Fisheye visualization and multi-path trees for presenting clinical practice guidelines: Methods and application to Covid-19

Jean-Baptiste Lamy[1], Mouin Jammal[2], Melody Saikali[5], Charbel Mourad[3], Cynthia Abi Khalil[1,4], Antoine Saab[1,5]
1: LIMICS, INSERM, Université Sorbonne Paris Nord, Sorbonne Universités, 93017 Bobigny, France
2: Department of Internal Medicine, Lebanese Hospital Geitaoui
3: Department of Medical Imaging, Lebanese Hospital Geitaoui
4: Nursing Administration, Lebanese Hospital Geitaoui
5: Quality and Patient Safety Department, Lebanese Hospital Geitaoui
University Medical Center, PO Box 175086 Beyrouth, Lebanon
Email: jean-baptiste.lamy@inserm.fr, mouinjammal@yahoo.fr, melodysaikali@hotmail.com, charbel.j.mourad@hotmail.com, cynthiakjammal@gmail.com, antoine_saab@outlook.com

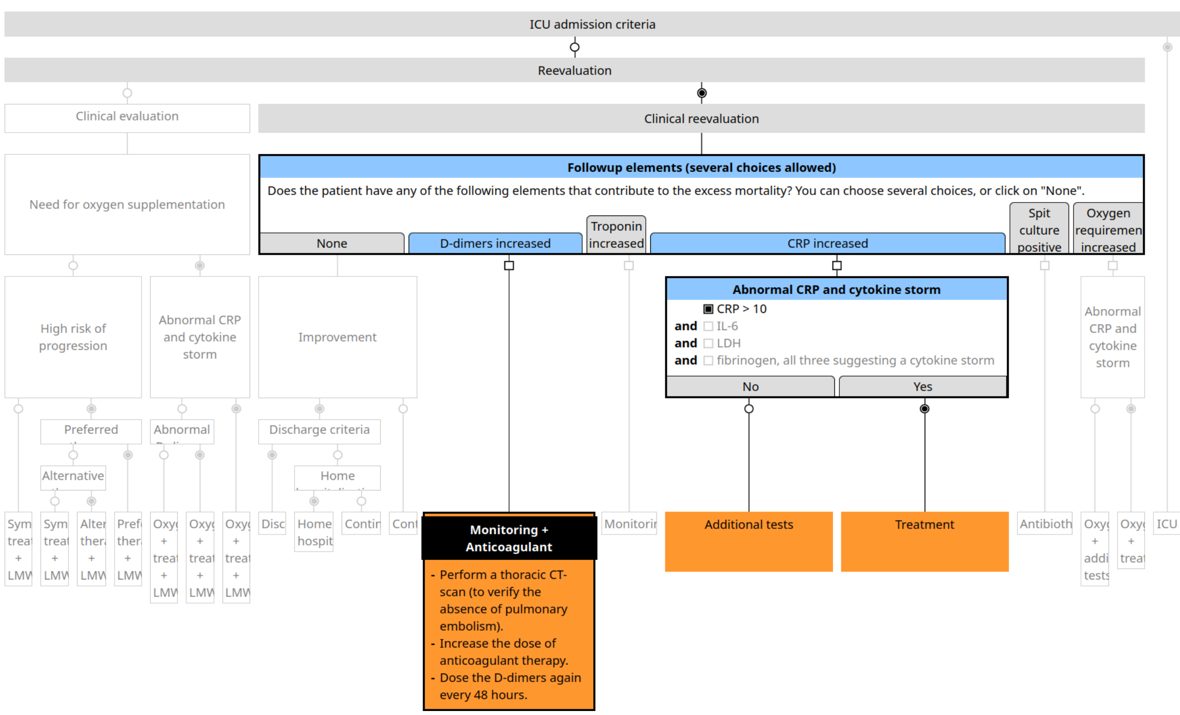

Figure 1. Interactive multi-path decision tree for the management of hospitalized Covid-19 patients. After some user interactions, three nodes are current: two question nodes ("Followup elements", a multiple-choice question node, and "Abnormal CRP and cytokine storm") and a recommendation node ("Monitoring + Anticoagulant"). Two other recommendations are accessible (labeled "Additional tests" and "Treatment").

