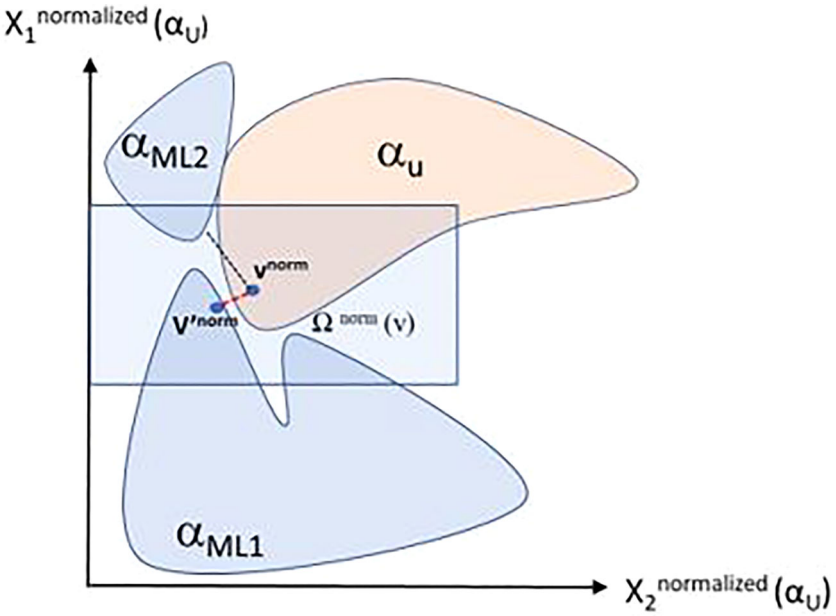

Fig. 5 DSS is finding a closer point in the normalized n-dimensional space from $v^{normalized}$ in the area where α_U turns into α_{ml}

that would direct the human expert towards the DSS decision. This is how to find this feature.

What is the worst feature dimension for a human decision? To find it we first identify the best feature value (we call it *typical*) for α_U for all i:

$$v_i^{typ}(\alpha_U) = \max_j f(\alpha_U, [v_1, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots v_n])$$

over all values x_i of ith dimension. For example, x_1= "white", x_2= "light grey", x_3 = "grey", x_4= "dark grey", x_5= "black", $j = 1.\ldots, 5$. $v_{i|^{typ}(\alpha_U}$: color = 'grey' when α_U = "wolf".

We do it for all dimensions i. Now we proceed to the dimension i best for the DSS decision

$$\max_i(f(\alpha_{ml}, [v_1, \ldots, v_{i-1}, , v_i, v_{i+1}, \ldots, v_n])$$
$$-f(\alpha_{ml}, [v_1, \ldots, v_{i-1}, v_i^{typ}, v_{i+1}, \ldots, v_n])$$

Here, the feature could be as follows v_i: color = 'light grey', $v_i^{typ}(\alpha_U)$: color ='grey' when α_U = 'wolf'.

8 Algorithm for Step 6: Explainability of ML

This algorithm attempts to explain the DSS decision for human expert in the same way as has been done by humans. DSS delivers most important features for its decision.

If at this point the user's decision still differs from the ML's decision, the DSS attempts to explain the difference between the ML decision α_{ml} and the user decision α_U in a way that is intuitive for a human user rather than a way that is based on the ML's internal representation. To do this, the DSS determines what input parameters were most important for the ML's decision. This can be done by finding the input vector $\mathbf{z}$ which is closest to the expert's input values $\mathbf{v}$ and which leads the ML to change its decision from α_{ml} to α_U. A crucial part of this step is that the distance between points $\mathbf{v}$ and $\mathbf{z}'$ is computed in normalized parameter space ($\mathbf{X^{norm}}(\alpha_{ml})$). The DSS can use a grid search in normalized parameter space to find points on the boundary between α_{ml} and α_U as shown in Fig. 6. For example, we can use Covariance Matrix Adaptation Evolution Strategy (CMAES) method (Hansen 2006). However, we consider a computationally simple and, in our opinion, more intuitive method described below. Once $\mathbf{z}$ is found, the parameters that have the largest one-dimensional distance between $\mathbf{z}'$ and $\mathbf{v}$ are taken as the parameters that are most important to explaining the difference between α_{ml} and α_U.

