Using Eye-tracking to Detect Search and Inference During Process Model Comprehension

Amine Abbad-Andaloussi[1], Clemens Schreiber[2], and Barbara Weber[1]

[1] University of St. Gallen, St. Gallen, Switzerland
{amine.abbad-andaloussi,barbara.weber}@unisg.ch
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
clemens.schreiber@kit.edu

Abstract. Understanding process models involves different cognitive processes. These processes typically manifest in users' visual behavior and thus can be captured using eye-tracking. In this paper, we focus on the detection of two very essential behaviors: information search and inference. Using a set of eye-tracking features allowing to discern these two behaviors, we train several machine learning (ML) models to predict whether the user is involved in a search phase or an inference one. Following a cross-validation approach inspired by the leave-one-out method, our ML models attain 85% precision, 82% recall, and an F1 score of 80%. The outcome of this work enables the creation of novel adaptive systems, detecting whether the user is involved in a search or inference phase and accordingly providing adequate support. Moreover, it opens up new opportunities to better understand how different process model, tool, user and task-related factors affect users' search and inference behaviors.

Keywords: Process model comprehension, eye-tracking, search behavior, inference behavior, machine learning

1 Introduction

Comprehending process models is essential for performing a wide range of technical and managerial activities [14,44]. For the former, tasks such as maintenance, process redesign, and enhancement rely heavily on the ability to understand the existing models [44]. On the managerial side, understanding these models is vital for eliciting requirements and enhancing communication between IT professionals and domain specialists [23,14]. With the rapid pace at which processes nowadays have to change to cope with evolving users' needs, modelers must maintain a comprehensive understanding of existing process models to effectively enhance their capabilities. Additionally, although process models can be created automatically through techniques such as process mining [36] or generative artificial intelligence (AI) [13], the need to understand these automatically generated models and to ensure their correctness is still crucial.

The literature on process model comprehension can be divided into two streams. The former covers studies investigating the factors influencing the un-

derstanding of process models (e.g., [31,10,39,6,1,43,27,33,44,34]), while the latter proposes approaches designed to support the comprehension of these models (e.g., [24,32,40,8]). Our work extends this research stream by laying the foundations for novel adaptive systems that provide context-specific support to users in real-time. Specifically, we focus on the detection of two distinct behavioral phases during process model comprehension: *search* and *inference*. Search denotes the identification and separation of relevant from non-relevant information [41], while inference refers to the creative process of deriving new knowledge following the recognition of task-relevant information [25]. Providing context-specific support during process comprehension tasks depends highly on the behavioral phase a user is involved in. For information search, the user may need extra guidance towards the information relevant to solving the task at hand, whereas for inference processes, showing relevant concepts and additional contextual information would be more beneficial.

To detect search and inference, we train new machine learning (ML) models and evaluate their ability to distinguish these two behaviors. Our first research question can be formulated as follows *RQ1. To what extent can we detect users' search and inference behaviors from eye-tracking data during process model comprehension tasks?* Subsequently, we investigate the importance of each of the used eye-tracking measures in estimating users' behavior by answering the following research question *RQ2. What eye-tracking features are important for inferring users' search and inference behaviors?* Our ML models form the core component of adaptive systems that can support users in real-time. Therefore, the training and evaluation of these models, including the investigation of the most predictive features denote the primary focus of this paper.

In a nutshell, throughout this work, we conduct an eye-tracking study where we instruct participants to perform search and inference in a specific order during comprehension tasks. Such an approach provides us with the ground truth denoting phases where users are either conducting search or inference. This ground truth (i.e., serving as labels) is used together with the collected eye-tracking measures (i.e., serving as features) in the training of our ML models following a supervised ML approach. Then, to evaluate our ML models' capability to detect search and inference, we use a special cross-validation approach inspired by the leave-one-out method. Therein, the ML models are tested with eye-tracking data from participants and tasks not previously used in the training set. As a result, our ML models attain 85% precision, 82% recall and a 80% F1 score. Moreover, our findings demonstrate a trade-off between the performance of the ML models and the window length at which the selected eye-tracking features are computed.

Our work has diverse implications in both online and offline settings. In the former, upon the detection of users' search and inference behavioral phases, context-adaptive systems can react in real-time to provide targeted support to users based on the phase they are involved in. In the latter, detecting users' search and inference phases can help to delve deeper into their characteristics and studying how different model, task, user and tool-related factors affect these behavioral phases. The remainder of this paper is structured as follows. Sect. 2

and Sect. 3 introduce the background and related work respectively. Sect. 4 explains our research method. Sect. 5 and Sect. 6 present and discuss the findings respectively. Sect. 7 summarizes this work and sets the path for future research.

2 Background

This section describes the theoretical underpinnings relevant to our study. Sect. 2.1 presents the theories underlying search and inference behaviors respectively. Sect. 2.2 introduces eye-tracking and presents the measures allowing to differentiate search and inference behaviors.

2.1 Search and Inference

Search and *inference* manifest with distinct behaviors during comprehension tasks [24]. Search involves distinguishing relevant from irrelevant information [41]. In process comprehension, this translates into identifying activities in the process model that are either *task-relevant* or *task-irrelevant* [31]. Search activities consume valuable resources including time and cognitive effort. Thus, users typically cease once they believe they have acquired sufficient information to complete the given task [11]. In tasks involving process model comprehension, this means a user will stop the search process once they identify all the task-relevant process activities, assuming that these activities are clearly identifiable in the process model [11]. *Cognitive stopping rules* [12] are typically used in this context by users to evaluate the sufficiency of collected information and upon that decide when to stop the search. For tasks that are structured and can be broken down into parts, users commonly rely on a particular stopping rule known as *mental lists*[11]. This rule consists of maintaining a mental checklist of necessary elements that must be located before ending the search [12]. For instance, when a user is asked to determine how some activities within the process model interconnect, then relying on this stopping rule implies holding a mental list of the activities that must be identified in the model before investigating how they interconnect. Once this list of activities is identified, the search phase is complete.