***Abstract*—Decision trees are commonly used for representing medical reasoning, both in structured representations for automatic reasoning and in visual presentations for clinicians. However, in the latter purpose, decision trees often suffer from two problems: (a) some reasoning processes are not well suited for trees, *e.g.* when considering several independent followup elements, (b) the size of the tree is limited by the screen. Here, we propose the dynamic and interactive visualization of decision trees. To solve problem (a), we used multi-path decision trees, allowing the selection of multiple choices at some nodes, which greatly reduces the size of the tree in some situations. We provide algorithms for *one-click navigation* in a multi-path tree, *i.e.* the user may click on any node of the tree at any time. For problem (b), we used the fisheye technique for reducing the visual space devoted to the inaccessible part of the tree. In addition, we represented the tree internally in a formal ontology, making it machine-interpretable and permitting the automatic execution of parts of the tree. We applied this approach in a clinical decision support system related to the management of Covid-19, and we showed the resulting user interface to 6 clinicians.**

Index Terms—*Fisheye, Multi-path tree, Decision tree, Clinical decision support systems, Covid-19.*

I. Introduction

Decision trees are very commonly used for representing a reasoning process, in particular in the medical field. They can be produced from expert knowledge, such as clinical practice guidelines [1]. These guidelines are narrative texts that provides recommendations for the clinical management of patients having a given condition, *e.g.* regarding diagnosis or therapy. They are written by medical experts after reaching a consensus. Actually, many guidelines include informal decision trees in addition to narrative texts. Decision trees can also be automatically generated, using machine learning algorithms, *e.g.* applied to patient data.

Decision trees have many advantages: they allow automatic reasoning if the tree is fully formalized, but they are also an intuitive visual presentation format for human users. In particular, they give an overview of the entire reasoning process, and they permit what we call *one-click navigation*, *i.e.* the user can move from any node in the tree to any other node simply by clicking on the desired node. This allows skipping nodes, or going back in the reasoning.

However, decision trees often suffer from two problems: (a) some reasoning processes are not well suited for trees, *e.g.* when a guideline considers several followup elements, each receiving an independent recommendation, (b) in user interfaces, the size of the tree is limited by the size of the screen, restricting the maximum size of the trees that can be displayed and the details that can be shown [2].

In the visualization literature, many techniques have been proposed for trees, including Treemaps [3], Cone Trees [4] or even glyphs [5]. However, most of them are not well-suited for decision trees. In the medical literature, many decision tree-based clinical decision support systems actually do not show the tree to the clinician. For example, the ASTI Guiding Mode proposed navigation in huge trees, presenting to the clinicians only the current node and its children. The size of the tree was too high to present it on screen. However, the lack of overview caused up to 44% of navigation to be inappropriate [6]. JN Babione *et al.* [7] applied Human-centered design to a decision support system for pulmonary embolism; it includes a decision tree overview of the reasoning process. J. Mrva *et al.* [8] proposed a 3D radial visualization of a decision tree that facilitates the exploration and the interpretation of the tree. D. Williams *et al.* [9] associated a decision tree with medical image visualization for the classification of Parkinson's disease.

In machine learning, *multi-path* decision trees have been proposed as a tree model in which, for a given sample, several paths can be selected [10]. Alternating decision tree [11] is a variant of this model. But it is yet to apply to the presentation of decision trees. A multi-path tree would allow the user to choose several children at some nodes, and the navigation may end at more than one leaf node. In this paper, we present a dynamic and interactive visualization tool for a *multi-path* decision tree. Its multi-path nature allows the selection of multiple choices at some node, greatly reducing the size of the tree in some situations (helping with problem a). We provide algorithms adapting the *one-click navigation* for user interaction with a multi-path tree. In addition, the proposed tool combines the *details-on-demand* and *fisheye* well-known principles for reducing the visual space devoted to the unselected part of the tree, allowing the visualization of larger trees (problem b). Finally, we represented the tree internally in a formal ontology, making it machine-interpretable. It prevents ambiguities and permits the automatic execution of parts of the tree, when patient data is available.