Let us use a random generator with $\mathbf{v}^{new}$ as average value and $(1\ldots, 1)$ vector as standard deviation to select in ^{new}, where

$$-\epsilon < f(\alpha_{ml}, \mathbf{x}) - f(\alpha_U, \mathbf{x}) < 0$$

Then we take a point $\mathbf{z}$ delivering the minimum $||\mathbf{z}^{new} - \mathbf{v}^{new}||$. Then in the cube, we randomly select a point $\mathbf{z}'$ around $\mathbf{z}$ in where

$$-\epsilon < f(\alpha_{ml}, \mathbf{x}) - f(\alpha_U, \mathbf{x}) < 0$$

such that $\mathbf{z}'$ gives us a minimum of $||\mathbf{z}'^{new} - \mathbf{v}^{new}||$. We iteratively set $\mathbf{z} = \mathbf{z}'$ and do the above iteratively till the distance $||\mathbf{z}'^{new} - \mathbf{v}^{new}||$ stops decreasing (Fig. 6).

The features i which do not belong to $\Omega(z_i')$ are important for decision making of DSS to obtain the decision α_{ml} that is different from α_U.Most important features i are those where $(z_i^{new} - v_i^{new}) \geq 1$.

As shown, the normalization *normal* vs *abnormal* is performed according to the opinion of an expert. If we

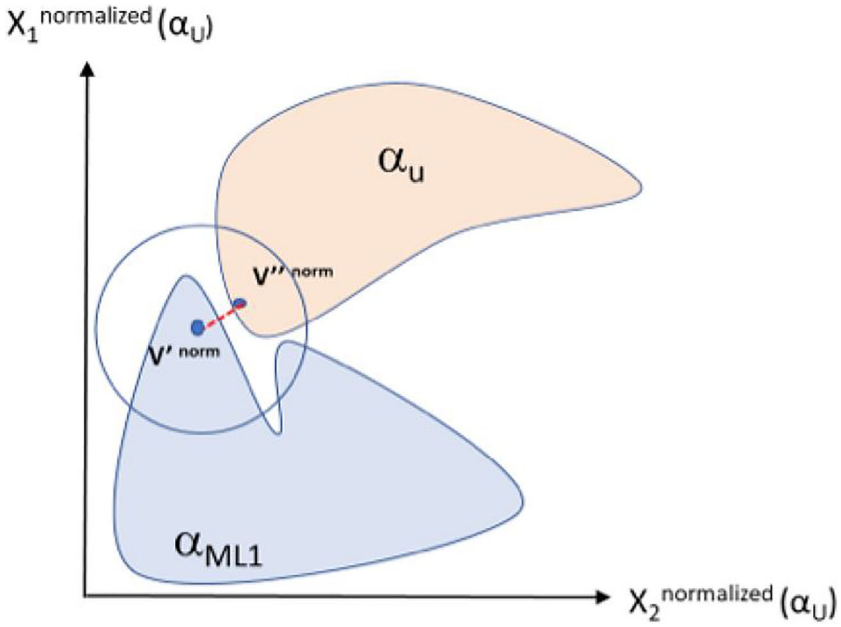

Fig. 6 DSS is finding a closer point in the normalized space

have a few points $\mathbf{v}'^{norm}$ where machine decision coincides with that of the expert and is equally close to our point $\mathbf{v}'^{norm}$, then from these points we choose closest point under normalization. It is possible that during the search for such minimal points, the decision of an expert coincides with the decision of a machine but the point itself my not exist in reality. This is possible. We assume that an expert can specify conditions for the search to avoid such a situation.

We will now present three studies to illustrate our approach.

9 Study 1: Evaluation with human experts

The influence of initial expert decision on the final decision by the user was evaluated in a series of experiments. In these experiments we analyzed how humans revise their initial decisions when they are presented by a machine decision. The participants were college students. They were asked to make judgments in the area in which they had some relevant knowledge. Specifically, they were presented with 10-sec fragments of songs of popular music groups and were asked to identify the group associated with that song (from a set of four options, each representing a different music group). After participants made their decision (*initial choice*), they were presented with the machine's decision about the same song (*machine choice*). They were told that the machine is not always accurate but were not given any specific information about the machine's accuracy. After being presented with the machine choice, participants were asked to make a final decision from the same set of options ("final choice").