Inference involves generating new insights through the creative process of interpreting the identified relevant information [25]. This process is closely linked to reasoning, where the integration of newly extracted information with pre-existing knowledge takes place [24].

Search and inference have been explored within various cognitive frameworks. For example, Kim et al. [24] identified *perceptual* and *conceptual* processes as two main processes involved in understanding diagrammatic representations such as graphical models. The perceptual process (related to information search) focuses on the search and recognition of pertinent information, whereas the conceptual process (related to inference) is concerned with reasoning and the derivation of insights from the models. Similarly, according to multimedia learning theory [28], after relevant information is selected during the search phase, it is organized and integrated with existing knowledge in the inference phase.

Our study aims at discerning search and inference using the eye-tracking measures that will be introduced in Sect. 2.2. Throughout the design and execution of our study, we formulate our experiment tasks in such a way that the task-relevant information can be clearly recognized in the process models. Also, we instruct our participants to continue with search until they have found all task-relevant information to foster the mental list stopping rule [12]. These design decisions are meant to facilitate the separation between search and inference behavioral phases in the users' data as will be explained in Sects. 4.1 and 4.2.

2.2 Using Eye-tracking to Detect Search and Inference Behaviors

Eye-tracking has many applications. Notably, it allows detecting visual and behavioral patterns, which might otherwise, not be clear based on verbal protocols [22]. In the field of process modeling, eye-tracking is well-established as it has been successfully used in numerous studies investigating different aspects associated with users' visual behavior (e.g., [10,37,32,19,42,3,34,35,15]). Among the popular eye-tracking concepts referred to in this field are: *fixations*, *saccades*, *areas of interest (AOI)* and *scan-paths*.

A fixation occurs during an interval characterized by eye movements of very low velocity, indicating that the pupil is still on a particular point within the visual field [22]. A saccade refers to rapid eye movements that indicate the pupil is transitioning from one position (i.e., a fixation) to another within the visual field [22]. Areas of interest can be defined in a stimulus (e.g., process model), to analyze how often a specific area (e.g., a process model activity) is entered, left and revisited [22]. Lastly, a scan-path denotes a sequence of fixations or visits to AOIs, reflecting the visual path followed by the user when engaging with an artifact (e.g., a process model) [22].

The literature comprises a number of eye-tracking measures that can be used to differentiate search and inference behaviors. The *average fixations duration* calculates the mean duration of fixations within a specified time frame [22]. This measure is typically lower during information screening and scanning as part of a search process, than when performing mental processing as part of an inference process [22]. In [17], Glöckner et al. assigned fixed thresholds to *short fixations* (of duration $< 250ms$) and *long fixations* (of duration $>= 500ms$) to differentiate superficial processing common in information search from deep mental processing associated with inference processes. Accordingly, authors in the literature have split their fixations into short and long (following Glöckner thresholds) to study the underlying cognitive processes [37].

Besides fixations, saccades can also provide interesting insights allowing to differentiate search and inference behaviors. The *average saccade amplitude* calculates the mean distance that saccades cover over a specific time frame [22]. This measure tends to decrease when users are deeply engaged in a thorough inspection of an object [22].

Scan-path precision is another key measure distinguishing between search and inference behaviors. It calculates a ratio that divides the number of fixations landing on the task-relevant AOIs (e.g., process model activities) over the total

number of fixations on the entire artifact [31]. Scan-path precision reveals the degree to which users concentrate on task-relevant activities [31]. During the search phase, users are engaged in identifying the task-relevant activities from a broader array of irrelevant ones. Consequently, their fixations may land on both relevant and irrelevant activities of the process model, which results in a lower scan-path precision ratio. In contrast, after identifying all relevant activities, users tend to focus exclusively on this set of activities, during the inference phase. This, in turn, leads to significantly higher scan-path precision.

Following these theoretical underpinnings, we use the average fixation duration, proportions of short and long fixations (to all fixations), average saccade amplitude and scan-path precision computed over specific time windows to differentiate search and inference behaviors in our eye-tracking data (cf. Sect. 4.3).

3 Related Work

In the literature, several authors have investigated users' behavior to understand the strengths and pitfalls associated with different process model representations (e.g., [31,10,39,6,1,43]) and task types (e.g., [27,33,44,35]). As a result, a number of insights (e.g., [10,33,35]), guidelines (e.g., [31,43]) and frameworks (e.g., [44,1,39,27]) have emerged within this research stream, notably on users' reading strategies [31], visual routines [39], attention distribution [31,6], cognitive integration [10] and task performance [27,33,35]. In turn, another stream of research focusing on developing techniques to support process model comprehension has emerged (e.g., [24,32,40,8]). Our study fits within this latter stream of research. In this vein, for instance, Kim et al. [24] showed that both perceptual and conceptual integration processes (related to search and inference respectively, cf. Sect. 2.1) can be facilitated by visual cues and contextual information. Visual cues are applied to emphasize the relation between the elements in different diagrams, and contextual information provides a general overview of the dependencies between multiple diagrams within a system. Another support mechanism, also based on visual cues, was tested by Winter et al. [40]. In their study, novices were provided with visual guidance based on experts' eye movements recorded while they were engaging with process models. In this way, the novices' attention was guided toward the relevant areas of the process models which affected positively their comprehension of the models at hand. From a task-perspective, Petrusel et al. [32] used task-specific visual cues to facilitate process model understanding. The applied visual cues consisted of coloring the task-relevant elements and doing layout adjustments. Both visual cues had a positive impact on task performance. Dynamic visualization techniques such as animations were also investigated for their support in problem-solving tasks involving process models. As shown in a study by Aysolmaz and Reijers [8], applying color transformations to indicate the status-changes of activities contributes to a better comprehension of process models.