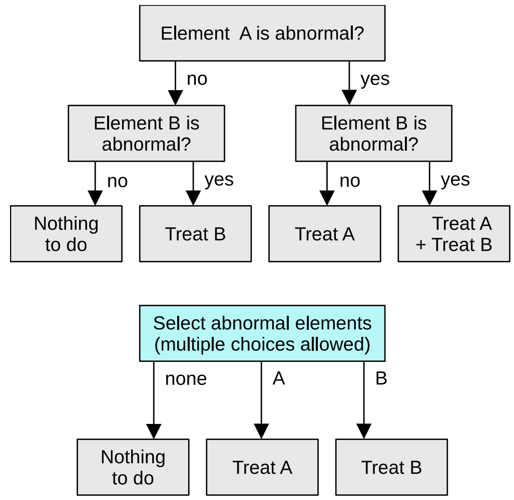

Figure 2. A decision tree with two independent followup elements A and B (top) and the corresponding multi-path tree with a multiple-choice question node (in light blue, bottom).

Using the proposed tool, we designed a decision support system implementing several decision trees related to the management of Covid-19 at various stages (phone center, home, hospital), based on international guidelines. One of the trees includes 5 followup elements, and thus was structured as a multi-path decision tree. We evaluated our tool with 6 clinicians, through qualitative reviews. Finally, we discuss the advantages and limits of our approach, and we suggest some perspectives.

II. Methods

A. Multi-path decision tree model

Let us consider a classical decision tree with two types of nodes: (a) question nodes, that ask a question to the clinician and have at least 2 children (one per possible answer), and (b) recommendation nodes, that give recommendations to the clinician and usually have no child (*i.e.* they are leaf nodes) but may occasionally have a single child. A common situation in clinical guidelines is to consider several followup elements and to propose a specific and independent response for each abnormal element. Decision trees can only deal with such situations by duplicating nodes across branches, as shown in Figure 2 (top). This is a minor problem when the tree automatically executed, however, when the tree is presented visually to human users, it makes the tree bigger and bigger and can seriously impair the usability.

To solve this problem, we propose the use of *multi-path* trees. A multi-path tree permits the simultaneous selection of several paths, thus, the final result of the navigation is not a single leaf node, but a subset of the leaves. More specifically, we propose the addition of *multiple-choice* question nodes, at which several children can be selected, in opposition to standard single-choice question nodes. An example is shown in Figure 2 (bottom). In addition, a multiple-choice question node may include a "none" child node, which can only be chosen is all other child nodes are not chosen (see example in Figure 2). For n followup elements, a classical tree requires at least 2^n leaves (assuming that the treatment of each followup element has no further question node), while a multi-path tree requires only $n+1$ leaves.

B. Formalization of interaction with a multi-path decision tree

Implementing *one-click navigation* in a multi-path tree is not trivial, because the navigation may include several paths, some of them being affected by the user interaction and others not, and because some previous nodes must remain open (*e.g.* multiple-choice question nodes, in order to let the user choose another answer, unless "none" has been chosen). In this section, we propose algorithms for permitting *one-click navigation* in a multi-path decision tree.

Let us note $\mathcal{N}$ the set of nodes in the decision tree, $\mathcal{S}$, $\mathcal{M}$ and $\mathcal{R}$ the subsets of single-choice question nodes, multiple-choice question nodes and recommendation nodes, respectively, and $root$ the root node of the tree. We note $parent(n)$ the parent of the node $n \in \mathcal{N}$, $children(n)$ the set of children of the node n and, for multiple-choice question nodes $m \in \mathcal{M}$, we note $none(m)$ the set of "none" nodes (with either 0 or 1 element) and $multi(m)$ the set of the remnant, multiple-selectable, nodes. During user navigation, the current state of the tree can be fully described by $C \subseteq \mathcal{N}$, the subset of current nodes. At the beginning, $C = \{root\}$.