Prior to the study, a survey was conducted to identify music groups with which the college students were relatively familiar. Four music groups identified in the survey as most popular and familiar to participants were selected for the present study.

Each participant took part in several test sessions. In each session, they were presented with 10–12 test items (song fragments) so that none of the items included the same song fragment. The key difference across the test sessions was the base accuracy of the machine choice, which was predetermined by us. For example, in one of the sessions, the machine was making a correct choice in 75% of items, whereas in another session it was making a correct choice 90% of times.

As seen from these results, the difference in decisions by an expert and a machine could have a positive influence on the final decision. In particular, this would happen when an expert has doubts about her/his initial decision as as that of a machine. As a rule, an expert either retains her/his original decision, or can change it to a machine decision when their decisions do not match. This occurs in 39.9% of all such conflict situations. However, in 76 cases (2.8% of all cases), the final expert opinion was different from the preliminary choice and the machine choice. This happened in 45 (6.6%) when the ML was wrong, and in 31 (1.5%) when the ML was right ($p < 0.0001$). Moreover, in 40 (88.9%) cases out of 45, when the machine was not right, the expert indicated the correct solution ($p < 0.0001$ compared to 50% of random assumptions). Even the doubts about the correctness of the initial decision had a positive impact on the final decision of the examination.

The question was as follows: *Is it possible that your final decision was different from your initial decision and the decision of the machine? (check all that apply)*. A survey of 67 students produced the following results:

1. This never happened: (40.3%);
2. I was not sure of my initial decision and did not agree with the machine solution, so I chose the third option (59.7%)*;
3. Random selection of remaining opportunities (10.5%);
4. I thought about the most likely solution and chose the third option without a machine solution (42.1%);
5. I tried to understand why the computer chose such a solution, and based on this, I chose the third option (63.2%);

2 students did not indicate reasons (3)–(5) and some of the 38 others used more than one reason.

The next experiments were conducted to measure the effect of preliminary decision on the final result.

Two groups were tested on the influence of initial solution in the following sequences:

1. **Sequence 1:** students would listen to the song. They will be told of a computer decision and were asked to put their choice
2. **Sequence 2:** students would listen to the song and put their initial decision. They would then be told of a computer decision. The students will put their final decision.

We considered 12 songs from 4 artists. These songs were presented in groups of 3 songs and the experiments were run as follows:

For the first group of 21 students, we considered:

Sequence 1, Sequence 2, Sequence 1, Sequence 2 For the second group of 21 students the same 12 songs were presented in the reverse order:

Sequence 2, Sequence 1, Sequence 2, Sequence 1 We used such a complicated design of experiments to remove influence of different knowledge level of students on identifying the authors of the songs.

Results with the machine accuracy of 66.7%, the accuracy for sequence 1 was 74.2%, accuracy for sequence 1 was 77.8% vs. 79.8% accuracy for sequence 2 ($p = 0.351$)

We then considered two other groups of students. The initial accuracy of correctly identifying artists for 12 songs without machine decision was 65.6% and 65.6% respectively. First group (16 students) would have 2 tests of sequence 2, whereas the second group (26 students) would have 2 tests of sequence 1. Each test was administered once a week and after each test the students were told of authors of the songs.

For the 3rd test, the students were given 24 songs from tests 1 and 2. Recall that for these songs the students were given the answers after listening to that music. The accuracy for the first group was 84.4% whereas for the second group the accuracy was 78.1% with a p-value $p = 0.015$.

10 Study 2: Interactive communication with user for data correction

Refinement by the DSS of initial data based on preliminary decision were implemented in the integrate system "Dinar-2" which assisted physicians in establishing the pathology and severity of cases when triaging emergency calls at the Center for Child Air-Ambulance Services in Yekaterinburg, Russia (Goldberg et al. 1991; Goldberg 1997). One of the goals of this Center was to provide remote consultation to regional medical centers and doctors involved in treating seriously ill children, and thereby reduce the need to airlift children to larger or more specialized hospitals.