As mentioned in Sect. 1, our work extends the existing research on supporting the comprehension of process models by establishing the basis for novel adap-

tive systems that can provide context-specific support based on users' behavior. This foundation is achieved through the development of ML models capable of discerning whether a user is engaged in information search or inference. Once deployed, these models could infer users' behavior and thereby enable more targeted support strategies tailored to fulfil specific user needs.

4 Research Method

4.1 Study Design

Material. The experiment involves a series of *model fragments* that represent various aspects of a logistics process and a collection of *comprehension tasks* designed to prompt participants to engage with these fragments. Since in real-world settings, systems comprise multiple components, which are modeled separately [24], we selected a process model consisting of multiple fragments. Hence, the comprehension tasks could refer to single fragments of the process model or to different fragments. This way, it is possible to observe extensive search behavior since the relevant information for the comprehension tasks could be distributed across multiple fragments. To further emphasize the need for inference, we decided to use the fragment-based modeling approach introduced in [21]. This approach is well suited to model systems, consisting of multiple components. However, in some circumstances, it requires extensive mental effort to infer information distributed over several fragments [34].

In a nutshell, the fragment-based modeling approach [21] uses a subset of the standard BPMN notation [30]. It allows to model separate process fragments, which are implicitly connected by data objects. A data object is always described by its name and its state. If a data object is assigned as an input for a task, this task can only be executed if the data object is in the required state. If a data object is assigned as an output of a task, the task execution leads to a state change of the data object. All possible state changes and their interdependence are additionally depicted in a data-object lifecycle [21].

Using the fragment-based approach, our logistics process is segmented into six process model fragments. These fragments are linked through three data objects, for which the state changes are modeled in their respective life cycles. A portion of this process is illustrated in Fig. 1, while the full process is available in our online appendix[3]. The design of the process fragments adheres to established process modeling guidelines [9,29], ensuring that each fragment has a well-organized layout and a reasonable number of activities and gateways (ranging from two to four parallel or exclusive gateways and six to eight activities). The fragments have overall comparable essential and accidental complexity levels [3,7]. To minimize the influence of specialized domain knowledge, activity names are phrased in layman's terms.

The experimental setup includes eight tasks that ask the participants to examine relationships (such as sequence flow, repetition, exclusiveness, or concurrency) between process activities either located inside the same process model

[3] See `https://github.com/aminobest/COOPIS2024BehaviorDetection`

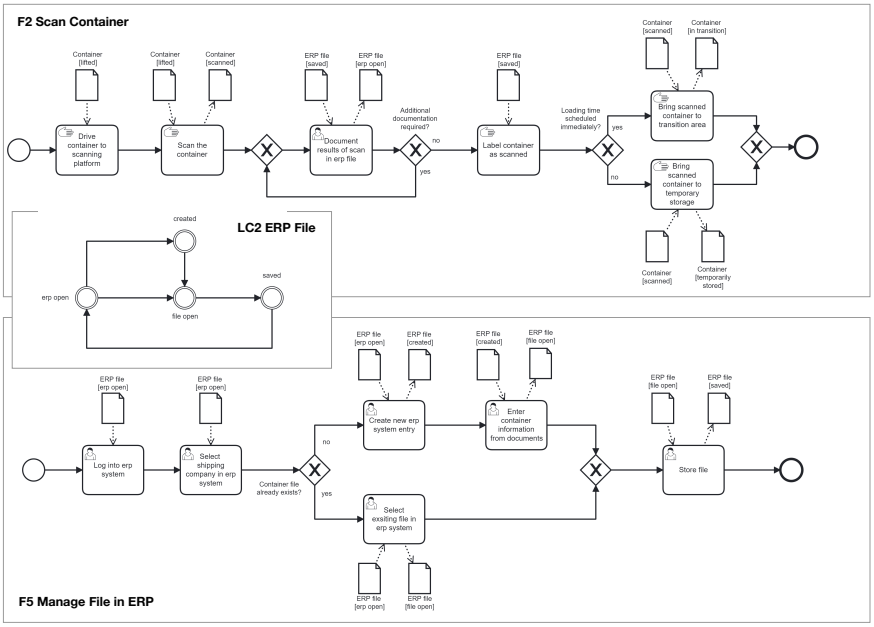

Fig. 1. A fraction of the process model used in the experiment (i.e., designed using the fragmented-based approach [21]). **The complete model and a high-resolution version of this figure can be accessed in our online appendix**[3].

fragment (termed *local tasks*) or distributed over multiple fragments (termed *global tasks*). This design allows the tasks to capture various workflow patterns at both local and global scales, thereby reflecting realistic process model comprehension scenarios. The tasks were presented as statements written based on the template shown in Fig. 2. The participants were asked to evaluate the correctness of these statements considering the process behavior encoded in the model fragments. In the statements, the names of the activities relevant to solving the tasks were put into quote marks to facilitate their recognition in the process model fragments during the search phase (cf. Sect. 2.1). Additionally, to minimize any potential learning effects, each task was designed to focus on different aspects of the model fragments used. The materials deployed in our experiment are available online[3].

Participants. The experiment covered 46 participants (22 from the University of St Gallen, 17 from Karlsruhe Institute of Technology, 4 from the research institute Forschungszentrum für Informatik FZI in Karlsruhe and 3 from Promatis i.e., a German IT company). The participants ranged in age from 20 to 50 years, with the majority (63%) falling within the 20-30 age bracket. They came from diverse backgrounds: 22 were engaged in academic research, 17 were students at various stages of their bachelor's and master's degree programs, and 7 worked