We defined two algorithms that formalize the user interactions. Algorithm 1 takes the current nodes C and makes a partition of $\mathcal{N}$ in four subsets, C, P, A and I, the subsets of current, past, accessible and inaccessible nodes. Past nodes are the ancestors of the current nodes. Accessible nodes are nodes that can become current in the future interaction with the tree, without having to move backward. Accessible nodes can be obtained by computing, recursively, the descendants of current nodes, excluding multiple-choice question nodes that have at least one of their multiple-selectable children that is current or past. Inaccessible nodes are those that cannot become current without backward moves. Nodes will be displayed differently whether they belong to subset C, P, A or I, with more or less details (see section II-D).

Algorithm 1 Algorithm partitioning nodes in four subsets, C, P, A and I, the subsets of current, past, accessible and inaccessible nodes.

function ancestors($n \in \mathcal{N}$):
  if $n = root$: return $\emptyset$
  return $\{parent(n)\} \cup$ **ancestors**$(parent(n))$

function accessible_descendants($n \in \mathcal{N}$, $C \subset \mathcal{N}$, $P \subset \mathcal{N}$):
  if ($n \in \mathcal{M}$) and ($multi(n) \cap (C \cup P) \neq \emptyset$): return $\emptyset$
  return $children(n)$
    $\cup \bigcup_{k \in children(n)}$ **accessible_descendants**(k, C, P)

function make_partition($C \subset \mathcal{N}$):
  $P = \bigcup_{n \in C}$ **ancestors**(n)
  $A = \bigcup_{n \in C}$ **accessible_descendants**$(n, C, P) \setminus C$
  $I = \mathcal{N} \setminus (P \cup A \cup I)$
  return C, P, A, I

Algorithm 2 Algorithm applying the user interaction and computing the new set of current nodes.

function descendants($n \in \mathcal{N}$):
  return $\{n\} \cup \bigcup_{k \in children(n)}$ **descendants**(k)

function multiple_choice_ancestors($n \in \mathcal{N}$):
  if $n = root$: return $\emptyset$
  if $parent(n) \in \mathcal{M}$ and $n \in multi(parent(n))$:
    return $\{parent(n)\} \cup$ **ancestors**$(parent(n))$
  return **ancestors**$(parent(n))$

function apply_user_interaction($C \subset \mathcal{N}$, $P \subset \mathcal{N}$, $n \in \mathcal{N}$):
  $x = n$
  while not (($x = root$) or ($parent(x) \in C \cup P$)):
    $x = parent(x)$
  if $x = root$:
    $C' = \{root\}$
  else if not (($parent(x) \in \mathcal{M}$)
      and ($x \in multi(parent(x))$)):
    $C' = (C \setminus$ **descendants**$(parent(x))) \cup \{n\}$
  else if ($n \in C$) and ($x = n$):
    $C' = (C \setminus$ **descendants**$(x))$
  else:
    $C' = \left(C \setminus \bigcup_{k \in none(parent(x)) \cup \{x\}}\right.$ **descendants**$\left.(k)\right)$
    $\cup \{n\}$
  $C' = C' \cup$ **multiple_choice_ancestors**(n)
  return C'

Algorithm 2 considers the set of current nodes C, the set of past nodes P, and the node n clicked by the user, and computes the new set of current nodes C' after interaction. The user is allowed to click on any node, thus n can be: (a) a child of a current node (*i.e.* usual 1-step forward navigation), (b) an indirect descendant of a current node (*i.e.* skipping nodes), (c) an ancestor of a current node (*i.e.* backward navigation), (d) a current multi-selectable node (user clicked again that node to deselect it) or even (f) an inaccessible node (in this case, the user completely switch to another branch of the tree, possibly because he realized that the previous navigation was wrong).