The Center has served the large geographic area, so for its air-ambulance services it would often take a long time to reach regional centers. Given the volume and complexity of requests for consultation and air-ambulance services, a computerized decision support system has been key to the efficient operation of the Center. Dinar-2 was developed to fill this need. This system provides assistance in diagnosing the type of pathology (8 distinct classes of pathology), and in determining its severity (between 3 and 5 levels of severity - depending on the class). It also assists in selecting the best course of action, and in selecting the health care center that is best suited for treating a given patient.

The Dinar-2 decision support algorithm consists of 3 stages:

1. Identification of informative patterns and groups of symptoms
2. Determination of the likely pathology based on 1
3. Determination of severity

These steps were implemented using rule-based machine learning algorithms.

Besides objective measurements and test results, the system had to take into account a significant amount of subjective information about the patient's condition. This made the decision support task more complicated because the subjective information was susceptible to conscious and subconscious biases on the part of the reporting physicians. Specifically, these biases tended to skew the provided

information toward making a patient's condition appear either more or less severe than it actually was.

Due to this, the Dinar-2 decision support system assigned an a-priori confidence interval to every input parameter that was based on subjective information. Then, the system perturbed the inputs within the bounds of these confidence intervals and checked whether it's computed diagnosis was consistent with the diagnosis initially proposed by the user (in this case a physician at the Center, in consultation with the regional doctor). If, under these perturbations, Dinar-2's diagnosis of the pathology or severity did not match that of the user, Dinar-2 would follow the proposed interaction flow (described in Section 2 above) to clarify the diagnosis.

A long history of the DINAR-2 relevance appears to be a valid confirmation of the effectiveness of this approach. After the initial deployment in 1989, Dinar-2 was soon accepted by 39 emergency medical centers throughout Russia, Kazakhstan and Belarus and has since been continuously used. So, even in 2017, according to the report of Neonatology Department of Sverdlovsk State Children Hospital, Russia, 2018 (Report of Neonatology 2018), it was shown that during this year, the DINAR-2 system helped assess 537 cases. In 131 of these cases (24%), effective remote diagnosis and consultation proved sufficient for resolving the patient's crisis, and the need to dispatch an air-ambulance was avoided.

11 Study 3: Neural ML:explainable ML adversarial question answering

In Galitsky (2020), an adversarial game between explainable, inductive learning-based Question Answering (Q/A) system and a Deep Learning based Q/A was examined. Both systems are applied to large-scale real world datasets. A hundred-dimensional GloVe word embedding is usually used in the neural Q/A.

A human search session from the adversarial standpoint is shown in Fig. 7. A search for the correct answer occurs as an interaction between an explainable Q/A, neural Q/A and a human. The capabilities and interaction modes of each agent are indicated in frames, and their inputs and outputs—without frames.

As neural Q/A does the heavy lifting of answering a high percentage of an arbitrary-phrased questions, a deterministic DSS *AMR* can lay the last-mile toward answering all user questions. Firstly, a technique for navigating a semantic graph, organized by AMR, can verify the correctness of a D neural Q/A answer, involving syntactic and NER tags as well as semantic role information. Secondly, when the neural Q/A answer is determined to be incorrect, AMR employs answer-finding means complementary to that of neural Q/A and identifies the correct answer within the answer text (context).