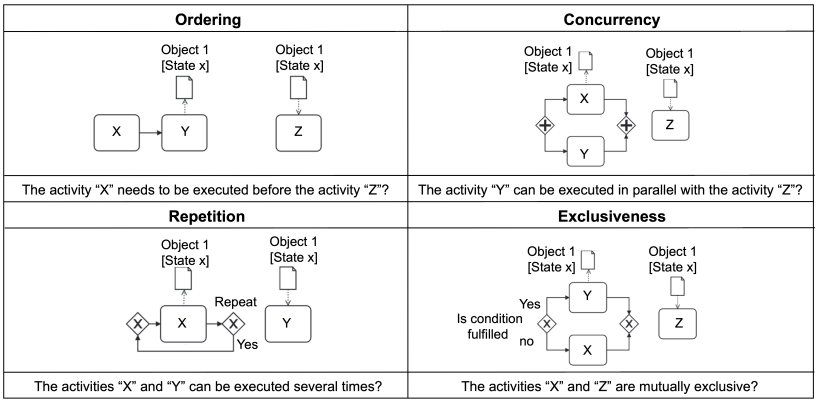

Fig. 2. Task template used in the experiment. **A higher resolution of this figure is available in our online appendix**[3].

in the IT industry. Regarding their familiarity with BPMN, as rated on a scale from 1 (unfamiliar) to 7 (very familiar), 48% of the participants reported high familiarity (scores of 5 to 7), while 42% indicated low familiarity (scores of 1 to 3). To prepare the participants for the experiment, they were all uniformly trained on the BPMN concepts used in this study and the fragment-based approach. Before beginning the main experiment, their comprehension was assessed using a set of test tasks similar to those they would encounter during the experiment. All of them passed the test tasks. Detailed demographic information about the participants is available in our online appendix[3].

4.2 Experiment Procedure

The experiment was conducted in individual eye-tracking lab sessions, each lasting approximately one hour. All participants provided signed consent for their involvement and all collected data was anonymized to ensure no identifiable information could be traced back to the individuals. Initially, a familiarization phase introduced each participant to the relevant BPMN concepts and the fragment-based modeling approach [21]. Following this familiarization, a quiz was administered to assess their understanding of the covered concepts and thus confirm their readiness for the experiment. The participant, then completed screening and demographic forms to respectively ensure their physical ability to participate in an eye-tracking experiment and to gather basic demographic data (e.g., gender, age range, familiarity with the concepts). Before data collection started, each participant was positioned in front of the eye-tracking device. They received instructions on the data collection process, including a directive to minimize head movements, and then underwent device calibration to accurately track their gazes. Each participant was also instructed against forming an overarching understanding of the entire process model (including the six fragments and three life cycles) from the first task. Instead, they were requested to approach

each task independently, focusing on identifying task-relevant activities (in the different models) first and then investigating their relationships. This approach was intended to reinforce the use of the mental list stopping rule during the search phase (cf. Sect 2.1) by prompting participants to first locate all relevant activities before deducing their connections in the inference phase. The data collection was conducted within the data collection framework EyeMind [4], using the research-grade Tobii Pro X3-120[4] eye-tracker. Therein, each participant was presented with a series of tasks in randomized order. After each task, they were asked to explain their answers and complete a self-assessment questionnaire to evaluate the task's perceived difficulty. For the current data analysis, only the eye-tracking data is analyzed.

4.3 Data Analysis

To address RQ1 (cf. Sect. 1) we train and test a set of ML models. Therein, we follow an inductive segmentation approach to differentiate between the search and inference phases in our data. This segmentation serves as a foundation for training our ML models to distinguish these phases effectively. We then assess the performance of these models across various scenarios. Afterward, to address RQ2 (cf. Sect. 1), we examine the extent to which each of the used eye-tracking features contributes to the predictions of the trained ML models.

Data segmentation approach (RQ1). An overview of this analysis is illustrated in Fig. 3. (1) The process begins by organizing the collected eye-tracking data into distinct trials. Each trial encompasses the data from one participant engaging with a specific task (for example, participant P10 performing task T3, participant P11 performing task T3). (2) Next, a specific point, referred to as a cut-mark, is determined within each trial. This cut mark corresponds to the point in time when the participant has located all the task-relevant activities. The cut marks were automatically derived through the automated detection of AOIs (corresponding to process activities in this context) provided by EyeMind [4], along with supplied information about which activities were relevant to each specific task. (3) The trial is then segmented into two parts: the segment before the cut-mark (i.e., Phase 1) and the segment after the cut-mark (i.e., Phase 2). (4) Subsequent to this segmentation, the eye-tracking measures presented in Sect. 2.2 are calculated for each segment. As depicted in Fig. 3, comparisons between Phase 1 and Phase 2 reveal that Phase 1 exhibits shorter average fixation durations, a higher ratio of short fixations, a lower ratio of long fixations, reduced scan-path precision and smaller average saccade amplitudes than Phase 2. (5) Using the Wilcoxon signed-rank paired test (which compares paired data without imposing normality requirements), significant statistical differences are confirmed between the measures in Phase 1 and Phase 2. (6) Considering these observed trends in the eye-tracking measures, in line with the theoretical

[4] See `https://go.tobii.com/X3UM`

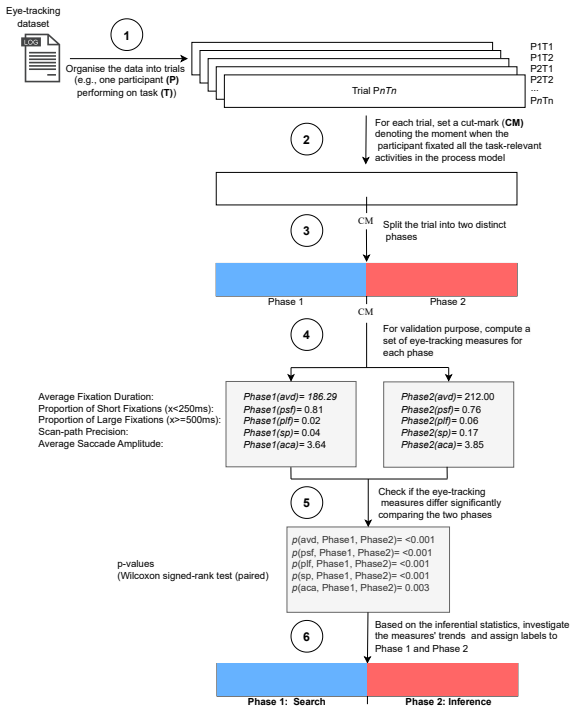

Fig. 3. The inductive segmentation approach. **A higher resolution of this figure is provided in our online appendix**[3].

foundations introduced in Sect. 2.2, and taking into account the instructions given to the participants (cf. Sect. 4.2), it is reasonable to infer that Phase 1 likely represents search behavior, while Phase 2 represents inference behavior. It is worthwhile to mention that while the inferential statistics validate the assumption that the participants generally followed the given instruction by first conducting search and then inference, there might have been instances where this instruction was not strictly adhered to. This threat to validity is discussed in Sect. 6.