The algorithm first follows the $parent$ relations, starting from n, to find the first node x that is either the $root$ or has for parent one of the current or past nodes. If x is the $root$ node, then the tree is reset to its initial state and only $root$ is current. Otherwise, if the parent of x is a recommendation node, a single-choice question node or a multiple-choice question node with x being its "none" node, then we remove from the current nodes the parent of x and all its descendants, and we add n. This is the usual behavior in a decision tree. Otherwise, the parent of x is a multiple-choice question node and x is not its "none" node, specific rules apply. If the user clicked on an already current multi-selectable node, then the node and its descendants are removed from current nodes. Otherwise, we remove from the current nodes the "none" children of the parent of x and all its descendants, and we add n. Notice that, in this case, the parent of x will remain current; this is intentional and will allow the user to select another choice in the multiple-choice question.

C. Formalization of the decision tree

We structured decision trees using a simple formal ontology in OWL, containing 16 classes, 25 properties and 118 axioms. It allows associating coded medical criteria with question nodes, using medical reference terminologies such as ICD10 (International Classification of Disease, release 10), ATC (Anatomical Therapeutical Chemical classification of drugs) and LOINC (Logical Observation Identifiers Names & Codes, for lab tests). Each question node may be associated with several criteria, and each criterion includes a reference terminology, a code within that terminology, an allowed time delay between the patient data and the execution of the tree (*i.e.* some clinical conditions, such as obesity, may no longer be valid after a given delay) and, for lab test results, the reference value and the comparison operator (*e.g.* < or >). Each question node is also associated with a logical operator, either "or" or "and", that permit the automatic execution of the question.

D. Visualization and fisheye

The tree is displayed vertically, with the root at the top. Recommendation nodes $\mathcal{R}$ are displayed in colored boxes, the color ranging from green to yellow to orange to red, and roughly indicating the level of danger of the situation for the patient. Question nodes are displayed in gray boxes, and inaccessible nodes I are displayed as white boxes with a gray border. At each node, children are ordered so as the nodes leading to the more "dangerous" recommendations are located on the right. Thus moving on the right usually means that the patient state is more serious.

We used the *details-on-demand* technique for displaying node content. Current nodes C are displayed with full details, including their possible answers. Other nodes are displayed as simple boxes that includes only the node's title, without answers. We also employed the *fisheye* technique for giving more space to accessible nodes A. The horizontal space is divided between accessible and inaccessible nodes so as at least half of the available space is devoted to accessible leaf nodes. If needed, inaccessible nodes are squeezed to make more space available.

During user interaction, smooth transitions are used for opening/closing node boxes, changing the node horizontal size, and scrolling to the current box. The animation permits the user to follow each box, avoiding being lost when they change place.

If coded decision criteria are available for a given current node, the corresponding line of text is displayed with a checked box if the criteria is true, and is grayed if the criteria is false (for lab test results only; missing disorders are considered as missing data and not as the absence of the disorder, because electronic health records are frequently incomplete).

E. Architecture and implementation

We developed our system as a client-server web application in Python using Brython, a Javascript-compiled version of Python, for the client, and Owlready, a module for ontology-oriented programming [12]. The role of the server is limited to loading the decision trees from the ontology and serializing them to the client, and most of the program is implemented in the client, allowing patient data to remain on the client and thus supporting data privacy.

III. Application to Covid-19

A. Clinical algorithm conception

Three scenarios were considered: (1) a phone hotline receiving a call from a patient with confirmed or suspected Covid-19, (2) the home care management of a Covid-19 patient, including oxygen therapy if needed, and (3) the hospitalization of a Covid-19 patient. A first version of the clinical algorithms was elaborated after a thorough synthetic review of updated practice guidelines stemming from 4 recognized institutions (World Health Organization [13], [14], Centers for Disease Control and Prevention - USA [15], International Society for Infectious Diseases - USA [16], Haute Autorité de Santé - France [17]).

Then, this version was verified for validity and terrain applicability by a multidisciplinary panel of experts including two internal medicine physicians with ground experience in treating Covid-19 patients, an experienced nursing leader, a senior patient safety officer and a healthcare quality specialist with experience in infection control and medical informatics.