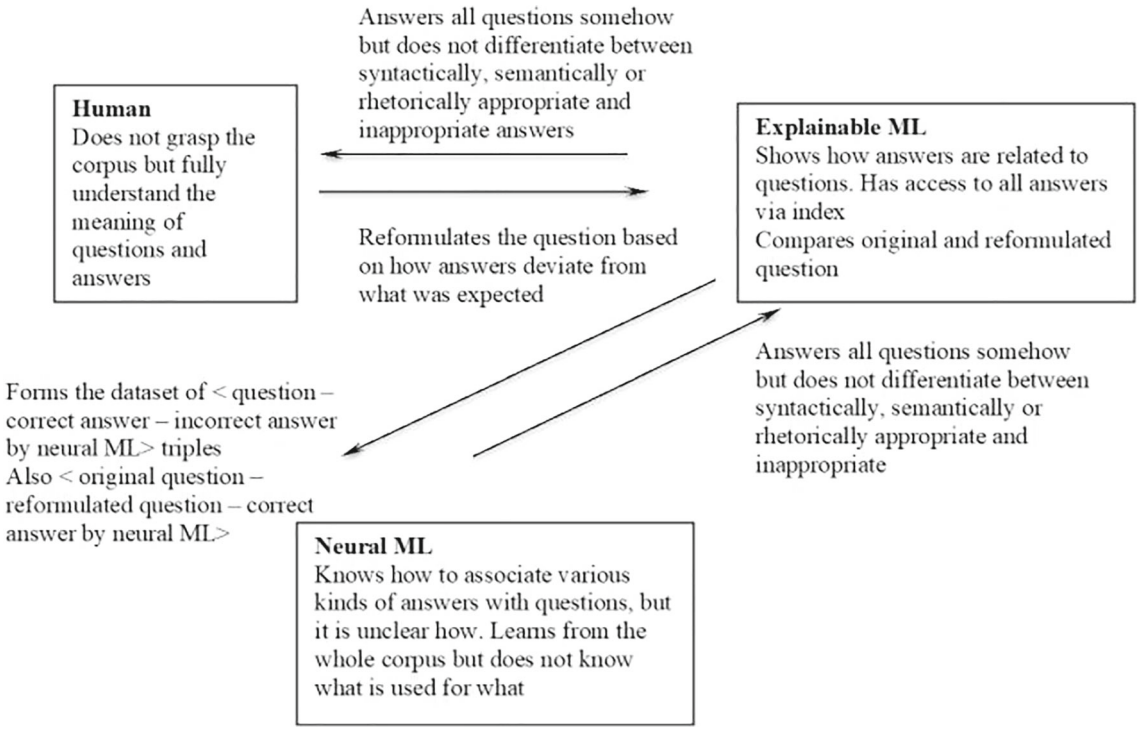

Fig. 7 Adversarial structure of interaction between explainable Q/A, neural Q/Q and a human user

Error identification and answer selection scenario of a adversarial neural and explainable Q/A system is shown in Fig. 7. It is implemented as a meta-agent.

Both the question and text from which an answer is to be extracted is subject to both syntactic and semantic parsing. Additionally, other tagging pipelines are applied including named entities, sentiment, emotion and others (Manning et al. 2014). At the next step, all available representation for questions are aligned with each other, and all representation for answer text (context) are aligned with each other as well. Finally, a search of the answer is an alignment of a hybrid (aligned) representation for the question against that of the answer. An answer fragment is a result of a maximal common subgraph between the aligned question and the aligned answer.

Interactions between the neural and explainable module works as follows (see Fig. 8). Firstly, the neural module works and obtains an answer. Then the meta-agent of the explainable Q/A components comes into play verifying that the answer is linguistically and semantically fit. To do that, it substitutes it into the question and performs syntactic and semantic matching with the answer text (context). Further details on a hybrid Q/A system are available in Galitsky (2020).

As a result when the neural Q/A was applied and delivered the correct answer in almost 90% of cases, this error-correction scenario boosted the state-of-the-art performance of a neural MRC by at least 4%.

12 Discussion

There are several benefits and opportunities afforded by the proposed approach. Requiring the user to first reach their own decision serves to counteract the loss of users' expertise and sense of responsibility that often occurs when