ML Training and Benchmarking (RQ1). This part aims at developing and benchmarking ML models predicting whether the user is conducting search or inference based on the eye-tracking measures introduced in Sect. 2.2. In the context of adaptive systems, a key requirement for these predictions would lie in the ability of the ML models to infer users' behavior with low latency. In turn, this would enable timed online support depending on whether the user is searching for information or inferring insights from the process models (cf. Sect. 1). To evaluate the extent to which one can go fine-grained in time, while

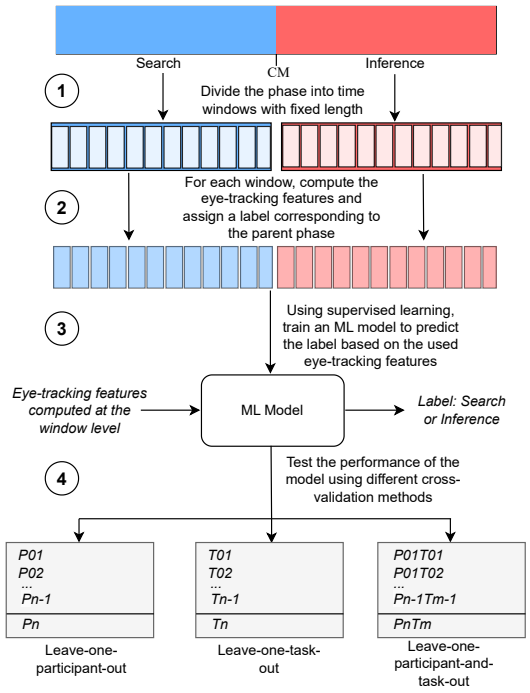

Fig. 4. Model training and testing. **A higher resolution of this figure is available in our online appendix**[3].

still having good model performance, we train and benchmark our ML models with features (i.e., eye-tracking measures) collected at window intervals ranging from 5 seconds to 60 seconds, with an increment of 5 seconds. Our ML training and benchmarking approach is summarized in Fig. 4. (1) Firstly, we segment the data of the search and inference phases into windows with a fixed length (e.g., 5 seconds). (2) Afterwards, for each window, we compute a set of features capturing the average fixation duration, proportion of short fixations and proportion of long fixations, scan-path precision and average saccade amplitude at the level of each window. Then, we assign each window the label of its parent phase (e.g., a window in the search phase will be assigned a *search* label), which is used in the next step to train the machine learning model. (3) Using a supervised learning approach, we train a classifier (i.e., a random forest implementation in Scikit-learn[5] with n_estimators=300 and max_depth=5) to predict the windows labels based on the given eye-tracking features. We choose random forests considering the enhanced performance of ensemble methods [20] and the ability of random forests to return the importance of each of the features used in their training.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

We rely on a cross-validation approach inspired by the leave-one-out method to test and benchmark the performance of our ML Models in different scenarios (i.e., similar to [16,5]). (4) The leave-one-participant-out approach consists of training the model with the data of all participants (in Set P) except one (i.e., $\{P_1...P_{n-1}\} \in P$) of whom the data is used to test the model (i.e., $P_n \in P$). Likewise, the leave-one-task-out approach consists of training the model with the data of all tasks (in Set T) except one (i.e., $\{T_1...T_{n-1}\} \in T$) of which the data is used to test the model (i.e., $T_n \in T$). Lastly, the leave-one-participant-and-task-out approach consists of training model with the data of all participants and all tasks (in Set $P \times T$), except one combination (i.e., $\{P_1T_1...P_{n-1}T_{m_1}\} \in P \times T$), which is used to test the model (i.e., $P_nT_m \in P \times T$). This cross-validation approach allows validating the performance of the model when predicting search and inference behaviors for new participants, new tasks and new participants performing new tasks for which the data were not used in the training of the model. As performance measures, we use precision, recall and F1 score (similar to [16]). The results of our benchmarking showing the performance of our ML models with data segmented in time windows of different lengths (in the range [5 seconds, 60 seconds]) are reported in Fig.5, Sect. 5.

Feature Importance (RQ2). This part aims at studying the extent to which each of the used eye-tracking measures contributes to the predictions of the trained ML models. To this end, we rely on the feature importance metric given by the used random forest implementation[5]. This metric computes the "*Mean Decrease Impurity (MDI)*" which refers to "the total decrease in node impurity [...] averaged over all trees of the ensemble [i.e., the random forest] [38]". We compute the feature importance metric in each cross-validation scenario (i.e., leave-one-participant-out, leave-one-task-out, leave-one-participant-and-task-out) and window length (5 to 60 seconds). Note that this computation assigns a ratio to each feature (i.e., eye-tracking measure) used in training the model, with these ratios collectively summing up to 1 in each individual analysis (with a different cross-validation scenario and window length). Since the feature importance values do not vary much across the different cross-validation scenarios, we average the values across all the scenarios and plot the aggregated results for each window length (cf. Fig.6, Sect. 5).
Our full analysis is documented in the Python Notebooks available in our appendix[3].