Clinical algorithms were produced as informal decision trees, representing the stepwise procedures for clinical decision-making about the evaluation and management of COVID19. One of the trees (hospital scenario) is indeed a multi-path tree. It has 35 nodes, including 19 leaves, making it difficult to display without advanced visual techniques.

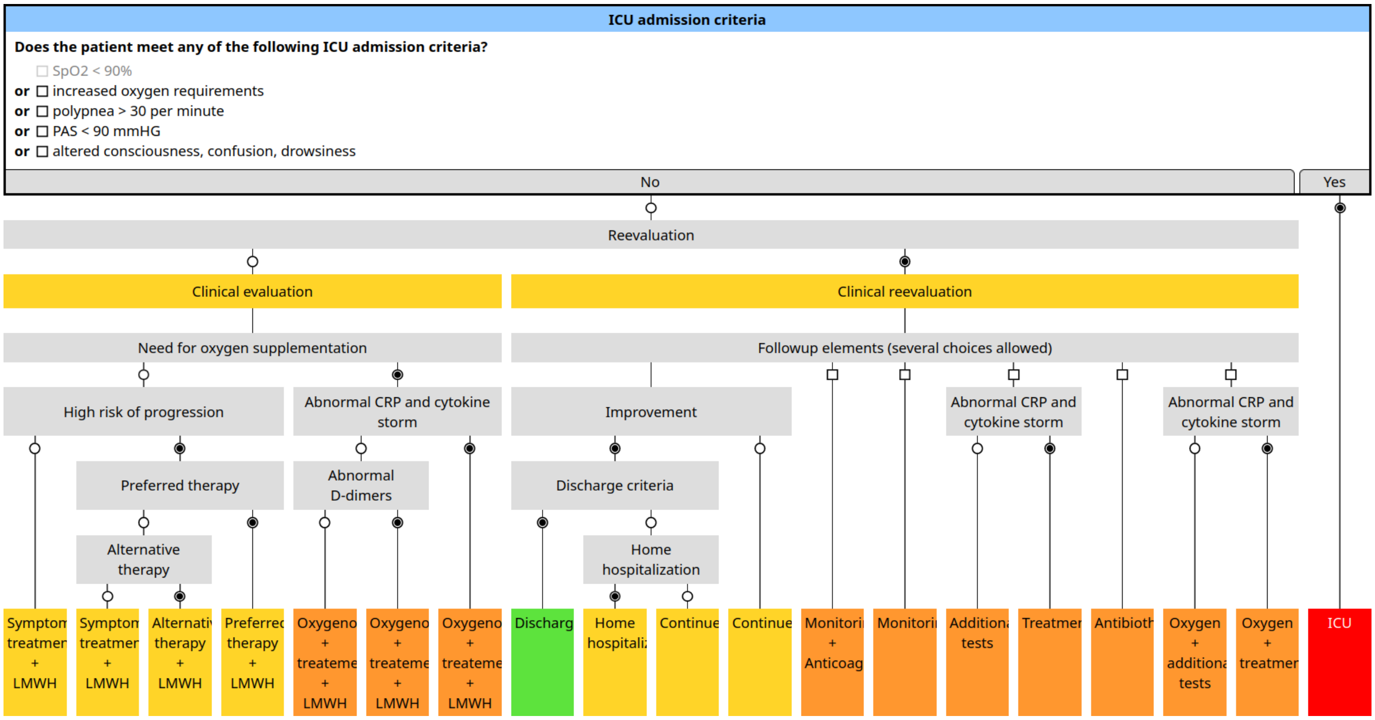

Figure 3. Interactive decision tree for the management of hospitalized Covid-19 patients, at the beginning of the user interaction.

B. Visualization of multi-path decision trees

Figure 3 shows the multi-path decision tree for hospitalization of Covid-19 patients, before user navigation. It gives an overview of the entire decision process, at a glance. Most nodes are yes/no questions, and use checked/unchecked radio buttons as symbols on the edge. To interact with the tree, the user can either click on the button at the bottom of a current node (*e.g.* "Yes" or "No"), or directly click on any node, for performing a faster or backward navigation.