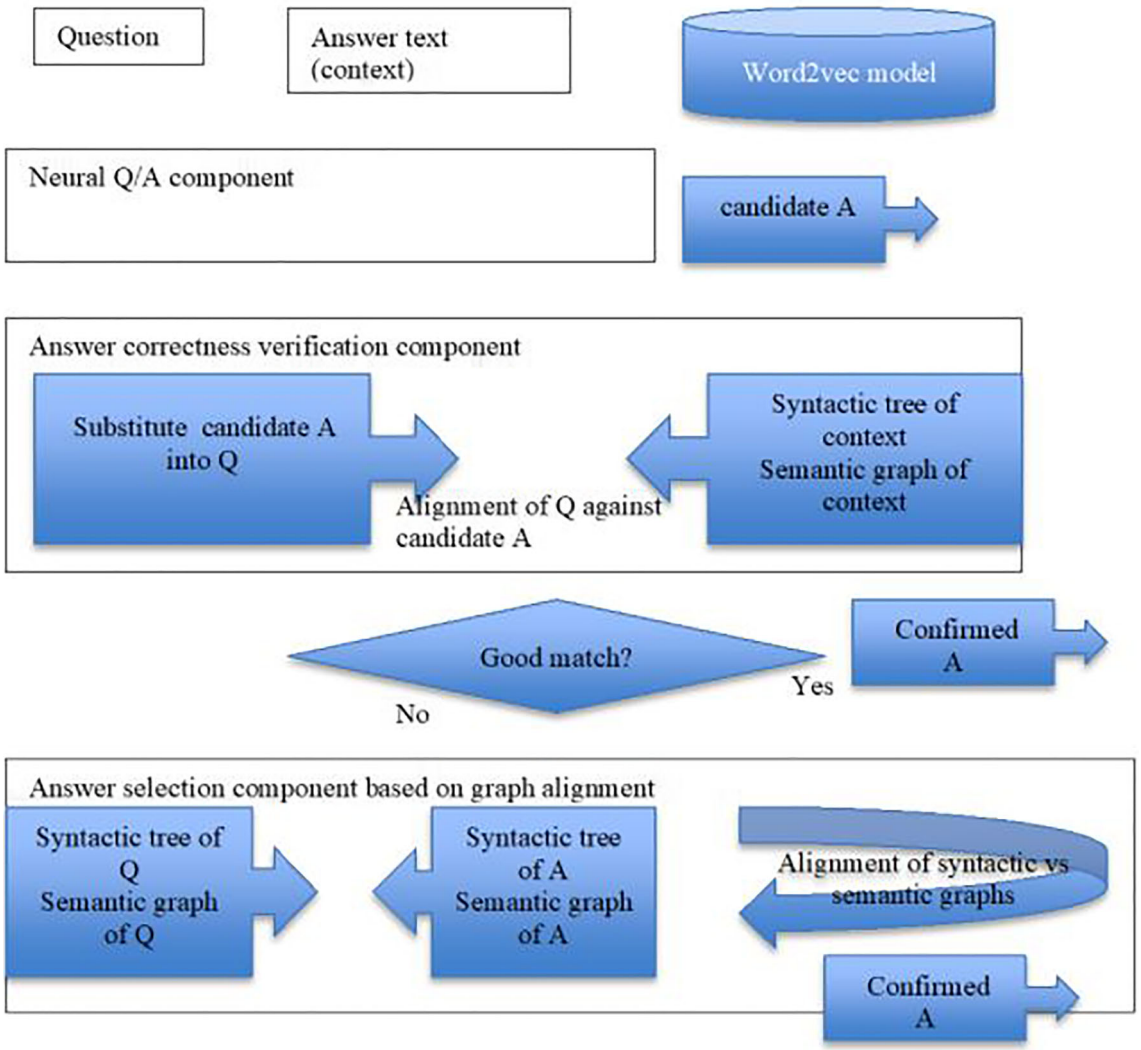

Fig. 8 Q/A architecture to support adversarial setting

users delegate decisions to a ML. It prevents the user from becoming complacent and motivates them to give more thought to their initial decision. It provides continued opportunity for user to revisit and refresh their domain knowledge. When the user and the algorithm don't agree, it forces the user to reconsider their decision in light of parameters highlighted by the algorithm. In the end, it makes it more likely that the user will critically evaluate the machine's decision. In applications where the algorithm is more accurate than human users, this even allows the user to challenge themselves to anticipate the algorithm's answer – either on their own, or explicitly, by adding game-playing elements to the interaction.

Explaining an ML classifier's decision while treating the classifier as a black box has been proposed before, for example (Baehrens et al. 2010; Bourneffouf et al. 2016). However, the fundamental point in our approach is that we did not consider the *abstract* question: *Why* α_{ml}*?* but much more specifically question: *Why* α_{ml} *and not* α_U*?* In medicine, this approach is called *Differential Diagnostics* (Siegenthaler 2011; Henderson et al. 2012).