5 Findings

ML Models' Performance (RQ1). The performance of our ML models considering the different cross-validation scenarios introduced in Sect. 4.3 (i.e., leave-one-participant-out, leave-one-task-out, leave-one-participant-and-task-out) is shown in Fig. 5. For each scenario, the performance measures (i.e., precision, recall and F1 score) are reported over the different applied window sizes.

In all the cross-validation scenarios, there is a consistent trend showing that the performance of the ML models increases until it reaches a peak at a win-

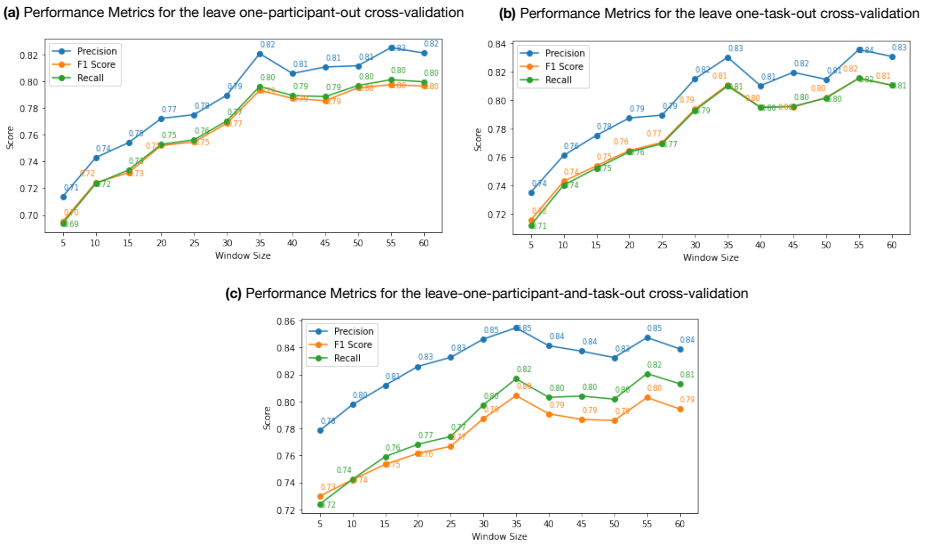

Fig. 5. Performance measures for the different cross-validation scenarios. **A higher resolution of this figure is available in our online appendix**[3].

dow size of 35 seconds. At this point, the best performance is observed in the ML model based on the leave-one-participant-and-task-out strategy, achieving a precision of 85%, a recall of 82%, and an F1 score of 80%. The performance metric values in other cross-validation scenarios (i.e., leave-one-participant-out and leave-one-task-out) remain nevertheless similar to one of the leave-one-participant-and-task-out scenario. Following this peak, the performance in terms of precision, recall, and the F1 scores remains relatively steady.

Features' Importance (RQ2). The importance scores of the features used in the training of our ML models are depicted in Fig. 6 together with the different window lengths at which these features were extracted. The scan-path precision is the most predictive feature (average importance ratio: 0.58), followed by the proportion of long fixations (0.17), then the average fixation duration (0.10), the average saccade amplitude (0.08) and finally the proportion of short fixations (0.05). From Fig. 6, it also emerges that generally the obtained importance scores do not vary much when changing the window length. Overall, these results highlight the importance of the scan-path precision and the proportion of long fixations in differentiating the search and inference behaviors in time windows of different lengths.

6 Discussion

The results presented in Sect. 5 show that our ML models can classify whether users are involved in a search or inference behavior. The performance, reflected by the precision, recall, and F1 scores, varies with window size. Notably, the

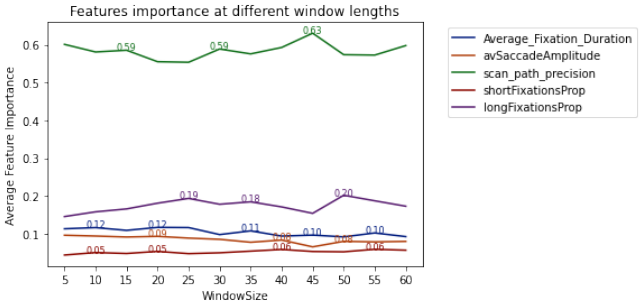

Fig. 6. Features importance at different window lengths. **A higher resolution of this figure is available in our online appendix**[3].

35-second mark reflects the peak point at which these metrics reach 85%, 82% and 80% respectively. Beyond this point, increasing the window size does not enhance the performance significantly (cf. Fig. 5). Therefore, a window size of 35 seconds can be optimal to reach a good performance when predicting whether a user is engaged in search or inference behavior. If our ML models are integrated into a system designed to detect users' behaviors based on eye-tracking data, for example, it would take at least 35 seconds to identify a user's behavior reliably. While it is possible to reduce this detection time, doing so would affect the ML model's performance. For instance, reducing the window to 5 seconds results in a performance of approximately 70% in terms of precision, recall, and F1 scores. This highlights a fundamental trade-off between window size and model performance. Hence, one should weigh the specific needs of their application when selecting the optimal window size and accordingly balance the demand for rapid detection of users' behavior with the necessity for robust predictions.

Regarding the importance of the features computed in the training of our ML models, the scan-path precision and the proportion of long fixations take the lead (cf. Fig. 6). Therefore, they represent the key predictors for search and inference phases during process model comprehension.

Threats to Validity. Our approach underlies a number of assumptions, which may threaten the validity of our study. Particularly, the inductive behavioral analysis explained in Sect. 4.3, assumes a clear cut between search and inference phases in the participants' behavior, such that they are first searching for all the task-relevant activities in the process model fragments and then inferring the answer to the given task. This assumption may not always hold true especially if people intertwine search and inference phases when engaging with process models. As this behavior would result in a more complex inductive data analysis and training of our ML models, we have opted for a simpler experiment design where (1) the tasks are formulated in a way that the relevant activities are marked explicitly in the text (in quotemarks, cf. Sect. 4.1) and (2) the participants are instructed to search for the relevant activities and then infer their relationship (cf. Sect. 4.2). This way, we enforced a clear cut between search and inference phases. Although there is no strict evidence that all the participants followed the

given instruction in all tasks, the inferential statistics (cf. Fig. 4) confirm that this pattern was followed to a large extent. At prediction time, our ML models are also expected to provide accurate predictions of the search and inference phases, if the cuts between the two phases are less clear as long as search and inference behavior do not overlap completely. Testing this hypothesis would require a follow-up eye-tracking study, where concurrent think-aloud techniques [22] can be used to triangulate participants' behavior with their insights.