Figure 1 shows the same decision tree, during user interaction. Inaccessible nodes are grayed and shrunken. The "Followup elements" node is a multiple-choice question, hence it remains current and opened even after selecting one of its children (other than "None"). Here, the user selected both "D-dimers increased" and then "CRP increased". This is an example of multi-path navigation, leading to the selection of more than one leaf. The user can continue the navigation further by answering the "Abnormal CRP and cytokine storm" question, or by selecting a third answer in the "Followup elements" node; selecting "None" will automatically deselect all other answers.

If available, patient data are taken into account for automatic navigation: the tree automatically select the appropriate child if possible. Automatic navigation is disabled when clicking on nodes for faster navigation, allowing the user to go backward in case of disagreement. In addition, true and false criteria are visualized via checkboxes and gray color, *e.g.* in Figure 1, we can see that CRP is above 10 but IL-6, LDH and fibrinogen are normal.

The interactive decision tree can be tested online at this address: www.lesfleursdunormal.fr/appliweb/orient_covid (NB only the hospitalization tree, #6, has been translated into English yet). The website includes a form for optional patient data entry, for demo purpose. Ideally, patient data are intended to be extracted from electronic health records.

C. Preliminary evaluation

The implemented trees were validated in terms of content and clinical logic by the two internists involved in the conception phase. In order to take into account the different temporalities relative to the clinical management decisions, the home care and the hospitalization trees were each split in two: one for initiation (*i.e.* the first encounter with the patient), and one for reevaluation.

The visual trees were presented individually and in separate encounters to 6 physicians and healthcare professionals not involved in the conception phase (2 senior internal medicine physicians, one senior infectious disease physician, one senior emergency physician, one junior physician and one infection prevention specialist). Qualitative remarks regarding the usability of the visual trees were collected, as well as the SUS (System Usability Scale) score. SUS has been chosen because it is more reliable and detects differences at smaller sample sizes than other questionnaires [18].

The mean SUS score was 92.5%, which is "excellent" according to the SUS scale. In the qualitative comments, the system was described as "user-friendly" (4 times), "good and clear visuals", "simple and practical". 5 professionals said it can improve adherence to guidelines. They also suggested uses for education, adaptation to a mobile app and application to other guidelines. Finally, they identified points of improvements, such as adding medication doses in recommendations, increasing the space devoted to focus nodes, or merging all trees in a single big one, and two potential risks: over-reliance on the system and anchoring bias.

IV. Discussion and conclusion

In this paper, we proposed two solutions for permitting the visualization of large decision trees: the extension of the classical decision tree model with multi-path support, and the use of fisheye and details-on-demand. We combined both solutions in a web tool for the interactive visualization of decision trees. We successfully applied this approach in a clinical decision support system for the management of Covid-19 patients at various levels, which present to clinicians guidelines recommendations using trees. A preliminary evaluation showed that it was well accepted by clinicians.

Indeed, many guidelines include informal decision trees, some of them being multi-path in spirit, although not formalized as such. Consequently, clinicians are used to decision trees. To our knowledge, this work is the first to propose formal multi-path decision trees for presenting guidelines. In the literature, multi-path event trees were proposed [19], but with a different semantics. A common approach for viewing decision tree is to display the entire tree in a panel, and the details of the current node in another panel, *e.g.* in [7]. However, this is not adapted for multi-path trees, where there can be several current nodes at a time.

We initially aimed at the automatic execution of the tree. However, we found during the clinical algorithm conception that many pieces of patient data are unlikely to be coded in electronic records, such as the list of symptoms expressed by the patient (*e.g.* rhinorrhea). Thus, we opted for partial automatic execution, limited to the steps for which patient data can reasonably be expected to be available, and we focused on the visual presentation of the tree.

Further evaluations are needed, in order to assess its usability more in depth, but also to evaluate it in terms of chance of erroneous navigation and time gain for clinicians. The semi-automatic navigation, taking into account structured patient data available, also has to be connected to electronic health records from hospitals, and properly evaluated.