Since our question is addressed to a machine, its formulation can be more detailed: *what minimal changes are needed for the inputs to change the machine decision from a to b?* An answer to such a question would not only give the standard answer *I understand why and I agree or disagree with the machine decision* but also suggest a correction in inputs. If changes in inputs are sufficient to change decisions and are within the measurement error, then the machine decision agrees with that of an expert. To adequately explain the machine decision, we need an adequate concept of minimal changes. Therefore, the overall data analysis is done in normed spaces. As shown, this *normal-abnormal* normalization is made from the point of view of solutions chosen by an expert.

We hope this try is relevant of the European Union's new General Data Protection Regulation which controls the applicability of machine learning (https://eugdpr.org/). These regulations restrict automated individual decision-making (that is, algorithms that make decisions based on user-level predictors) which *significantly affect* users. The law effectively creates a right to explanation, whereby a human user can request an explanation of an algorithmic decision that was made about them.

The DSS elements presented here may be used separately. Approach to explaining the ML decision and the algorithm for evaluating the users initial decision α_U can be used independently from each other.

Thus, the preliminary decision by an expert allows one to explain the machine decision as *why would the machine arrive at a decision different from that of an expert?*. This explanation could be given even in the presence of many potential decisions and prior to an interactive interaction as suggested in Molnar (2019).

On the other hand, the modification of subjective information becomes the main problem in accepting the correct decision as shown in our example with medical diagnosis above

Finally, we would like to mention a few words about the description of error ranges. It is clear that our errors are not simply 0/1 values but possess a statistical distribution with some mean. In our paper, however, we considered a simplified 0/1 description for simplicity of presentation.

Can we consider the results of our experiments to be a proof of suitability of an initial solution? No, we cannot. We understand the limitations of our experiments. We need to continue experiments with more objects and different experts under different conditions of accepting decisions, especially under direct or administrative interest of a correct solution.

Our approach has several limitations. The user's interaction with the DSS requires time which may be unavailable, or example in a system that assists with time-sensitive tasks such as operating machinery or driving a car. In applications when machine learning decisions are more accurate than an expert, the preliminary decision becomes a formality. In these circumstances, we believe that the expert ambitions could, in fact, result in worse decisions compared with that of a machine.

We continue to conduct experiments on the influence of initial expert's decision before machine assisted decision on the final decision. We are proposing to build such an automated system of explainable ML decisions with treatment of oncology patients at Mass General Hospital.

13 Conclusion

There are several advantages to structuring decision support systems in such a way that a user offers her/his own decision to the decision support system as a first step. This makes it possible to introduce a Bi-directional Adversarial Agent between the user (expert) and a machine learning system. Such an agent brings the positions of the expert and the ML *closer* in the event of a conflict between their respective decisions. We expect this approach to be implemented in practice with the goal of improving the accuracy and explainability of the final solution. This would serve to maintain, and possibly even improve, the domain knowledge of experienced users.

Declarations

Conflict of interest The authors declare no competing interests.

References

Baehrens D, Schroeter T, Harmeling S, Kawanabe M, Hansen K (2010) Klaus-robert Mller how to explain individual classification decisions. arXiv:0912.1128 [stat.ML] 11(jun): 18031831

Bourneffouf et al. (2016) Exponentiated gradient exploration for active learning. Computers 5:1–12

Casgrain P, Ning B, Jaimungal S (2019) Deep Q-learning for Nash equilibria: Nash-DQN. arXiv preprint arXiv:1904.10554

Cronin P, Ryan F, Coughlan M (2008) Undertaking a literature review: a step-by-step approach. Br J Nurs. 17(1):38–43

Galitsky B (2020) Employing abstract meaning representation to lay the last-mile toward reading comprehension. In: Artificial Intelligence for Customer Relationship Management: keeping customers informed, Springer, Cham

Galitsky B, Goldberg S (2019) Chapter 3 explainable machine learning for chatbots in B. Galitsky developing enterprise chatbots: learning linguistic structures. Springer, pp 57–89

Galitsky B, Shpitsberg I (2016) Autistic learning and cognition. Computational Autism, pp 245–293

Goldberg S (1997) Inference engine the systems of the dr. Watson type. DIMACS Workshop Rutgers University, New Jersey

Goldberg SI, Lomovskikh VE, Makhanek AO, Sklyar MS (1991) Expert system DINAR-2.-methodological basis for the pediatric emergency aid organization in a large region. In: Medical informatics europe, vienna, austria, 270–274