Moreover, not all behavioral patterns associated with search and inference can be captured using our eye-tracking measures. For instance, people tend to fixate empty spaces when coordinating the visual image with their mental model [18]. This behavior can be incorporated within the inference phase as it reflects the process of integrating new information (extracted from the process model) with existing knowledge (as part of the mental model) to derive an answer to the given task (cf. Sect 2.1). However, the scan-path precision measure would suggest an inconsistent behavior since the fixations will not land on the task-relevant activities and thus the scan-path precision would be relatively low. To capture similar behavior accurately, a more fine-grained analysis is required which can be conducted as part of future work.

Another potential limitation is associated with the labels assigned to the windows at different time granularities. While the inferential tests confirm that the used eye-tracking measures significantly differentiate search and inference phases, these measures were computed at the whole phase level. Moving into time windows with reduced time granularity, we cannot guarantee that the windows are totally free of noise (e.g., partial search behavior in an inference phase). However, we expect this noise to be marginal and thus without significant impact on the goodness of the assigned labels, especially when the windows are large enough. It also remains to be tested, whether our ML models would perform adequately on other test sets, for instance, when users are performing search and inference on process models in other process modeling notations. Our study provides a good basis and starting point for inferring search and inference behaviors but more research is needed to ensure the reliability of our ML models in other settings.

Implications. Addressing the limitations pointed out in this section would enhance the performance of the proposed ML models, opening up a wide array of potential applications. In an online setting, detecting whether users are involved in information search or inference can help to provide timed and targeted support. For instance, the system can accordingly raise the saliency of the task-relevant parts in the process model to support the search process, or provide documentation and explain the relevant concepts to support the inference process. Therein, generative AI and large language models (LLMs) can be used to generate context-specific documentation and explanations. Besides process modeling, such a system can be useful in supporting software development tasks but could also find applications in e-learning platforms to help with information search and the inference of new insights when studying new material. In offline settings, the detection of search and inference behavioral phases can help isolate

them and study how they are influenced by different model and tool properties as well as various task types and user profiles. For instance, one can explore how novices and experts differ in their search and inference behavior, or investigate how different model representations (e.g., hybrid models [2]) or tool features (e.g., customized sub-process navigation [26]) affect the speed and accuracy of these behaviors.

7 Conclusion and Future Work

In this paper, we propose a set of ML models using eye-tracking features to differentiate search and inference phases in users' behavior when performing comprehension tasks on process models. The performance indicators demonstrate the ability of our ML models to detect these phases, while the analysis of features' importance suggests that the scan-path precision measure and the proportion of long fixations are the best predictors among the used eye-tracking features.

Following the reflections made in Sect. 6, a follow-up study capturing users' behavior in a more open setting where participants can freely intertwine search and inference phases is deemed important as future work to test our ML models in this different and more realistic environment. Therein, considering also various comprehension tasks and process modeling notations could contribute to more robust ML models. Additionally, by using concurrent think-aloud [22] in the new data collection, we can isolate search and inference phases more clearly and deepen our analysis of the underlying cognitive processes. Another important line of research would be to study the impact of different model, tool, task and users' characteristics on search and inference phases. Our data allows us, particularly, to delve into the impact of task type since our experiment material covers both local and global tasks (cf. Sect. 4.1).

References

1. Abbad-Andaloussi, A.: A Framework for Enhancing the Modeling and Comprehension of Declarative Process Models. Ph.D. thesis (2021)
2. Abbad-Andaloussi, A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study. Information Systems **91**, 101505 (2020)
3. Abbad-Andaloussi, A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: Complexity in declarative process models: Metrics and multi-modal assessment of cognitive load. Expert Systems with Applications **233**, 120924 (2023)
4. Abbad-Andaloussi, A., Lübke, D., Weber, B.: Conducting eye-tracking studies on large and interactive process models using eyemind. SoftwareX **24**, 101564 (2023)
5. Abbad-Andaloussi, A., Soffer, P., Slaats, T., Burattin, A., Weber, B.: The impact of modularization on the understandability of declarative process models: a research model. In: Information Systems and Neuroscience (NeuroIS). Springer (2020)
6. Abbad-Andaloussi, A., Zerbato, F., Burattin, A., Slaats, T., Hildebrandt, T.T., Weber, B.: Exploring how users engage with hybrid process artifacts based on declarative process models: a behavioral analysis based on eye-tracking and think-aloud. Software and Systems Modeling **20**, 1437–1464 (2021)