The perspectives of this work include the adaptation of the multi-path decision tree model to other guidelines but also to decision trees from machine learning, its application to larger trees, the design of a user-friendly tool for authoring decision trees, and the clinical validation of the system.

V. Acknowledgments

This work was funded by the French Research Agency (ANR) and the Lebanese national research center (CNRS-L) through the Orient-Covid project.
[Grant No. ANR-21-LIBA-0004]

References

[1] H. Woolf S, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw, "Clinical guidelines: potential benefits, limitations, and harms of clinical guidelines," *BMJ (Clinical research ed.)*, vol. 318, no. 7182, pp. 527–30, 1999.

[2] A. Wlodyka, R. Mlynarski, G. Ilczuk, E. Pilat, and W. Kargul, "Visualization of decision rules - from the cardiologist's point of view," *Computers in Cardiology*, vol. 35, pp. 645–648, 2008.

[3] B. Johnson and B. Shneiderman, "Treemaps: a space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the 2nd International IEEE Visualization Conference*, San Diego, 1991, pp. 284–291.

[4] G. G. Robertson, J. D. MacKinlay, and S. K. Card, "Cone trees: animated 3D visualizations of hierarchical information," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91)*, 2001, pp. 189–194.

[5] A. H. I. Pires, R. S. A. D. Lima, C. G. R. dos Santos, B. S. Meiguins, and A. G. M. Soares, "A summarization glyph for sets of unreadable visual items in treemaps," in *International Conference Information Visualisation (iV)*, 2020.

[6] B. Séroussi, J. Bouaud, D. Sauquet, P. Giral, P. Cornet, H. Falcoff, and J. Julien, "Why GPs do not follow computerized guidelines: an attempt of explanation involving usability with ASTI guiding mode." *Stud Health Technol Inform*, vol. 160, no. Pt 2, pp. 1236–40, 2010.

[7] J. N. Babione, W. Ocampo, S. Haubrich, C. Yang, T. Zuk, J. Kaufman, S. Carpendale, W. Ghali, and G. Altabbaa, "Human-centred design processes for clinical decision support: A pulmonary embolism case study," *Int J Med Inf*, vol. 142, p. 104196, 2020.

[8] J. Mrva, S. Neupauer, L. Hudec, J. Ševcech, and P. Kapec, "Decision support in medical data using 3D decision tree visualisation," in *IEEE International conference on E-Health and Bioengineering (EHB 2019)*, 2019.

[9] D. Williams, D. Mudali, H. Buddelmeijer, P. Noorishad, S. Meles, R. Renken, K. Leenders, E. Valentijn, and J. Roerdink, "Visualization of decision tree state for the classification of Parkinson's disease," pp. 25–41, 2016.

[10] H. Guo and M. Fan, "Multi-path Decision Tree," in *International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, 2013, pp. 1411–1413.

[11] Y. Freund and L. Mason, "The alternating decision tree learning algorithm," in *International Conference on Machine Learning (ICML)*, vol. 99, 1999, pp. 124–133.

[12] Lamy JB, "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies," *Artif Intell Med*, vol. 80, pp. 11–28, 2017.

[13] World Health Organization, *Clinical management of COVID-19: living guideline*, 2023.

[14] ——, *Clinical care for severe acute respiratory infection: toolkit: COVID-19 adaptation*, 2022.

[15] Centers for Disease Control and Prevention, *Phone advice line tool : COVID-19 : recommendations for children (2-17 years) or adults (18+ years) with possible COVID-19*, 2022.

[16] National Institutes of Health, "COVID-19 Treatment Guidelines Panel. Coronavirus Disease 2019 (COVID-19) Treatment Guidelines," 2022.

[17] Haute Autorité de Santé, *Réponses rapides dans le cadre de la COVID-19 - Prise en charge de premier recours des patients atteints de Covid-19*, 2022.

[18] J. R. J. Lewis and J. Sauro, "Revisiting the factor structure of the system usability scale," *Journal of User Experience*, 2017.

[19] S. Kim and J. C. Lee, "Management of loss of offsite power avoiding reactor trip," p. 107248, 2020.