Goldberg S (2007) Nikita Shklovskiy-Kordi.; Boris Zingerman. Time-oriented multi-image case history - way to the disease image analysis. VISAPP (Special Sessions), pp 200–203

Goldberg SI, Niemierko A, Shubina M, Turchin A (2010) "Summary Page": A novel tool that reduces omitted data in research databases. BMC Medical Research Methodology 10:91–97

Goldberg S, Katz G, Weisburd B, Belyaev A, Temkin A (2019) Integrating user opinion in decision support systems. In: Arai K, Bhatia R (eds) advances in information and communication. FICC, Lecture Notes in Networks and Systems, 70, Springer

Goldberg S, Galitsky B, Weisburd B (2019) Framework for interaction between expert users and machine learning systems. http://ceur-ws.org/vol-2448/SSS19_paper_upload_217.pdf

Goodman B, Flaxman S (2017) European Union regulations on algorithmic decision-making and a right to explanation AI Mag Magazine, 38(3)

Hansen N (2006) The CMA evolution strategy: a comparing review, Towards a new evolutionary computation. In: Advances on estimation of distribution algorithms, Springer, 1769–1776, CiteSeerX 10.1.1.139.7369

Henderson M, Tierney L, Smetana G (2012) The Patient history: evidence-based approach to differential diagnosis., McGraw-Hill, New York NY

Illankoon P, Tretten P, Kumar D (2019) Modeling human cognition of abnormal machine behavior. Human-Intelligent Systems Integration 1:13–26

Ioannis K, Andrew B, Shiying H, Tanya V, Huihan L, Spanos C (2019) A deep learning and gamification approach to improving human-building interaction and energy efficiency in smart infrastructure. Appl Energy 237:810–821

Krawczyk B, Minku LL, Gama J, Stefanowski J, Wozniak M (2017) Ensemble learning for data stream analysis: a survey. Information Fusion 37:132–156

Lee CJ, Sugimoto CR, Zhang G, Cronin B (2013) Bias in peer review. J Am Soc Inf Sci Tec 64:2–17

Manning CD, Surdeanu M, Bauer J, Finkel J, Bethard SJ (2014) Mcclosky, The stanford coreNLP natural language processing toolkit, Proceedings of 52nd Annual Meeting of the Association for Computational linguistics: System Demonstrations, pp 55–60, Baltimore, Maryland USA, June 23–24

Ribeiro MT, Singh S, Guestrin C (2016) Why Should I Trust You? Explaining the Predictions of Any Classifier. https://arxiv.org/pdf/1602.04938.pdf

Molnar C (2019) Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. https://christophm.github.io/interpretable-ml-book/

NIH News in Health (2014) A monthly newsletter from the National Institutes of Health, part of the U.S. Department of Health and Human Services https://newsinhealth.nih.gov/2014/10/cold-flu-or-allergy

Plous S (1993) The psychology of judgment and decision making. McGraw-Hill, New York

Ratliff Lillian J et al (2014) Social game for building energy efficiency: incentive design, 52nd annual Allerton conference on communication, control, and computing. IEEE, 1011–8

Report of Neonatology (2018) Department of sverdlovsk state children hospital, Russia, 43–47

Scott M, Lundberg GGE, Lee S-I (2019) Consistent individualized feature attribution for tree ensembles. https://arxiv.org/pdf/1602.04938.pdf

Shklovsky-Kordi N, Zingerman B, Rivkind N, Goldberg S, Davis S, Varticovski L, Krol M, Kremenetzkaia AM, Vorobiev A, Serebriyskiy I (2005) Computerized case history - an effective tool for management of patients and clinical trials Engelbrecht R et al (eds)

Siegenthaler W (2011) Differential diagnosis in internal medicine: from symptom to diagnosis., Thieme Medical Publishers

Xiaofeng W, Tuomas S (2002) Reinforcement learning to play an optimal nash equilibrium in team Markov games. NIPS'02: Proceedings Of the 15th International Conference on Neural Information Processing Systems, January 1603–1610

Ni Z, Yu Y, Wencong S (2015) A game-theoretic economic operation of residential distribution system with high participation of distributed electricity consumers. Appl Energy 154:471–9