7. Antinyan, V.: Evaluating essential and accidental code complexity triggers by practitioners' perception. IEEE Software **37**(6), 86–93 (2020)
8. Aysolmaz, B., Reijers, H.A.: Animation as a dynamic visualization technique for improving process model comprehension. Information & Management **58**(5), 103478 (2021)
9. Becker, J., Rosemann, M., Von Uthmann, C.: Guidelines of business process modeling. In: Business Process Management: Models, Techniques, and Empirical Studies, pp. 30–49. Springer (2002)
10. Bera, P., Soffer, P., Parsons, J.: Using eye tracking to expose cognitive processes in understanding conceptual models. MIS Quarterly **43**(4), 1105–1126 (2019)
11. Browne, G.J., Pitts, M.G.: Stopping rule use during information search in design problems. Organizational Behavior and Human Decision Processes **95**(2) (2004)
12. Browne, G.J., Pitts, M.G., Wetherbe, J.C.: Cognitive stopping rules for terminating information search in online tasks. MIS quarterly pp. 89–104 (2007)
13. van Dun, C., Moder, L., Kratsch, W., Röglinger, M.: Processgan: Supporting the creation of business process improvement ideas through generative machine learning. Decision Support Systems **165**, 113880 (2023)
14. Figl, K.: Comprehension of procedural visual business process models: a literature review. Business & Information Systems Engineering **59**, 41–67 (2017)
15. Franceschetti, M., Abbad-Andaloussi, A., Schreiber, C., A. López, H., Weber, B.: Exploring the cognitive effects of ambiguity in process models. In: International Conference on Business Process Management. Springer (2024)
16. Fritz, T., Begel, A., Müller, S.C., Yigit-Elliott, S., Züger, M.: Using psychophysiological measures to assess task difficulty in software development. In: Proceedings of the 36th international conference on software engineering (2014)
17. Glöckner, A., Herbold, A.K.: Information processing in decisions under risk: Evidence for compensatory strategies based on automatic processes. MPI collective goods preprint (2008/42) (2008)
18. Grant, E.R., Spivey, M.J.: Eye movements and problem solving: Guiding attention guides thought. Psychological Science **14**(5), 462–466 (2003)
19. Gulden, J., Burattin, A., Andaloussi, A.A., Weber, B.: From analytical purposes to data visualizations: a decision process guided by a conceptual framework and eye tracking. Software and Systems Modeling **19**, 531–554 (2020)
20. Han, J., Pei, J., Tong, H.: Data mining: concepts and techniques. Morgan kaufmann (2022)
21. Hewelt, M., Weske, M.: A hybrid approach for flexible case modeling and execution. In: Business Process Management Forum: BPM Forum 2016, Rio de Janeiro, Brazil, September 18-22, 2016, Proceedings 14. pp. 38–54. Springer (2016)
22. Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., van de Weijer, J.: Eye Tracking: A comprehensive guide to methods and measures. OUP Oxford (2011)
23. Indulska, M., Green, P., Recker, J., Rosemann, M.: Business process modeling: Perceived benefits. In: Conceptual Modeling - ER conference. Springer (2009)
24. Kim, J., Hahn, J., Hahn, H.: How do we understand a system with (so) many diagrams? cognitive integration processes in diagrammatic reasoning. Information Systems Research **11**(3), 284–303 (2000)
25. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive science **11**(1), 65–100 (1987)
26. Lübke, D., Ahrens, M.: Towards an experiment for analyzing subprocess navigation in bpmn tooling (2022)

27. Mandelburger, M.M., Mendling, J.: Cognitive diagram understanding and task performance in systems analysis and design. MIS Quarterly **45**(4), 2101–2157 (2021)
28. Mayer, R.E.: Human nonadversary problem solving. Human and machine problem solving pp. 39–56 (1989)
29. Mendling, J., Reijers, H.A., van der Aalst, W.M.: Seven process modeling guidelines (7pmg). Information and software technology **52**(2), 127–136 (2010)
30. OMG, O.M.G.: Business process modeling notation v 2.0 (2006), `https://www.omg.org/spec/BPMN/2.0/`
31. Petrusel, R., Mendling, J.: Eye-tracking the factors of process model comprehension tasks. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) Advanced Information Systems Engineering, CAiSE Conference (2013)
32. Petrusel, R., Mendling, J., Reijers, H.A.: Task-specific visual cues for improving process model understanding. Information and Software Technology (2016)
33. Ritchi, H., Jans, M.J., Mendling, J., Reijers, H.A.: The influence of business process representation on performance of different task types. Journal of Information Systems (2019)
34. Schreiber, C., Abbad-Andaloussi, A., Weber, B.: On the cognitive effects of abstraction and fragmentation in modularized process models. In: Business Process Management: 21st International Conference, BPM 2023, Utrecht, Netherlands, September 11–15, 2023 (2023)
35. Schreiber, C., Abbad-Andaloussi, A., Weber, B.: On the cognitive and behavioral effects of abstraction and fragmentation in modularized process models. Information Systems **125**, 102424 (2024)
36. Van Der Aalst, W.: Process mining: data science in action, vol. 2. Springer (2016)
37. Wang, W., Chen, T., Indulska, M., Sadiq, S., Weber, B.: Business process and rule integration approaches—an empirical analysis of model understanding. Information Systems **104**, 101901 (2022)
38. Wang, Y., Pan, Z., Zheng, J., Qian, L., Li, M.: A hybrid ensemble method for pulsar candidate classification. Astrophysics and Space Science **364**, 1–13 (2019)
39. Winter, M., Neumann, H., Pryss, R., Probst, T., Reichert, M.: Defining gaze patterns for process model literacy–exploring visual routines in process models with diverse mappings. Expert Systems with Applications **213**, 119217 (2023)
40. Winter, M., Pryss, R., Probst, T., Reichert, M.: Applying eye movement modeling examples to guide novices' attention in the comprehension of process models. Brain sciences **11**(1), 72 (2021)
41. Wolfe, J.M.: Guided search 2.0 a revised model of visual search. Psychonomic bulletin & review **1**, 202–238 (1994)
42. Zimoch, M., Mohring, T., Pryss, R., Probst, T., Schlee, W., Reichert, M.: Using insights from cognitive neuroscience to investigate the effects of event-driven process chains on process model comprehension. In: International conference on business process management. pp. 446–459. Springer (2017)
43. Zimoch, M., Pryss, R., Schobel, J., Reichert, M.: Eye tracking experiments on process model comprehension: lessons learned. In: EMMSAD 2017 Essen, Germany. pp. 153–168. Springer (2017)
44. Zugal, S.: Applying cognitive psychology for improving the creation, understanding and maintenance of business process models. Ph.D. thesis, University of Innsbruck (2013)